Event Graph Neural Network for Opinion Target Classification of Microblog Comments

YAN XIANG, ZHENGTAO YU, JUNJUN GUO, YUXIN HUANG, and YANTUAN XIAN,
Kunming University of Science and Technology

Opinion target classification of microblog comments is one of the most important tasks for public opinion analysis about an event. Due to the high cost of manual labeling, opinion target classification is generally considered as a weak-supervised task. This article attempts to address the opinion target classification of microblog comments through an event graph convolution network (EventGCN) in a weak-supervised manner. Specifically, we take microblog contents and comments as document nodes, and construct an event graph with three typical relationships of event microblogs, including the co-occurrence relationship of event keywords extracted from microblogs, the reply relationship of comments, and the document similarity. Finally, under the supervision of a small number of labels, both word features and comment features can be represented well to complete the classification. The experimental results on two event microblog datasets show that EventGCN can significantly improve the classification performance compared with other baseline models.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**; **Natural language processing;**

Additional Key Words and Phrases: Opinion target, text classification, graph neural network, social media analysis, weak-supervised classification

ACM Reference format:
Yan Xiang, Zhengtao Yu, Junjun Guo, Yuxin Huang, and Yantuan Xian. 2021. Event Graph Neural Network for Opinion Target Classification of Microblog Comments. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 21, 1, Article 17 (November 2021), 13 pages.
https://doi.org/10.1145/3469725

1 INTRODUCTION

Social media, such as microblogs and tweets, has become a platform for people sharing their comments on hot events. Especially those legal-related events that spread rapidly on a microblog platform would trigger public opinion and affect the development of legal decisions to a certain extent. One of the characteristics of microblog comments is that commentators usually express their opinions on certain specific opinion targets. For example, the main opinion targets in microblog comments of the event of a female Mercedes-Benz owner's rights protection are the legal institution merchant and consumer shown in Table 1. In addition, opinion targets of

Authors' addresses: Y. Xiang, Kunming University of Science and Technology, Kunming, China, 650500; email: YanX@kust.edu.cn; Z. Yu (corresponding author), J. Guo, Y. Huang, and Y. Xian, Kunming University of Science and Technology, Kunming, China; emails: ztyu@hotmail.com, guojjgb@163.com, huangyuxin2004@163.com, xianyantuan@qq.com.

2375-4699/2021/11-ART17 $15.00
https://doi.org/10.1145/3469725

Table 1. Example of a Microblog of a Female Mercedes-Benz Owner's Rights Protection

Microblog Content		On April 9, a female car owner in Xi'an, Shaanxi Province, became famous for crying about rights protection on a Mercedes-Benz roof. It was reported that the two sides had reached a settlement. On April 12, Xiaolei (pseudonym), the family member of the car owner, said there was no settlement between the two sides. On the 13th, Mercedes-Benz said it was already negotiating with customers to resolve the matter. #female owners of Xi'an Benz protects her rights#second video of emergency call @ emergency call
Comments	Legal institution	The market supervision department is a decoration.
	Merchant	reply@ahah-sun: Mercedes-Benz is guilty. No negotiation required.
	Consumer	Consumers are vulnerable groups.

different legal-related events are different. For example, the main opinion targets in the event of the Chongqing bus falling into the river are the government agency bus driver and media. To classify microblog comments of an event into different opinion target categories is the basis of fine-grained opinion mining. Microblog comments are usually no more than 140 characters and opinion target classification of microblog comments therefore is a short text classification task.

Traditional text classification studies mainly focus on feature engineering and classification algorithms. The most commonly used feature is the bag-of-words feature, n-grams, **and term frequency–inverse document frequency (TF-IDF)** [1], whereas classification algorithms include Naive Bayes, k-Nearest Neighbor, and Support Vector Machine [2, 3]. They are primarily dependent on the hand-crafted features at the cost of labor and efficiency. In recent years, the neural network models have been widely concerned, and the models based on the Recurrent Neural Network (RNN) [4, 5] and **Convolutional Neural Network (CNN)** [6, 7] have achieved good results for text classification. The neural network-based models can capture the distributed representation of text semantics, which can greatly reduce the work of feature design.

However, these methods need a large number of labeled samples to establish the classifier with good performance. Collecting such training data requires domain experts to read through tremendous documents and carefully label them with domain knowledge. Because of the real-time occurrence of legal-related events, labeling a large number of samples is unrealistic. So, the preceding supervised text classification method is not suitable for this task. To alleviate the problem of insufficient labeled data, some methods based on language models [8] or masked language models [9] are adopted, which are pre-trained over a large amount of text and then fine-tuned on text classification. These methods achieve state-of-the-art performance. In fact, in addition to labeled data, we can also make use of a large number of unlabeled data. Researchers have made use of denoising autoencoders [10, 11] or variational autoencoders [12, 13] to help text classification by introducing extra loss functions over unlabeled data. They utilize latent variables to reconstruct input labeled and unlabeled sentences and predict sentence labels with these latent variables.

Although these semi-supervised methods utilized unlabeled samples effectively, they treat each of the short sentences as independently and identically distributed (IID). Local context only in the sentence itself is focused, and the relational information between sentences is lost. Graph-based methods have recently been applied to solve such issues, which do not treat the text as a sequence but as a set of co-occurrent words instead. For example, Defferrard et al. [14] first employed the graph CNN in the text classification task and outperformed the traditional CNN models. Further, semi-supervised classification based on **graph convolutional networks (GCNs)** had received wide attention [15]. Yao et al. proposed a text GCN (TextGCN), which turned the text classification problem into a node classification one [16], but fine-grained text level word interactions were not considered [17]. Zhang et al. [18] incorporated the word nodes as the document embedding to capture the contextual word relationships within each document. Moreover, Huang et al. [19]

improved TextGCN by introducing the message passing mechanism and reducing the memory consumption. Inspired by the attention mechanism, the **graph attention network (GAT)** [20] introduced the attention mechanism into the GCN by modifying the convolution operation. Most existing graph-based studies focus on long texts and achieve unsatisfactory performance on short texts due to the sparsity and limited labeled data. For modeling the short texts, Hu el al. [21] proposed a heterogeneous information network framework (HIN), which can integrate any type of additional information as well as capture their relations to address the semantic sparsity.

These graph neural network-based methods provide good solutions for text classification, but they are not designed especially for the opinion target classification of microblog comments. In fact, legal-related event microblog data has its own characteristics. Microblog includes content and comment, and these two kinds of different documents have various types of connections. On the one hand, contents and comments of the same legal-related event will be associated with one or more event keywords. For example, the keyword Mercedes-Benz is mentioned by the content and the comment of merchant at the same time, shown in Table 1. In addition, there are reply connections between two comments. Some comments aim at other comments of different users, and they usually discuss the same opinion target. For instance, the comment "reply@ahah-sun: Mercedes-Benz is guilty. No negotiation required" in Table 1 replied to the comment "Will anyone dare to buy Mercedes-Benz in the future" published by the user ahah-sun. Both comments are about the merchant. Using these connections effectively can make unlabeled data learn the classification characteristics of labeled data, thus improving the problem of insufficient label data in text classification. Therefore, we propose an event graph neural network (EventGCN) for opinion target classification of microblog comments. We offer the following contributions:

- We construct an event graph neural network for opinion target classification of microblog comments by taking both contents and comments as document nodes, and combining explicitly three kinds of document nodes connection to get an event graph (i.e., keywords-based, reply-based, and similarity-based graph).
- We use the pre-trained word vectors on the large-scale Sina microblog data as the initial features of the word nodes in the event graph, and give more weights to those nodes related closely to the keywords of the event, which further improve the classification performance.
- We test the proposed event graph neural network and the other state-of-the-art models on two legal-related event microblog datasets for opinion target classification, and the results show that the proposed models obtain the best classification performance.

The rest of this article is organized as follows. In Section 2, we describe our proposed method for opinion target classification. Section 3 gives the experiments and results. Finally, Section 4 concludes the article.

2 METHODS

In this study, we utilize the GCN as a basic component for text feature learning, due to its simplicity and effectiveness in practice. In this section, we first give a brief explanation of an event GCN. Then, we introduce details on how to construct an event graph adjacency matrix and feature matrix from a text corpus. Ultimately, we complete opinion target classification.

2.1 The Basic Event GCN

A GCN is a generalization version of traditional CNNs that can operate directly on a graph. In this article, we propose an event GCN based on GCN, shown in Figure 1.

Formally, an event graph is represented as $G = (V, E, A)$, where V is the set of nodes, E is the set of edges, and $A \in \mathbf{R}^{n \times n}$ is the graph adjacency matrix of n nodes. Let $v_i \in V$ denote a node and

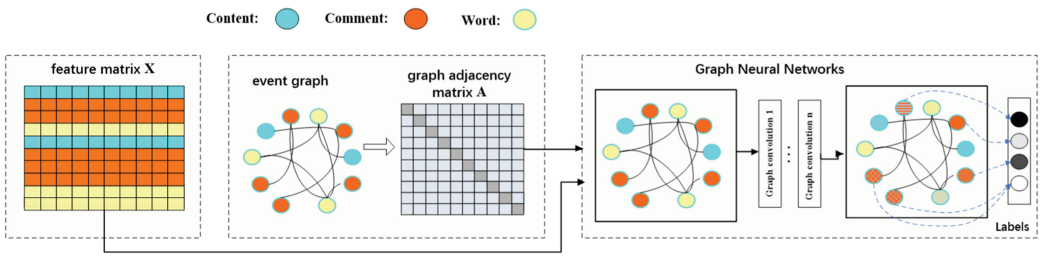


Fig. 1. The proposed EventGCN.

$e_{ij} = (v_i, v_j) \in E$ denote an edge pointing from v_j to v_i. In Figure 1, all of the content, comments, and words are nodes. Let $X \in \mathbf{R}^{n \times d}$ be a matrix containing all n nodes with their features. Each row $x_v \in \mathbf{R}^d$ is the feature vector for v. In EventGCN learning, hidden layer representations are obtained by encoding both graph structure and features of nodes with a kind of propagation rule. For a one-layer EventGCN, the new k-dimensional node feature matrix is computed as follows:

$$L^{(1)} = \sigma(\tilde{A}XW_0), \tag{1}$$

where $L^{(1)} \in \mathbf{R}^{n \times k}$, $\tilde{A} = D^{\frac{-1}{2}}AD^{\frac{-1}{2}}$ is the normalized symmetric adjacency matrix and $W_0 \in \mathbf{R}^{d \times k}$ is a weight matrix. σ is a sigmoid activation function.

We can incorporate high-order neighborhoods information by stacking multiple EventGCN layers:

$$L^{(j+1)} = \sigma(\tilde{A}L^{(j)}W_j), \tag{2}$$

where j denotes the layer number and $L^{(0)} = X$.

2.2 The Structure of the Event Graph

In this article, we propose the event graph, which fully utilizes the relationship between different nodes based on event keywords, reply association, and document similarity.

2.2.1 The Graph Adjacency Matrix. The proposed graph adjacency matrix is shown in Figure 2. We construct the graph adjacency matrix by the formula shown in Equation (3).

$$A_{ij} = \begin{cases} D(i,j), i \text{ and } j \text{ are documents} \\ PMI(i,j), i \text{ and } j \text{ are words} \\ TF\text{-}IDF_{ij}, i \text{ is a document}, j \text{ is a word} \\ 1, i = j \end{cases} \tag{3}$$

In this adjacency matrix, the weight of the edge between a document node and a word node is the TF-IDF. The term frequency is the number of words appearing in the document, and the inverse document frequency is the logarithmic proportional inverse fraction of the number of documents with this word.

The weight of the edge between word nodes is the **point-wise mutual information (PMI)**:

$$PMI(i,j) = \log \frac{p(i,j)}{p(i)p(j)}, \tag{4}$$

$$p(i,j) = \frac{\#W(i,j)}{\#W}, \tag{5}$$

$$p(i) = \frac{\#W(i)}{\#W}, \tag{6}$$

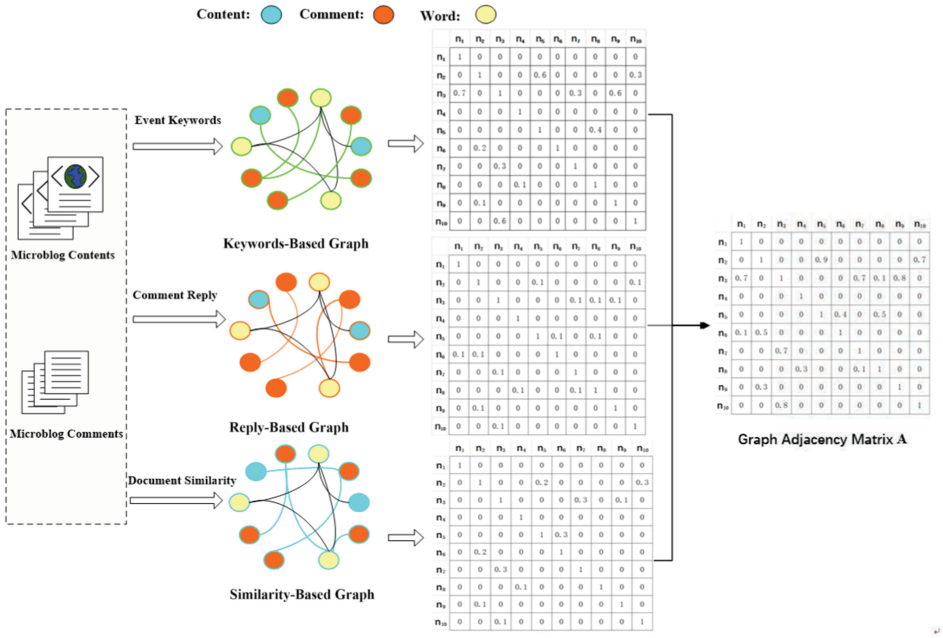


Fig. 2. The proposed graph adjacency matrix.

where $\#W(i)$ is the number of sliding windows in a corpus that contain word i, $\#W(i,j)$ is the number of sliding windows that contain both word i and j, and $\#W$ is the total number of sliding windows in the corpus.

The weight of the edge between document nodes $D(i,j)$ in the formula (3) is calculated by the following:

$$D(i,j) = E(i,j) + R(i,j) + S(i,j), \tag{7}$$

where $E(i,j)$, $R(i,j)$, and $S(i,j)$ are weights of the edge between document nodes based on the keywords-based graph, reply-based graph, and similarity-based graph, respectively, shown in Figure 2.

To get the keywords-based graph, we first extract the event-related keywords from all the microblog contents by TextRank [22]. As discussed before, these keywords will appear in contents or comments, representing different opinion targets. So, two documents having the same keywords would be prone to be the same opinion target category. Based on these keywords, we get the keywords-based weight of the edge between document nodes by the following formula:

$$E(i,j) = \frac{\#\{e_k | e_k \in E \& e_k \in D_i \& e_k \in D_j\}}{\#E}, \tag{8}$$

where $\#E$ is the total number of the keywords in the certain event corpus, e_k is the $k'th$ keyword in the keywords set, and D_i represents the $i'th$ document.

Next, for two comments with the reply association, they would be prone to discuss the same target category. So, we get the reply-based weight of the edge between document nodes by the formula in Equation (9).

$$R(i,j) = \begin{cases} 0.1, D_i \ and \ D_j \ have \ reply \ relationship \\ 0, others \end{cases} \tag{9}$$

What is more, two similar documents are more likely to be the same category. So, we get the similarity-based weight by the following formula:

$$S(i,j) = \frac{\#\{w_k | w_k \in D_i \& w_k \in D_j\}}{\#D_i + \#D_j}, \tag{10}$$

where $\#D_i$ and $\#D_j$ represent the word numbers of the two documents D_i and D_j separately, whereas w_k represents a word in the document.

According to the preceding calculations, we construct a complete graph adjacency matrix. In addition, every node is assumed to be connected to itself. So, the diagonal elements of A are set to 1 because of self-loops.

2.2.2 Initiation of the Node Feature. We first extract the event-related keywords set E according to the preceding method. Let $x_i \in \mathbf{R}^d$ represent the d-dimensional word vector of the $i'th$ word in the vocabulary of the corpus. The word vector is embedded by the pre-trained word vector. In addition, If the $i'th$ word is in E, then its word vector x_i will be embedded by the pre-trained embedding multiplying coefficient α. By doing this, we get a keyword-strengthened word vector matrix. For a word node, its initial feature is represented as the corresponding word vector in the word vector matrix. For a document node, its initial feature is represented as the average word vector in the sentence. Finally, the graph feature matrix $X \in \mathbf{R}^{n \times d}$ is represented as a matrix containing all n nodes with their initial features.

2.3 The Opinion Target Classification

After building the event graph, we feed the graph into a simple two-layer EventGCN. The node (word/document) features of the second layer have the same size as the labels set and are fed into a softmax classifier:

$$Z = softmax(\tilde{A}ReLU(\tilde{A}XW_0)W_1), \tag{11}$$

where ReLU is an activation function and $W_1 \in \mathbf{R}^{k \times C}$ is a weight matrix to be trained. The loss function is defined as the cross-entropy error over all labeled comments:

$$L = -\sum_{d \in D_l} \sum_{c=1}^{C} Y_c \ln Z_c, \tag{12}$$

where D_l is the set of document indices that have a labeled opinion target category and C is the dimension of the output features, which is equal to the number of categories. Y is the label indicator matrix.

3 EXPERIMENT

3.1 Datasets

We collected two legal-related event datasets from the Sina microblog platform for model training and evaluation. Three experts labeled opinion target categories for the comments at the same time and finally selected the comments with consistent labels as labeled samples. The basic information of the datasets is shown in Table 1. The first dataset is about an event that a female Mercedes-Benz owner protected her rights. This dataset contains 33,220 samples with four opinion targets, namely legal institution, merchant, consumer, and others. The second dataset is about an event that the Chongqing bus fell into the river. The dataset contains 20,294 samples with four opinion targets, namely government agency, bus driver, media, and others. We divide 70% labeled samples as the test set for final classification performance evaluation and use the remaining 30% samples to be alternative training data. In the experiments from Section 3.1 through 3.4, we use 12% labeled samples as training data.

Table 2. Summary Statistics of Datasets (# Represents Number)

Datasets	#Microblog	#Total Documents	#Labeled Samples	#Test	#Words	Classes
Dataset 1: The Female Mercedes-Benz Owner's Rights Protection	168	32,220	Legal institution: 865 Merchant: 640 Consumer: 290 Others: 130	1,514	13,098	4
Dataset 2: The Chongqing Bus Falling into the River	97	20,294	Government agency: 286 Bus driver: 564 Media: 662 Others: 146	1,116	8,694	4

3.2 Baselines

For a comparison, we adopt nine classification methods as baselines.

SVM [23]. Support Vector Machine trained on the TF-IDF weighted word frequency vector for comments.

BiLSTM [24]. This method uses a standard Bi-Long Short-Term Memory network as the sentence encoder. The last hidden state of the LSTM is used for prediction. In our experiment, the hidden states are set to 128.

TextCNN [6]. This aims to generate sentence classification based merely on the textual features. Our experiments follow the setting of Kim [6] with Nonstatic_CNN and filters width = 3, 4, 5.

FastText [25]. Inspired by the work in efficient word representation learning, the FastText model structure is similar to CBOW. It treats the average of word/n-gram embeddings as document embeddings, then feeds document embeddings into a linear classifier. We evaluated it with bigrams.

Bert base [9]. The model has 12 layers of Transformer, and each layer of the Transformer has 12 self-attention heads. The size of the hidden layer is 768, and the total parameter amount is 110M. We use the Chinese pre-trained BERT model released by Google[1] and fine-tune the model for the opinion target classification task.

ABAE_labeled [10]. This is a neural topic model. The labeled and unlabeled samples are encoded and reconstructed, and the model is trained by combining reconstruction loss with classification loss simultaneously. The topic distribution is taken as the final classification feature. The learning rate is 0.001, and the optimizer is Adam.

MATE_labeled [11]. This is a weak supervised learning model based on the ABAE model. Several seed words for each opinion target are selected, and these opinion target embeddings are initialized with the average value of the corresponding seed word embedding, which are fixed in the whole training process. In the experiment, 10 seed words were selected for each type of opinion target, and the other model parameters were consistent with ABAE_labeled.

TextGCN [16]. This is a graph-based text classification model that builds a single large graph for whole corpus. For TextGCN, we simply set feature matrix as an identity matrix, which means that every word or document is represented as a one-hot vector as the input.

GAT [20]. This is a novel convolution-style neural network that leverages masked self-attentional layers. We apply a two-layer GAT model following the transductive learning task in the work of Velickovic et al. [20]. The first layer consists of eight attention heads, followed by an exponential linear unit (ELU) nonlinearity. The second layer is used for classification, which has a single attention head, followed by a softmax activation.

[1] https://github.com/google-research/bert.

Table 3. Classification Results of Opinion Targets in Dataset 1 Based on Different Models

Categories	Merchants			Legal Institution			Consumer			Weighted Average		
Models	P	R	Fl	P	R	Fl	P	R	Fl	P	R	Fl
SVM	0.740	0.644	0.689	0.730	0.824	0.744	0.447	0.43	0.453	0.681	0.679	0.677
BiLSTM	0.791	0.756	0.773	0.738	0.897	0.810	0.742	0.490	0.590	0.761	0.760	0.753
TextCNN	0.759	0.782	0.770	0.733	0.873	0.796	0.852	0.460	0.597	0.766	0.756	0.748
FastText	0.754	0.791	0.772	0.752	0.848	0.797	0.794	0.500	0.613	0.761	0.758	0.752
Bert base	0.747	0.787	0.766	0.713	0.799	0.745	0.826	0.570	0.675	0.749	0.743	0.741
ABAE-labeled	0.716	0.703	0.709	0.853	0.702	0.770	0.441	0.763	0.559	0.751	0.710	0.721
MATE-labeled	0.723	0.708	0.715	0.853	0.713	0.776	0.461	0.768	0.575	0.755	0.718	0.728
TextGCN	0.796	0.849	0.822	0.757	**0.961**	0.847	**0.890**	0.312	0.462	0.819	0.788	0.763
GAT	0.733	0.834	0.780	0.784	0.792	0.788	0.847	0.794	0.820	0.776	0.807	0.790
EventGCN	**0.890**	**0.917**	**0.903**	**0.868**	0.880	**0.874**	0.824	**0.750**	**0.785**	**0.869**	**0.870**	**0.869**

Table 4. Classification Results of Opinion Targets in Dataset 2 Based on Different Models

Categories	Government Agency			Media			Bus Driver			Weighted Average		
Models	P	R	Fl	P	R	Fl	P	R	Fl	P	R	Fl
SVM	0.596	0.932	0.727	0.880	0.509	0.645	0.669	0.882	0.761	0.750	0.728	0.706
BiLSTM	0.798	0.682	0.736	0.703	0.953	0.809	**0.900**	0.706	0.791	0.766	0.803	0.773
TextCNN	**0.964**	0.578	0.723	0.946	0.906	0.926	0.682	**0.894**	0.774	0.838	0.778	0.784
FastText	0.702	0.784	0.741	0.723	0.941	0.818	0.807	0.722	0.762	0.761	0.823	0.784
Bert base	0.711	0.529	0.607	0.665	0.907	0.767	0.765	0.702	0.732	0.716	0.775	0.736
ABAE-labeled	0.628	0.848	0.721	0.714	0.753	0.728	0.776	0.759	0.768	0.694	0.790	0.733
MATE-labeled	0.641	0.843	0.727	0.726	0.764	0.74	0.784	0.764	0.774	0.705	0.794	0.742
TextGCN	0.758	0.98	0.855	0.858	0.949	0.901	0.84	0.62	0.713	0.84	0.797	0.809
GAT	0.862	0.889	0.875	0.822	0.833	0.828	0.744	0.867	0.801	0.794	0.855	0.822
EventGCN	0.873	**0.984**	**0.925**	**0.965**	**0.957**	**0.961**	0.870	0.870	**0.870**	**0.912**	**0.920**	**0.916**

3.3 Parameter Setting

We used default parameter settings of baseline models in their original papers or implementations. For those models that need word embedding, we used the 300-dimensional word vector trained on the 0.73G microblog dataset, which has about 850,000 vocabularies.[2] For EventGCN, we set coefficient α as 10 and keywords as 30. With those parameters, the model achieved the best results on the validation set. Following previous studies of Yao et al. [16], we set the number of sliding windows as 20, learning rate as 0.02, dropout rate as 0.5, and L2 loss weight as 0. We randomly selected 10% of labeled data in the training set as the validation set and trained EventGCN for a maximum of 500 epochs using Adam.

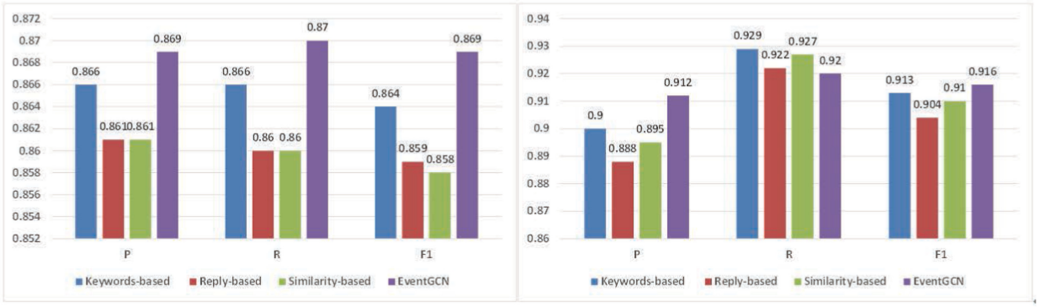
3.4 Experiment Results and Analysis

In this section, we report our experimental results and findings.

3.4.1 Overall Text Classification Performance. First, we compare the classification results of EventGCN with the other nine baseline models on two datasets, and calculate precision (P), recall (R), F1 of each category, and weighted average, as shown in Tables 3 and 4.

According to the results in Tables 3 and 4, the overall performance of SVM is the worst, and its weighted average F1 value on datasets 1 and 2 is only 0.677 and 0.706. It shows that when there is only a small amount of labeled data, the opinion targets cannot be distinguished well based on

[2]https://github.com/Embedding/Chinese-Word-Vectors.



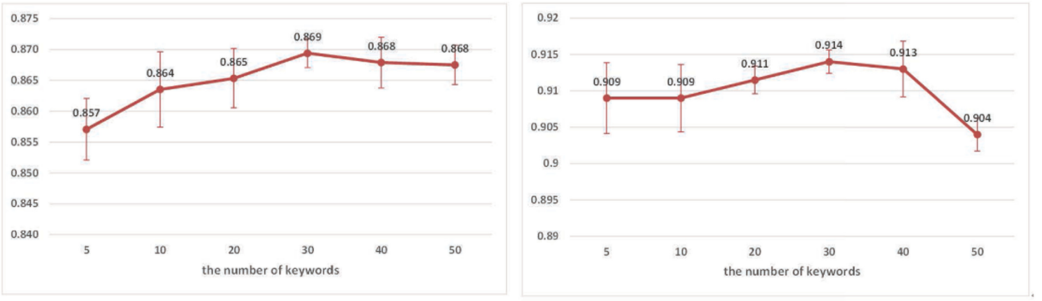
(a) Results of dataset 1 (b) Results of dataset 2

Fig. 3. Classification results of EventGCN based on different document nodes relationships.

the TF-IDF feature. The performance of Bi-LSTM, TextCNN, and FastText is similar. The weighted average F1 values of these three models on dataset 1 are about 0.75, and those on dataset 2 are about 0.78. Compared with the Bert base classification model, the performance of these three models is better. The reason may be that in our experiment, the Bert base model only directly uses the pre-trained vectors in a large-scale corpus, whereas Bi-LSTM, TextCNN, and FastText learn to extract classification features according to the specific dataset. As far as neural topic models are concerned, although the ABAE-labeled model and the MATE-labeled model make use a large number of unlabeled data, topic features adopted by them may be insufficient for opinion target classification. Therefore, their weighted average F1 values on the two datasets are only about 0.72 and 0.73. TextGCN explicitly uses the relationship between data, which is in line with the characteristics of social media data. Therefore, TextGCN obtains weighted average F1 values of 0.763 and 0.809 on the two datasets, which are more than the seven models mentioned previously. GAT improves weighted average F1 values upon TextGCN by 0.27 and 0.13 on the two datasets, respectively, showing that assigning different weights to nodes of a same neighborhood is beneficial. As a novel graph-based method, EventGCN fully considers the characteristics of the event microblog data, and achieves the highest F1 values of 0.869 and 0.916 on the two datasets by strengthening the related information of document nodes and initial nodes features.

Moreover, for most models, the fewer the label samples of a certain opinion target category, the worse its classification result. For example, in dataset 1, the consumer category has the least labeled samples compared with the other two categories, and it has the worst performance. But graph-based methods do not necessarily have this regularity. For example, in dataset 2, the bus driver category has more tags than that of the government agency category, but bus driver results of the graph-based models are worst, whereas government agency results are better. This indicates that the performance of graph-based models does not mainly depend on the number of labeled samples. The key of graph-based models is whether they can successfully capture relationships between document nodes of the same category. If the association learning between labeled samples and unlabeled samples of some category is sufficient, then unlabeled samples are more likely be predicted correctly.

3.4.2 Ablation Experiment of Different Document Nodes Relationships. In this section, we conduct more experiments to study the effect of different document nodes relationships. Figure 3 shows the weighted average values of precision, recall, and F1, where Keywords-based, Reply-based, and Similarity-based mean that we only use formulas (8), (9), and (10) to calculate the weights of document nodes and get three different graph adjacency matrixes in Figure 2. The initiation of the node feature is the same as EventGCN.



(a) Results of dataset 1 (b) Results of dataset 2

Fig. 4. Classification results of EventGCN with different numbers of keywords.

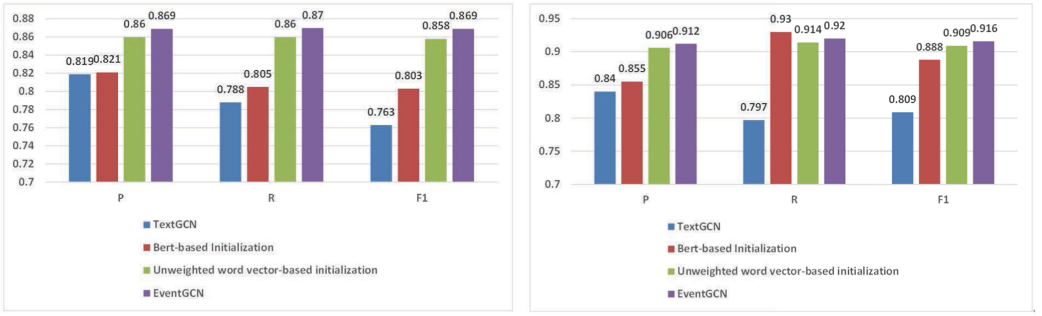
For dataset 1, the values of P, R, and F1 of the keywords-based graph improve about 0.05 compared with those of the reply-based graph and the similarity-based graph, which means that for our task, using keywords to establish the correlation between different documents is more effective than using the reply relationship or the similarity of the documents. The results of the reply-based graph and the similarity-based graph are similar. In addition, when the three kinds of graphs are used at the same time, the F1 value is improved by 0.11 compared with using the similarity-based graph alone. It shows that the proposed combination of the three graphs is effective.

For dataset 2, using the reply-based graph alone has the lowest values. Better results can be achieved by using the similarity-based graph or the keywords-based graph alone. In particular, the keywords-based graph has obvious advantages. The same as dataset 1, the combination of the three graphs achieves the best results.

3.4.3 Effect of Keywords Numbers. The experiments in the previous section have proved that the keywords-based graph plays an important role in modeling. In this section, we test the effect of different numbers of keywords. For each number of keywords, we calculate the mean and standard deviation of the weighted average F1 over 10 runs of experiments, shown in Figure 4.

For dataset 1, the weighted average F1 first increases with the increase of the keywords number, reaching the highest value at 30. For dataset 2, the weighted average F1 also reaches the highest when the keywords number is 30, whereas there is a big decrease when the number is 50. Relatively low standard deviations appear in 30 and 50 keywords of dataset 1, as well as 20 and 30 keywords of dataset 2. For both datasets, the performance is similar when the keywords number is between 20 and 40. This suggests that few keywords could not generate sufficient relation information for document nodes, whereas too many keywords could introduce noise information to some non-strongly correlated documents. On the whole, we can obtain good results for our datasets when the number of keywords is between 20 and 40.

3.4.4 Effect of Initial Nodes Features. In this section, we test the effect of initial features of nodes. Figure 5 shows the weighted average values of precision, recall, and F1 of different initialization methods. Unweighted word vector-based initialization means that when we calculate the feature matrix X, we set the coefficient α as equal to 1. Bert-based initialization means that we take characters as nodes instead of words. Meanwhile, we set the character node feature as the Chinese Bert character vector and the document node feature as the document vector encoded by the Bert pre-trained model. The Bert pre-trained model here is the same as the baseline Bert base.



(a) Results of dataset 1 (b) Results of dataset 2

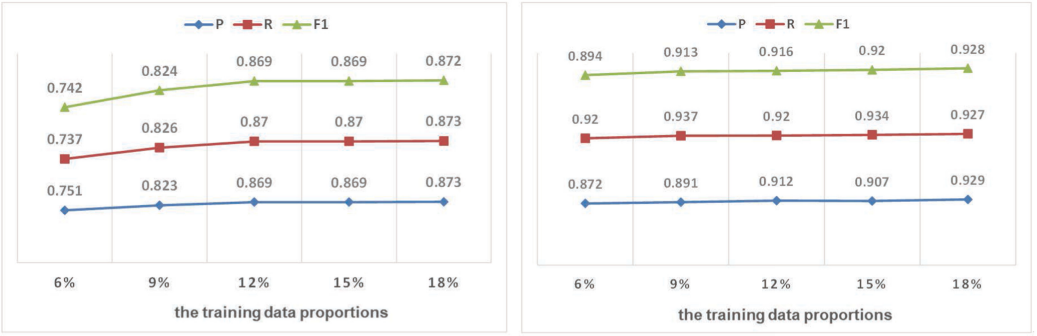
Fig. 5. Classification results of EventGCN based on different initial nodes features.

We note that the performance of different initialization methods is quite different, which suggests that for the graph convolution-based method, the initialization of node features is very important. P, R, and F1 values of TextGCN are lowest. In addition to the influence of the adjacency matrix, the reason for the relatively poor performance of TextGCN is that its initial nodes feature is highly sparse one-hot vectors. On the one hand, high dimensions of one-hot vectors make W_0 the high dimension, too, which is not beneficial for learning. On the other hand, this kind of initial nodes feature does not have prior knowledge like Bert or word2vec.

The Bert-based initialization model has obvious improvement compared with TextGCN, and its recall value in dataset 2 is even higher than that of EventGCN. It shows that Bert-based pre-trained vectors are more effective compared with the one-hot vector. However, Bert-based pre-trained vectors are not better than word2vec pre-trained on microblogs. The main reason may be that the Bert-based initialization model uses characters as nodes. But the relationships between character nodes, as well as the relationships between character nodes and document nodes, are not strong enough. As a result, the weight representation of the edge between nodes is not good in the adjacency matrix, which damages the performance of the Bert-based initialization model. What is more, EventGCN has a slight improvement compared with the unweighted word vector-based initialization model, which indicates that it is beneficial to emphasize the initial nodes features by keywords. On the whole, the experimental results show that it is advantageous for the graph convolution-based classification model to make full use of prior knowledge to initialize node features.

3.4.5 Effects of the Size of the Labeled Training Data. To evaluate the effect of the size of the labeled training data, we tested EventGCN with different training data proportions. Figure 6 reports the weighted average values of test samples with 6%, 9%, 12%, 15%, and 18% of labeled samples as the training set.

We note that EventGCN can achieve high values with limited labeled documents. For weighted average F1, EventGCN achieves 0.742 on dataset 1 and 0.894 on dataset 2 with only 6% of labeled samples, which are higher than multiple baseline models with 12% of labeled samples shown in Tables 3 and 4. With the further increase of labeled samples, the effect is further improved but not obvious. It suggests that the proposed event graph carries abundant information of documents in regard to opinion targets. EventGCN is based on this event graph and can propagate document label information to the entire graph well, and thus can identify the opinion target category of unlabeled document. On the whole, EventGCN has an encouraging advantage for the weak supervised classification.



(a) Results of dataset 1 (b) Results of dataset 2

Fig. 6. Classification results of EventGCN by varying training data proportions.

4 CONCLUSION

In this study, we propose a novel opinion target classification method for microblog comment, called *EventGCN*. We regard both microblog content and comment as document nodes, and utilize event keywords of microblog to capture the relationship of document nodes and to initialize node features. By doing this, we build an event GCN for a whole corpus, which can make full use of limited labeled documents well to capture features of unlabeled document nodes for opinion target classification. Experiment results on two microblog datasets show that EventGCN obtains outstanding results based on very few annotation data and outperforms multiple state-of-the-art methods of opinion target classification. In addition, we find that it has obvious advantages for opinion target classification in capturing the keywords-based graph and initial features based on event keywords. In the future, EventGCN can be applied to other event microblogs to test its capability.

REFERENCES

[1] Sida I. Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 90–94. https://www.aclweb.org/anthology/P12-2018/.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022. http://jmlr.org/papers/v3/blei03a.html.

[3] George Forman. 2008. BNS feature scaling: An improved representation over TF-IDF for SVM text classification. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM'08)*. ACM, New York, NY, 263–270. DOI: http://dx.doi.org/10.1145/1458082.1458119

[4] Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH'10)*. 1045–1048. http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.

[5] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*. 2873–2879. http://www.ijcai.org/Abstract/16/408.

[6] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1746–1751. DOI: http://dx.doi.org/10.3115/v1/d14-1181

[7] Ming Hao, Bo Xu, Jing-Yi Liang, Bo-Wen Zhang, and Xu-Cheng Yin. 2020. Chinese short text classification with mutual-attention convolutional neural networks. *ACM Transactions on Asian and Low-Resource Language Information Processing* 19, 5 (2020), Article 61, 13 pages. https://dl.acm.org/doi/10.1145/3388970.

[8] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'18) (Volume 1: Long Papers)*. 2227–2237. DOI: http://dx.doi.org/10.18653/v1/n18-1202

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19) (Volume 1: Long and Short Papers)*. 4171–4186. DOI: http://dx.doi.org/10.18653/v1/n19-1423

[10] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17) (Volume 1: Long Papers)*. 388–397. DOI: http://dx.doi.org/10.18653/v1/P17-1036

[11] Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3675–3686. DOI: http://dx.doi.org/10.18653/v1/d18-1403

[12] Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2018. Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 215–226. DOI: http://dx.doi.org/10.18653/v1/d18-1020

[13] Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019. Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL'19) (Volume 1: Long Papers)*. 5880–5894. DOI: http://dx.doi.org/10.18653/v1/p19-1590

[14] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*. 3837–3845. https://proceedings.neurips.cc/paper/2016/hash/04df4d434d481c5bb723be1b6df1ee65-Abstract.html.

[15] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*. https://openreview.net/forum?id=SJU4ayYgl.

[16] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*. 7370–7377. DOI: http://dx.doi.org/10.1609/aaai.v33i01.33017370

[17] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*. 346–353. DOI: http://dx.doi.org/10.1609/aaai.v33i01.3301346

[18] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. 334–339. DOI: http://dx.doi.org/10.18653/v1/2020.acl-main.31

[19] Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 3442–3448. DOI: http://dx.doi.org/10.18653/v1/D19-1345

[20] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*. https://openreview.net/forum?id=rJXMpikCZ.

[21] Linmei Hu, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. 4820–4829. DOI: http://dx.doi.org/10.18653/v1/D19-1488

[22] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP'04)*. 404–411. https://www.aclweb.org/anthology/W04-3252/.

[23] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2, 3 (2011), Article 27, 27 pages. DOI: http://dx.doi.org/10.1145/1961189.1961199

[24] Alex Graves, Navdeep Jaitly, and AbdelRahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, Los Alamitos, CA, 273–278. DOI: http://dx.doi.org/10.1109/ASRU.2013.6707742

[25] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás 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 (EACL'17) (Volume 2: Short Papers)*. 427–431. DOI: http://dx.doi.org/10.18653/v1/e17-2068

Received October 2020; revised March 2021; accepted June 2021