Proceedings of the Eighteenth International AAAI Conference on Web and Social Media (ICWSM 2024)

DoubleH: Twitter User Stance Detection via Bipartite Graph Neural Networks

Chong Zhang[1], **Zhenkun Zhou**[2*], **Xingyu Peng**[1], **Ke Xu**[1]

[1] State Key Lab of Software Development Environment, Beihang University, Beijing, China
[2] School of Statistics, Capital University of Economics and Business, Beijing, China
chongzh@buaa.edu.cn, zhenkun@cueb.edu.cn, xypeng@buaa.edu.cn, kexu@buaa.edu.cn

Abstract

Given the development and abundance of social media, studying the stance of social media users is a challenging and pressing issue. Social media users express their stance by posting tweets and retweeting. Therefore, the homogeneous relationship between users and the heterogeneous relationship between users and tweets are relevant for the stance detection task. Recently, graph neural networks (GNNs) have developed rapidly and have been applied to social media research. In this paper, we crawl a large-scale dataset of the 2020 US presidential election and automatically label all users by manually tagged hashtags. Subsequently, we propose a bipartite graph neural network model, DoubleH, which aims to better utilize homogeneous and heterogeneous information in user stance detection tasks. Specifically, we first construct a bipartite graph based on posting and retweeting relations for two kinds of nodes, including users and tweets. We then iteratively update the node's representation by extracting and separately processing heterogeneous and homogeneous information in the node's neighbors. Finally, the representations of user nodes are used for user stance classification. Experimental results show that DoubleH outperforms the state-of-the-art methods on popular benchmarks. Further analysis illustrates the model's utilization of information and demonstrates stability and efficiency at different numbers of layers.

Introduction

Stance detection is the task of automatically mining text author's position from text (Mohammad et al. 2016), and has found widespread applications in fields such as political elections and public opinion research. Social media provides a platform for individuals to voice their opinions, and studies on the 2016 US presidential election (Bovet and Makse 2019; Caetano et al. 2018) and the 2019 Argentine presidential election (Zhou et al. 2021) have demonstrated the value of social media data for analyzing election events. In recent years, as the number of social media users has increased, social media has become increasingly intertwined with political events (Gayo-Avello, Jungherr, and Rivero 2023). Consequently, accurately and efficiently analyzing this information has become an urgent issue.

Stance detection in social media networks primarily focuses on users and texts. In the user stance detection task, user-related information is used to determine the stances of users. Supervised methods employ a range of features, such as tweets, user profile information, following and retweeting, to train the classifier (Pennacchiotti and Popescu 2011; Wong et al. 2021; Magdy et al. 2016). In social networks, members of homogeneous groups tend to share similar stances (Yang and Eisenstein 2017; Kovacs and Kleinbaum 2020) on various topics. The idea of homogeneity is widely used in semi-supervised and unsupervised methods. The semi-supervised methods are grounded on the labels of some users, and employ label propagation on the user's following network (Barberá 2015) or retweet network (Borge-Holthoefer et al. 2015; Weber, Garimella, and Batayneh 2013), to label other users. The unsupervised methods utilize user similarity information to label users through user network (Jiang, Ren, and Ferrara 2022) or clustering (Darwish et al. 2020).

Most of the existing research on user stance detection is devoted to the feature selection and combination of text and network information. Unsupervised or semi-supervised methods find similar user pairs through the relationship between users and train models based on user information. However, it is difficult for these methods to outperform supervised learning methods. Manually labeling is time-consuming, and the user's stance can be better obtained if more labeled data and higher labeling efficiency are available. Existing methods consider user characteristics such as user profiles and tweets (Preoţiuc-Pietro et al. 2017; Rao et al. 2010), or emphasize the relationship between users (Darwish, Magdy, and Zanouda 2017; Jiang, Ren, and Ferrara 2022; Darwish et al. 2020). These methods interact between users and other users with homogeneous information. However, users are not the only objects that users interact with. For retweet relations between users, users establish a retweet relation through tweets as a bridge. Tweets contain rich semantic information and form a binary relationship with users as heterogeneous information. Therefore, our aim is to automatically label user data and better utilize and balance both homogeneous and heterogeneous information.

In this paper, our focus is on the 2020 US presidential election. Initially, we collect all pertinent users and tweets leading up to the election and divide them into supporters

*Corresponding author: Zhenkun Zhou.

of either Joe Biden or Donald Trump. For large amounts of user data, manual labeling methods such as crowdsourcing or preset political user lists (Li and Goldwasser 2019) would be cost-intensive. We observe that hashtags in tweets represent the topics that the tweets discuss. In election events, hashtags are an expression of user sentiment or opinion with a straightforward structure and clear tendency. The effectiveness of hashtags in stance labeling has been indicated in prior research (Bovet, Morone, and Makse 2018). Therefore, we manually collect and label a collection of hashtags, where each hashtag is tagged as supporting Biden or Trump. Next, we label the tweets based on the hashtags they contain. As tweets are an expression of user opinion, we calculate the label ratio of each user's tweets and establish a threshold based on the distribution of the tweet label ratio of the user. Finally, we annotate all users by comparing the tweet label ratio with the threshold.

In recent years, graph neural networks (GNNs) have developed swiftly and have state-of-the-art performance in multiple fields. GNNs learn the representation of each node by aggregating the information of neighboring nodes and can retain structural information in the graph embedding. In the user's stance detection task, GNNs extract and mine the information in the retweet graph of users and tweets, and learn the representation of nodes grounded on the relationship between nodes. In the information aggregation of GNNs, user and tweet nodes can be regarded as information of different types of nodes or can be both considered as text nodes and employ heterogeneous or homogeneous graph neural networks to learn the representations. In this paper, we propose a novel graph neural network, named DoubleH, based on a user-tweet bipartite graph for user stance detection. DoubleH aims to exploit both homogeneous and heterogeneous information to update node representation. As shown in Figure 1, users and tweets are used as two different types of nodes to construct the user-tweet bipartite graph. In each iteration, the heterogeneous information represented by one-hop neighbors and the homogeneous information of two-hop neighbors are extracted and processed separately, and then jointly interact with the central node. DoubleH leverages the learned node representations to classify user stances. Our method makes better use of homogeneous and heterogeneous information. We conduct several experiments to verify the advantages of our approach over the baselines. To sum up, our contributions are as follows:

- We gathered data on users and tweets related to the 2020 US presidential election, and used hashtags to automatically label user data. This enabled us to create a large-scale dataset suitable for supervised learning.
- We propose a new method called DoubleH, which leverages the user's profile, personal information, and tweets as initial inputs, with the aim of parsing and aggregating both homogeneous and heterogeneous information from the user-tweet bipartite graph.
- The experimental results demonstrate that DoubleH not only outperforms several graph-based baselines but also exhibits considerable efficiency. Further experiments highlight the impact of the number of layers on the model and the distinct utilization of homogeneous and heterogeneous information.

Related Work

We categorize the existing research on user stance detection into three groups: supervised learning, semi-supervised learning, and unsupervised learning, based on the requirement for fully labeled data. In addition, graph neural networks have achieved good performance in social network research, we finally introduce methods related to graph neural networks and other advanced GNNs.

Stance Detection

Supervised Methods. Supervised stance detection models exploit labeled data to train a classifier to assign stance labels. Several content-based approaches use textual information such as tweets, user profiles, and hashtags to train classifiers. Hashtags have been identified as an important feature to classify users' political positions in earlier studies (Conover et al. 2011). Socio-linguistic features such as utterances, e.g., emoticons, abbreviations, and word n-grams were used to distinguish between Republicans and Democrats (Rao et al. 2010). Some deep learning models for encoding text have also been applied to stance detection tasks, Augenstein et al. (2016) employed LSTM and Bidirectional LSTM to encode the tweet text on stance detection. Tran, Phung, and Ngo (2022) extracted contextual word embeddings of tweets by the large-scale pre-trained language model and applied a convolutional neural network (CNN) for further feature extraction for training a stance detection model. However, these advanced deep learning models operate at the tweet level, not the user level. The concept of homophily ideology (Yang and Eisenstein 2017; Kovacs and Kleinbaum 2020) suggests that users interact more with individuals who share similar ideologies to them. To leverage this, Darwish, Magdy, and Zanouda (2017) calculated the similarity between users based on interactive elements such as user mentions and retweets to classify the stances of new users and optimize content features using the relationship between users.

Semi-supervised and Unsupervised Methods. Semi-supervised and unsupervised stance detection methods do not require a full set of labeled users. The idea of language homogeneity between similar groups of people makes the user's relationship network important for label classification. Label propagation is widely used in user and tweet stance detection, based on the user following network (Barberá 2015), user retweet network (Borge-Holthoefer et al. 2015; Weber, Garimella, and Batayneh 2013), or tweet relationships (Dutta et al. 2022; Giasemidis et al. 2020). The model propagates and learns according to existing knowledge, and the quality and imbalance of the initial label set affect the model's effectiveness, which is also the current limitation of semi-supervised learning. STEM exploited interaction networks and used embeddings of similar and opposite stances to divide the speakers into stance-partitions (Pick et al. 2022). To judge the target's position in argument stance

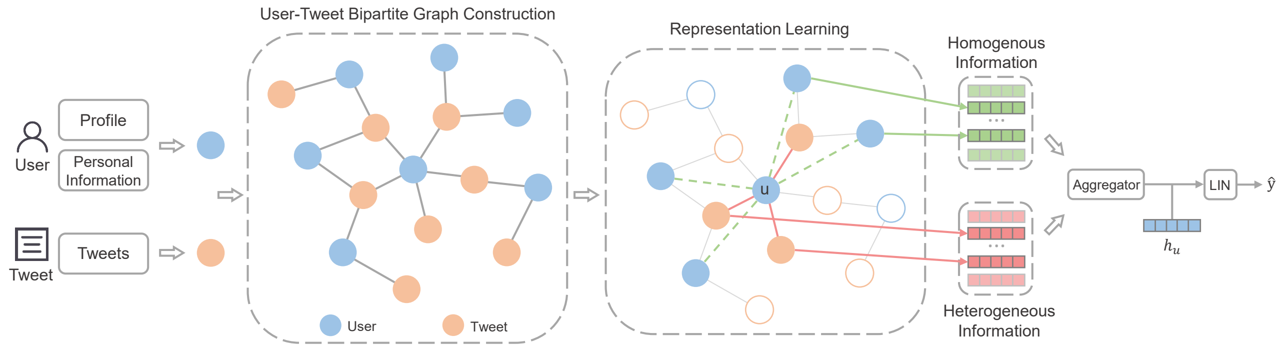

Figure 1: The framework of DoubleH. u represents the target node, h_u is the final representation of u.

detection, Kobbe, Hulpuș, and Stuckenschmidt (2020) extracts the effect triple from the target and statement of the target. In the task of user stance detection, (Darwish et al. 2020) projected users into a common feature space for clustering by user retweets, hashtags, and other information, which had a good performance but also has a high computational cost. To make better use of the user relation network, Retweet-BERT (Jiang, Ren, and Ferrara 2022) trained similar users according to the idea of homogeneity and employ the large-scale training model, which achieved good results. At the same time, it emphasized the importance of graphs in social network research.

Graph Neural Networks

Graph neural networks have received growing attention recently (Cai, Zheng, and Chang 2018; Bruna et al. 2014; Henaff, Bruna, and LeCun 2015; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Chen, Wu, and Zaki 2020). In pioneering work, Kipf and Welling (2017) proposed a simplified graph neural network model, called graph convolutional networks (GCN), and achieved state-of-the-art classification results on several benchmark graph datasets. Li and Goldwasser (2019) employed GCN on the social information graph which has two types of users and tweets to capture the documents' social context. Liang et al. (2021) used GCN blocks to derive the target-adaptive graph representation of the context for stance detection. Graph Attention Network (GAT) introduced the multi-head attention mechanism to the GNN architecture and dynamically learn the weights on the edges when performing message passing (Veličković et al. 2017). In addition, the graph neural networks with neighbor sampling and aggregation as the process (Hamilton, Ying, and Leskovec 2017; Ying et al. 2018) have achieved good development in terms of efficiency and effect. Some studies are dedicated to discovering different aggregation functions, so that information can be better disseminated and interacted with. SGC (Wu et al. 2019) and SSGC (Zhu and Koniusz 2021) proposed different spectral graph convolutions to make calculations faster. GIN (Xu et al. 2019), based on the Weisfeiler-Lehman Test, models the injective multiset function for aggregation with strong representation ability. Despite the success of graph neural networks in unsupervised and semi-supervised settings, few studies have applied them to supervised methods. Thus, we propose a novel bipartite graph-based graph neural network for supervised user stance detection.

Methodology

In this section, we will provide a detailed description of our model. Firstly, we will introduce the dataset and the data labeling method employed. Then, we will explain the construction of the user-tweet bipartite graph. Finally, we will illustrate how the model learns the representation of nodes by utilizing a homogeneous-heterogeneous joint information aggregation mechanism.

Dataset and Labeling

In the task of user stance detection, previous studies have mainly relied on user-related information or the relationship network between users, such as the following network and retweet network. However, we have noticed that existing research mostly utilizes only user-related information. In order to comprehensively analyze the user's stance, we have collected tweets between October 1, 2020 and November 2, 2020 using Twitter Streaming API. We then filtered the tweets related to the two election candidates, Joe Biden and Donald Trump, using specific keywords (*biden* or *trump*). Our final dataset contains a total of 138.9 million tweets in English, along with the corresponding metadata (e.g., the time of the tweet, the author, etc.) and user information (user ID, profile, location, etc.).

For such an enormous amount of data, manually labeling all data would be costly. We notice that the hashtags contained in a tweet are an explicit statement of position and are applied to detect user stances in previous studies (Conover et al. 2011; Bovet, Morone, and Makse 2018). First, we obtain the hashtags with high frequency in the data, and hashtags that clearly express a position are tagged by three experts based on the meaning of the hashtag, for example, $\#gojoe$ expresses support for Joe Biden. We interpret opposition to one candidate as support for the other (e.g., $\#sleepyjoe$ is considered pro-Trump). We then iteratively expand the hashtag set based on the co-occurrence relationship to discover new hashtags that are significantly related to the initial set. We evaluate the hashtags via hashtag co-occurrence networks (Zhou et al. 2021; Bovet, Morone, and

Figure 2: Hashtag clouds. The words or multi-word phrases in the clouds are the most frequently occurring hashtags related to the 2020 US presidential election. Blue hashtags indicate support for Biden or against Trump (pro-Biden), while the hashtags in red represent support for Trump or opposition to Biden (pro-Trump). The size of the hashtags is proportional to their frequency.

Makse 2018) to ensure consistency in hashtag classification. Figure 2 shows a subset of these hashtags, selected based on their frequency in the dataset. As a result, the expanded hashtag set of 282 hashtags is leveraged to label tweets, in which 85 of the set are pro-Trump, and the rest are pro-Biden. We select tweets that not only contain at least one hashtag but all hashtags in the tweet support the same candidate and then label the tweets according to the supported candidates. We obtain 1,123,749 labeled tweets, 724,057 of which were pro-Biden and 399,692 were pro-Trump. These tweets will be utilized to obtain users stance labels.

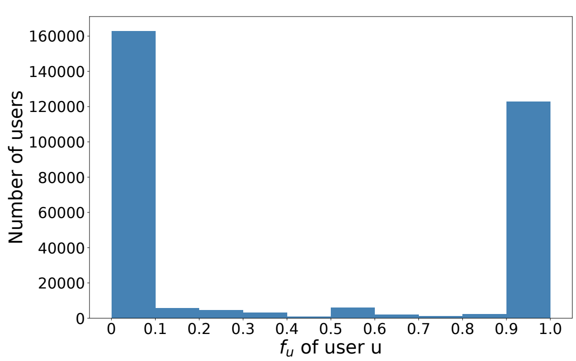

Figure 3: The distribution of f_u. The horizontal axis represents the interval of the relative frequency f_u of the tweet's positive examples of user u. The vertical axis is the number of users whose f_u is in a certain interval.

Labeling user data based on their profile and personal information can be challenging. Firstly, about 18.01% of users have no profile, which makes it difficult to infer their stance from personal details like their username or location. Additionally, many users with profiles do not explicitly express their position. However, tweets and retweets can serve as a valuable source of information to identify users' views. To label the stance of a user u, we leverage their tweets and retweets, denoted as $T_u = t_1^u, t_2^u, \cdots .t_n^u$ and their corresponding labels $L_t = L_{t_1}, L_{t_2}, \cdots, L_{t_n}$. The relative frequency f_u of positive tweet examples for user u is:

$$f_u = \frac{|\{L|L \in L_t \,\&\, L = 1\}|}{|L_t|}. \tag{1}$$

We create a frequency distribution $D(f_u)$ for all users based on their relative frequency f_u. Figure 3 shows the frequency distribution histogram. It can be observed that $D(f_u)$ follows a roughly U-shaped distribution, and many users have f_u values of 0 or 1, indicating that their related tweets are mostly of the same label. For users with other f_u values, we need to set a threshold to determine their stance. Therefore, the label L_u of user u is:

$$L_t = \begin{cases} 1 & f_u \geq k_u \\ 0 & f_u \leq k_u \end{cases} \tag{2}$$

where k_u is the threshold for user labels according to f_u. We set the value of k_u to 0.5 based on $D(f_u)$ and get 177415 users who support Joe Biden and 134203 users who hold Donald Trump. We remove duplication in tweets and users and exclude hashtags used for annotation from all text. As a result, all labeled users and associated tweets make up our dataset. The statistics for the dataset are shown in Table 1.

N_u	N_t	R_{re}	Avg. L_t	Avg. W_t
311,618	1,123,749	0.57	159.09	24.21

Table 1: The static information of the dataset. N_u is the number of users. N_t is the number of tweets. R_{re} is the proportion of the number of ‘retweets‘ to all tweets. L_t represents the length of tweet. W_t indicates the number of words in a tweet.

Bipartite Graph Construction

Existing studies on user stance detection are based on the assumption of linguistic homogeneity among people with similar stances (Yang and Eisenstein 2017; Kovacs and Kleinbaum 2020). Various graphs, such as retweet graphs, following graphs, etc., have been used to detect stances based on the relationship between users. In such methods, user profiles are used to learn user representations, and stances are detected based on known user tags, personal information, and user relation network information. However, in real-world data, many users lack detailed profiles, and the profiles may not provide clear information about their stances. This makes it challenging to accurately detect stances based solely on user profiles. To overcome this limitation, we utilize user-related tweets and user information for user stance detection, which can provide more informative data about the users' stances.

Taking user set U and tweet set T, we get the post and retweet relation between users and tweets. $E_p = \{< u, t >, < t, u > | u \in U, t \in T, u\, post\, t\}$ is the directed post relation edges between users and tweets, $E_r = \{< u, t >, < t, u > | u \in U, t \in T, u\, repost\, t\}$ represents the directed

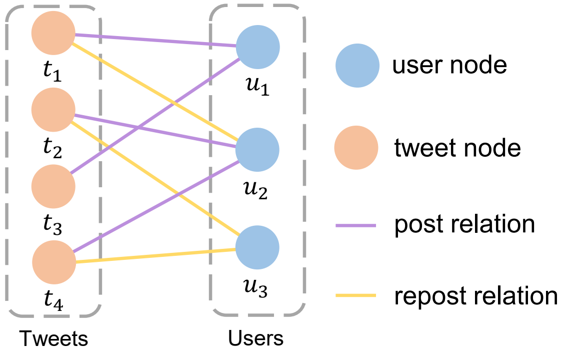

Figure 4: An example of the user-tweet bipartite graph.

retweet relation edges between users and tweets. The user-tweet directed bipartite graph $G = (V, E)$ is constructed by:

$$V = U \cup T, \tag{3}$$

$$E = E_p \cup E_r, \tag{4}$$

where V and E are the node set and edge set of the graph respectively. As illustrated in Figure 4, all user nodes are exclusively connected to tweet nodes, where retweet nodes function as a bridge that connects users to each other. We represent the notion of ideological homogeneity through a directed bipartite graph, where if a user retweets a tweet from another user, it indicates that the two users share similar stances. In comparison with a user's personal information, tweets provide more information and explicitly express the user's position. Therefore, we incorporate tweet information into the graph while maintaining user relationships. This approach facilitates the model's ability to learn superior user node representations by utilizing tweet information.

DoubleH

Preprocessing. To process scattered user information, we use templates to organize various information about users. For instance, to handle location information, we add the phrase "My location is" before the user's location information, and then insert this sentence into the user profile. Similar templates are used to integrate other information (e.g., the name of the user) into a complete user profile. In the user data, 35.2% of the users do not have location information, so for the missing information, we do not add templates with corresponding information into the user configuration files. Limited and variable initial user information is also one of the challenges of user stance detection. As for tweet data, we remove non-informative pieces of information such as links and emoticons. Since we label data by hashtags, all relevant hashtags are removed from user and tweet data. To obtain sentence-level embeddings for users and tweets, we use Sentence Transformers (S-BERT) (Reimers and Gurevych 2019), a Siamese network that outperforms other transformer-based methods for sentence-based tasks and significantly reduces time complexity. We utilize a pre-trained S-BERT model to directly retrieve integrated user profiles and tweets, using the S-BERT output as the initial representation for users and tweets.

Algorithm 1: DoubleH mini-batch forward propagation algorithm.

Input: Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; Input features $\{x_v, \forall v \in \mathcal{B}\}$; Depth K; Weight matrices $W^k, W_E^k, \forall k \in \{1, \cdots, K\}$; Non-linearity σ; Layernorm $\mathcal{L}$; Dropout $\mathcal{D}$; Neighbour sampler functions $\mathcal{N}_1 : u \to 2^u$ and $\mathcal{N}_2 : v \to 2^v$; Aggregate functions $AGGREGATE_k, \forall k \in \{1, \cdots, K\}$.
Output: Representations z_v for all $v \in \mathcal{B}$.

```
B^k ← B;
for k = K, ..., 1 do
    B^{k-1} ← B^k;
    for u ∈ B^k do
        B^{k-1} ← B^{k-1} ∪ N_1(u);
        for v ∈ N_1(u) do
            B^{k-1} ← B^{k-1} ∪ N_2(v);
        end for
    end for
end for
h_u^0 ← x_v, ∀v ∈ B^0;
for k = 1, ..., K do
    for u ∈ B^k do
        h_{N_1}^k ← σ(W_{E_1}^k D(L(h_{u'}^{k-1}, ∀u' ∈ N_1(u))));
        h_{N_2}^k ← σ(W_{E_2}^k D(L(h_{u'}^{k-1}, ∀u' ∈ N_2(N_1(u)))));
        h_{N(u)}^k ← AGGREGATE_k(h_{N_1}^k ∪ h_{N_2}^k);
        h_u^k ← σ(W^k · CONCAT(h_u^{k-1}, h_{N(u)}^k);
        h_u^k ← h_u^k / ||h_u^k||_2;
    end for
end for
```

Representation Learning. According to the user-tweet bipartite graph, we aim to learn the user node representation with user and tweet information. Specifically, following the message passing mechanism in GNNs, we intend to iteratively update the node representation of the bipartite graph with both user and tweet nodes at each layer. In the bipartite graph, the one-hop neighbors of each node are different node types from the center node, and the two-hop neighbors are of the same node type as the center node. For user nodes, the information of other users is homogeneous, while the information of tweets is heterogeneous. In our model, we consider homogeneous and heterogeneous information separately. The update process of each node embedding is described in Algorithm 1.

Since we exploit the homogeneous and heterogeneous information of each node, in each iteration, we need to collect the one-hop and two-hop neighbors of the node. In order to improve the training efficiency, we use the mini-batch and the method of neighbor sampling to generate neighbors. First, for each node in the batch, we sample its one-hop neighbors by $\mathcal{N}_1$, and then another neighbor sampler $\mathcal{N}_2$ samples the neighbors of the one-hop neighbors. $\mathcal{N}_1(v)$and $\mathcal{N}_2(v)$ are both fixed-size, uniform draw from the set $\{u \in \mathcal{V} : (u, v) \in \mathcal{E}\}$.

For user nodes, all one-hop neighbors are tweet nodes, and all two-hop neighbors are user nodes. Tweets are direct expressions of the user's position, so tweet information can help update the user's node representation. In addition, the two-hop neighbors of user nodes provide network information on retweet relationships between users. In other words, homogeneous information provides network relationships, and heterogeneous information adds additional information to nodes. Therefore, we intend to treat these two types of information separately. First, we process heterogeneous information $\{h_{u'}^{k-1}, \forall u' \in \mathcal{N}_1(u)\}$ into $h_{\mathcal{N}_1}^k$ by layernorm $\mathcal{L}$, linear weight matrices W_E^k and activation function σ. Similarly, homogeneous information $\{h_{u'}^{k-1}, \forall u' \in \mathcal{N}_2(\mathcal{N}_1(u))\}$ is processed into $h_{\mathcal{N}_2}^k$. To accurately utilize different types of edges in the bipartite graph, we use a type linear layer W_E^k. In each iteration, $W_{E_1}^k$ and $W_{E_2}^k$ the algorithm 1 are not the same for different node types. For heterogeneous neighbor information $h_{\mathcal{N}_1}^k$ and homogeneous neighbor information $h_{\mathcal{N}_2}^k$, we aggregate them into a neighborhood vector $h_{\mathcal{N}(u)}^k$ by aggregation function. We concatenate the node's previous layer representation h_u^{k-1} with the aggregated neighborhood vector $h_{\mathcal{N}(u)}^k$. The representation of nodes is finally updated by the linear layer, nonlinear layer, and normalization layer.

After K iterations, the model updates the representations of all nodes. The predicted label of the original user is computed as:

$$y_u = softmax(Wh_u^K + b), \tag{5}$$

where h_u^K is the representation vector of nodes of the final layer, and W and b are the weight and bias respectively. The goal of training is to minimize the cross-entropy between the ground truth label and the predicted label:

$$loss = -\sum_u g_u \log(y_u), \tag{6}$$

where g_u is the one-hot vector of the ground truth label.

Experiments

In this section, we evaluate the effectiveness of our model[1] and report the experimental results.

Baselines. In this paper, we focus on addressing the utilization of user-tweet bipartite graphs in user stance detection, so we mainly choose the graph-based methods as our baselines. We divide the baseline models into two categories: (i) Homogeneous graph methods, including GCN (Kipf and Welling 2017), GAT (Veličković et al. 2017), GraphSAGE (Hamilton, Ying, and Leskovec 2017), PinSage (Ying et al. 2018), GIN (Xu et al. 2019), Retweet-BERT (Jiang, Ren, and Ferrara 2022) and GPPT (Busbridge et al. 2019a). Retweet-BERT is a graph neural network method for user ideology detection. It employs the user's retweet graph and user profile information, combined with the large-scale pre-trained model to classify the ideology of users. GPPT is a recently proposed graph neural network model based on pre-training and prompt Tuning. The model reduces the training objective gap of the model by pre-training, prompt tuning functions, and fine-tuning. It has achieved the most advanced performance on graph-related tasks. (ii) Heterogeneous graph method: including RGAT (Busbridge et al. 2019b), RGCN (Schlichtkrull et al. 2018), HGT (Hu et al. 2020) and L-BGNN (Xie et al. 2022). HGT is a transformer-based model designed for heterogeneous graphs, employing node and edge type related parameters to capture heterogeneous attention on different edges. L-BGNN aggregates information in a bipartite graph by employing inter-domain message passing and intra-domain alignment.

Setup. We randomly split the dataset and use 80% of the instances for training, 10% for validation, and 10% for testing. For our model and other baselines, we use the Adam optimizer with a learning rate of 1e-3. The dropout is 0.1. The hidden size of the model is 768, and the batch size is 1024. The layer of the model is in [1, 2, 3]. We set the sample size of the neighbor sampling function $\mathcal{N}_1$ to 10, and the sample size of the neighbor sampling function $\mathcal{N}_2$ is also 10. For models with an aggregator, we set the aggregation function to sum. For GraphSage, pinsage and DoubleH, we use the same neighbor sampler with 10 as sample size when collecting first-order neighbors, and other baseline models collect all neighbors according to the description of the corresponding model. For DoubleH and all baselines, we use the BERT$_{base}$ model as the pre-training model for S-BERT and run all models on same graph. To ensure the reliability and robustness of the results, we train all models for 20 epochs on the same device (One V100 for GPU computing) and repeat the experiment 5 times and report the average performance.

Experimental Results

Table 2 presents the performance of DoubleH and baselines, and we can observe that DoubleH generally outperforms other baselines. Among the homogeneous graph approaches, where user and tweet nodes are treated as the same node type, GraphSAGE achieves the best performance, second only to DoubleH in all three metrics. In addition, GIN and GPPT also achieve competitive performance. However, GCN and GAT have mediocre performance. Pinsage is a graph neural network that is primarily used in recommendation systems, which finds important neighbors of the target node through random walks and aggregates these important neighbor nodes during information aggregation. In the recommendation system, the weight gap between users and items is apparent, so it is relevant for nodes to find important neighbors. However, in our dataset, each tweet of a user is an expression of a stance, and tweets with low side weights may have a more obvious stance bias. Therefore, the performance of Pinsage is not as prominent as other methods. Retweet-BERT utilizes the retweet network of users to reduce the similarity between neighboring users as the goal, and the pre-training is grounded on the profiles of users. The user's ideology is obtained by the pre-trained representation. We extract the retweet subnetwork of the data and input it into the Retweet-BERT model for pre-training. There are two reasons for the deficient performance of the model. First, the model's design is based on the assumption of linguistic

[1]The code of DoubleH can be found at https://github.com/Daisean/DoubleH.

Model	Feature				Performance		
	User Info.	Tweet Info.	Relation	Bipartite Info.	Acc.	Auc	F1
GCN	✓	✓			0.7595±0.04	0.7507±0.04	0.7109±0.05
GAT	✓	✓			0.7719±0.05	0.7639±0.06	0.7272±0.06
PinSage	✓	✓			0.7241±0.07	0.7078±0.07	0.6794±0.06
GraphSAGE	✓	✓			0.8426±0.06	0.8369±0.07	0.8133±0.06
GIN	✓	✓			0.8331±0.06	0.8280±0.04	0.8032±0.05
Retweet-BERT	✓				0.6820±0.08	0.6571±0.04	0.5612±0.07
GPPT	✓	✓			0.8042±0.04	0.7978±0.05	0.7723±0.04
RGCN	✓	✓	✓		0.7607±0.05	0.7516±0.04	0.7117±0.04
RGAT	✓	✓	✓		0.7832±0.07	0.7734±0.05	0.7396±0.06
HGT	✓	✓	✓		0.8352±0.06	0.8308±0.06	0.8068±0.05
L-BGNN	✓	✓		✓	0.8235±0.04	0.8174±0.08	0.7941±0.09
DoubleH	✓	✓	✓	✓	**0.8579±0.04**	**0.8529±0.03**	**0.8320±0.05**

Table 2: Performance of different models on the dataset. We run five trials with different random seeds and report the mean and standard deviation of the test set results. Bold indicates the top-performing method.

homogeneity and relies heavily on the user's network relationship. Yet, in reality, not all users who have a stance on a particular topic retweet relevant tweets. Moreover, some tweets are not retweeted by other users, resulting in more than half of the users being absent from the retweet network. As a result, Retweet-BERT is not pre-trained with all users, limiting its performance. Second, as we discussed in Section 4.1, some users in our dataset lack a profile, and the quality of the profiles for the remaining users varies. Therefore, relying solely on user information is insufficient for detecting stance, and incorporating user-related tweets can significantly enhance the model's performance. For heterogeneous graph methods, RGCN and RGAT perform comparably to GCN and GAT respectively, and slightly outperforms the corresponding homogeneous model. HGT and L-BGNN achieve better performance in heterogeneous graph models. Heterogeneous graph neural network models utilize heterogeneous attribute information of different nodes and edges to update node representations. For the user-tweet bipartite graph, although user nodes and tweet nodes are different types of nodes, they are both texts, and there is no obvious difference in attributes. Therefore, better utilization of different relationships between nodes is the main reason for the improved performance.

In traditional graph neural network models, each node only interacts directly with its one-hop neighbors, while information from two-hop neighbors is transmitted to the target node through one-hop neighbors in the next layer. However, in the user-tweet bipartite graph, a node's interaction with its one-hop neighbors represents the information exchange between the user and the tweet, while interaction with its two-hop neighbors represents the information exchange between the user and other users. DoubleH improves upon this by allowing each user node to interact directly with its two-hop neighbors, thereby enabling it to obtain more accurate information than other models. Compared to the homogeneous graph method, our approach leverages a richer set of relationship information among heterogeneous nodes, along with the bipartite graph details involving homogeneous nodes. In contrast to recent techniques in the field of heterogeneous graph analysis, we enhance the model's incorporation of bipartite graph information by concurrently processing both homogeneous and heterogeneous data sources. As a result, DoubleH achieves state-of-the-art performance, outperforming all baseline models across all metrics. In the next section, we will conduct further analyses on the ablation of information utilization, the models' layer structures, and the efficiency of DoubleH.

Ablation Study

DoubleH achieves good performance by effectively aggregating homogeneous and heterogeneous information. To further understand the role of these two types of information, we performed ablation learning on one-layer and two-layer DoubleH with only one of the information for aggregation each time. The results are shown in Table 3.

Model	Hete Info.	Homo Info.	F1
1-Layer DoubleH	✗	✓	0.6796
	✓	✗	0.8129
	✓	✓	0.8243
2-Layer DoubleH	✗	✓	0.7427
	✓	✗	0.8234
	✓	✓	0.8305

Table 3: Ablation learning of heterogeneous and homogeneous information on DoubleH with 1 and 2 layers.

The results show that when only 1-hop neighbor information (heterogeneous information) is utilized in aggregation, the model's performance will not degrade too much, whereas using only 2-hop neighbor information (homogeneous information) leads to a significant drop in performance. For homogeneous information, if only homogeneous information is utilized for each aggregation, the model directly ignores its one-hop neighbors, and user nodes only interact with user nodes. As previously stated, the personal information of numerous users is incomplete, resulting in disparities in the feature quality across user nodes. Conversely, tweet information maintains comprehensive semantics. Hence, if the model exclusively relies on user node information, it cannot achieve optimal performance. Only retweet network information is exploited in aggregation, causing DoubleH to behave similarly to Retweet-BERT. On the other hand, heterogeneous information exhibits a more important role. The results in Table 3 show that 1-layer and 2-layer DoubleH perform comparably to GraphSAGE and GIN when only using heterogeneous information. Among the existing graph neural network models, messages are transmitted via direct neighbors, and nodes can only interact with their one-hop neighbors at a time. DoubleH extracts and aggregates homogeneous information and heterogeneous information respectively. The 1-hop and 2-hop neighbors become the 'new one-hop neighbors' of the central node, thereby breaking the traditional node interaction mechanism. For the user stance detection task, this novel mechanism can balance and utilize the tweet information and network information of the bipartite graph, thus enabling the model to achieve better node representation. Furthermore, with the advent of more diverse graph data structures, this mechanism can also be extended to interact with more hops.

Model Layer Number

In the graph neural network methods, the number of layers k of the model represents that each node can fuse k-hop neighbor information. Table 4 shows the performance of some baseline models with different layer numbers.

We can observe that the performance of GCN, GAT, and RGCN degrades when the number of layers is increased to 2. The performance of the 1-layer GIN model is slightly inferior to that of GCN and GAT, while GIN with 2 layers shows stable performance. On the other hand, the performance of the GraphSAGE model improves as the number of layers increases, with the 3-layer GraphSAGE model being the best among all baselines, comparable to the performance of the 2-layer GraphSAGE. The Heterogeneous Graph Transformer (HGT) empowers nodes to effectively leverage diverse edge information via its distinctive heterogeneous mutual attention mechanism. As a result, even at layer 1, HGT has already demonstrated remarkably competitive performance. From the performance of the baseline models, we can see that the 1-layer models do not perform well because each user node only integrates the heterogeneous information of its direct neighbors, losing the homogeneous information of its neighbors from the second hop. When the number of layers of the model increases, stronger models show the ability to fuse multi-hop information. DoubleH allows nodes to directly interact with one-hop neighbors and two-hop neighbors when fusing information, which is why DoubleH with one layer outperforms all baseline models. The 2-layer and 3-layer DoubleH models show even better performance.

Model	Layer Number	Acc.	F1
GCN	1	<u>0.7595</u>	<u>0.7109</u>
	2	0.7466	0.6953
GAT	1	<u>0.7719</u>	<u>0.7272</u>
	2	0.7444	0.7065
RGCN	1	<u>0.7607</u>	<u>0.7117</u>
	2	0.7414	0.6933
RGAT	1	0.7721	0.7294
	2	<u>0.7832</u>	<u>0.7396</u>
GraphSAGE	1	0.7654	0.7095
	2	0.8385	0.8092
	3	<u>0.8426</u>	<u>0.8133</u>
GIN	1	0.7444	0.6894
	2	0.8322	0.8018
	3	<u>0.8331</u>	<u>0.8032</u>
HGT	1	0.8351	0.8064
	2	<u>0.8352</u>	<u>0.8068</u>
	3	0.8328	0.8033
DoubleH	1	0.8512	0.8243
	2	0.8564	0.8305
	3	<u>**0.8579**</u>	<u>**0.8320**</u>

Table 4: Performance of models with different layers. Bold indicates the best performance, and underlined text refers to the best performance of the same model at different layers.

Efficiency Analysis

Each additional layer of the model will increase the computational complexity of the model, thereby increasing the training time of the model. Among our baselines, the Retweet-BERT model is inductive, and the other models are transductive. The transductive models can achieve better performance but need to retrain when encountering new incoming data in practical applications. For the stance detection task of social media users, the accuracy rate of the model, F1 value, and other metrics have received too much attention. However, the model should serve practical problems, and the best performance model with a large amount of calculation is difficult to be applied. Therefore, the efficiency of the model deserves to be taken into consideration. We take the training time as the efficiency of the transductive model. The shorter the training time, the higher the efficiency of the model. We record the efficiency of the model with the F1 value above 0.75, Figure 5 shows the efficiency-performance graph for different baselines. From the figure, we noticed that the 2-layer DoubleH has a good performance with lower efficiency than GraphSAGE, GIN and HGT. The overall performance of GraphSAGE is better than that of HGT and GIN. The 3-layer DoubleH with the best F1 value needs the longest training time among all models, so it is not recommended to be employed for practical problems. The 1-layer DoubleH has both good performance and efficiency, becoming the model with the best overall performance.

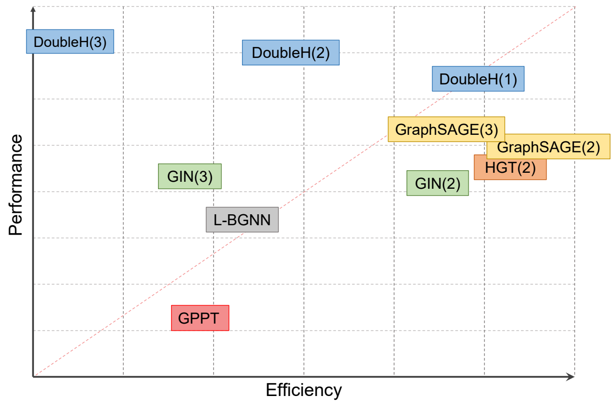

Figure 5: The efficiency-performance graph of baselines. The horizontal axis is the model, and the vertical axis is the performance of the model based on the F1 value.

In addition, to intuitively understand the scale of the model, we counted the parameters of some models, and the results are shown in Table 5. Notably, due to the nature of the heterogeneous graph method, which necessitates consideration of node relationships and the aggregation of information based on diverse edge types, the total parameter count exceeds that of the homogeneous graph model. Taking into account the model's performance, it becomes evident that the 1-layer DoubleH surpasses all baseline models in terms of performance, and remarkably, it operates with a relatively modest parameter count. As the DoubleH model is expanded to include additional layers, its parameter count naturally increases, resulting in performance improvement. In general, there isn't a significant disparity in the parameter count among all the models. In contrast to heterogeneous graph techniques, the proposed model manages to deliver superior performance while utilizing a smaller parameter count. Furthermore, the parameter count remains within the same order of magnitude as homogeneous graph methods, yet attains state-of-the-art performance levels.

Conclusion

In this paper, we propose a novel model for user stance detection based on the user-tweet bipartite graph. Firstly, we collect social media data related to the 2020 US presidential election and annotate the tweets with manually tagged hashtags, and label the user's stance based on the ratio of related tweets. Our model leverages both homogeneous and heterogeneous information of each node to learn node representations. DoubleH enables each node to interact directly with its one-hop and two-hop neighbors at the structural level and handles homogeneous and heterogeneous information separately at the information level. Experimental results demonstrate the effectiveness of DoubleH in utilizing both homogeneous and heterogeneous information, achieving state-of-the-art performance while maintaining high efficiency and

Model	Layer Num.	Parameter Num.	F1
Graphsage	3	4.14M	0.8133
GIN	3	1.18M	0.8032
GPPT	2	1.21M	0.7723
RGAT	2	2.37M	0.7396
HGT	2	11.8M	0.8068
DoubleH	1	5.31M	0.8243
DoubleH	2	11.8M	0.8305
DoubleH	3	18.29M	0.8320

Table 5: The model performance with the specific number of layers and parameters. Note that for baseline models we only show the number of layers and parameters at their best performance.

reducing the number of layers.

DoubleH performs information interaction on the entire user-tweet bipartite graph. However, for new incoming data, the model needs to rebuild the graph and retrain, making the model less flexible and not well-suited for transductive learning. This is a limitation shared by all transductive models. In future work, we will address this issue and improve our model's scalability and flexibility.

Potential Broader Impact and Ethical Considerations

In this article, we use Twitter as a case study, which is one of the most influential social media platforms globally (Salloum et al. 2017), to showcase DoubleH's analytical capabilities. Our model is a versatile social network analysis tool that can analyze the attributes of any pair of elements with a binary relationship, making it widely applicable. For instance, it can be used for analyzing other events with clear stances, such as the upcoming 2024 US presidential election. Moreover, our model can leverage more detailed user information to enhance the information of user nodes for further optimization. However, we want to emphasize that our research should not be utilized for inappropriate surveillance and malicious harassment of users based on their stances.

The main goal of our research is to leverage social network information for analyzing users' stances. To achieve this, we collect user and tweet information from Twitter and construct a user-tweet bipartite graph. Since we analyze the stances of all users in our data set regarding the 2020 US general election, the data set contains users' views and stances on this topic. However, the privacy of the data set must be protected to prevent the disclosure of personal information of specific users and avoid any improper surveillance that may support a certain position. Therefore, we will take measures to carefully consider the disclosure of the data set, such as checking the user's relevant information and re-representing tweets so that their authors cannot be re-identified. We would like to emphasize again that our research does not support the use of relevant models and data for inappropriate surveillance purposes, the misclassified data will only be used to optimize the model. Furthermore, we prioritize ethical considerations and hold a profound respect for individuals and their fundamental rights. In our engagement with research participants, we have proactively communicated the research's objectives, ensuring a transparent understanding. Throughout the research process, we are committed to safeguarding privacy and refraining from any breaches. The acquisition of information is conducted through lawful channels, and we meticulously adhere to all principles of research ethics. In addition, we store the data through an internal server and only send the hashtag list to the annotators, so there will be no leakage of user information.

Acknowledgements

This study was supported by the National Natural Science Foundation of China under projects No. 62302319 and R&D Program of Beijing Municipal Education Commission (Grant No. KM202210038002).

References

Augenstein, I.; Rocktäschel, T.; Vlachos, A.; and Bontcheva, K. 2016. Stance Detection with Bidirectional Conditional Encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 876–885. Austin, Texas: Association for Computational Linguistics.

Barberá, P. 2015. Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data. *Political Analysis*, 23(1): 76–91.

Borge-Holthoefer, J.; Magdy, W.; Darwish, K.; and Weber, I. 2015. Content and Network Dynamics Behind Egyptian Political Polarization on Twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, 700–711. New York, NY, USA: Association for Computing Machinery. ISBN 9781450329224.

Bovet, A.; and Makse, H. A. 2019. Influence of fake news in Twitter during the 2016 US presidential election. *Nature communications*, 10(1): 1–14.

Bovet, A.; Morone, F.; and Makse, H. A. 2018. Validation of Twitter opinion trends with national polling aggregates: Hillary Clinton vs Donald Trump. *Scientific reports*, 8(1): 1–16.

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

Busbridge, D.; Sherburn, D.; Cavallo, P.; and Hammerla, N. Y. 2019a. Relational Graph Attention Networks. *CoRR*, abs/1904.05811.

Busbridge, D.; Sherburn, D.; Cavallo, P.; and Hammerla, N. Y. 2019b. Relational Graph Attention Networks. *CoRR*, abs/1904.05811.

Caetano, J. A.; Lima, H. S.; Santos, M. F.; and Marques-Neto, H. T. 2018. Using sentiment analysis to define twitter political users' classes and their homophily during the 2016 American presidential election. *Journal of internet services and applications*, 9(1): 1–15.

Cai, H.; Zheng, V. W.; and Chang, K. C.-C. 2018. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9): 1616–1637.

Chen, Y.; Wu, L.; and Zaki, M. J. 2020. Reinforcement Learning Based Graph-to-Sequence Model for Natural Question Generation. In *Proceedings of the 8th International Conference on Learning Representations*.

Conover, M. D.; Goncalves, B.; Ratkiewicz, J.; Flammini, A.; and Menczer, F. 2011. Predicting the Political Alignment of Twitter Users. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 192–199.

Darwish, K.; Magdy, W.; and Zanouda, T. 2017. Improved Stance Prediction in a User Similarity Feature Space. ASONAM '17, 145–148. New York, NY, USA: Association for Computing Machinery. ISBN 9781450349932.

Darwish, K.; Stefanov, P.; Aupetit, M.; and Nakov, P. 2020. Unsupervised User Stance Detection on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1): 141–152.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Dutta, S.; Caur, S.; Chakrabarti, S.; and Chakraborty, T. 2022. Semi-Supervised Stance Detection of Tweets Via Distant Network Supervision. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, 241–251. New York, NY, USA: Association for Computing Machinery. ISBN 9781450391320.

Gayo-Avello, D.; Jungherr, A.; and Rivero, G. 2023. Social media and electoral prediction: Ten years after. *Handbook of computational social science. Edward Elgar Publishing*.

Giasemidis, G.; Kaplis, N.; Agrafiotis, I.; and Nurse, J. R. C. 2020. A Semi-Supervised Approach to Message Stance Classification. *IEEE Transactions on Knowledge and Data Engineering*, 32(1): 1–11.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *NIPS*, 1024–1034.

Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep Convolutional Networks on Graph-Structured Data.

Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, 2704–2710.

Jiang, J.; Ren, X.; and Ferrara, E. 2022. Retweet-BERT: Political Leaning Detection Using Language Features and Information Diffusion on Social Networks.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR*

2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

Kobbe, J.; Hulpuș, I.; and Stuckenschmidt, H. 2020. Unsupervised stance detection for arguments from consequences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 50–60. Online: Association for Computational Linguistics.

Kovacs, B.; and Kleinbaum, A. M. 2020. Language-Style Similarity and Social Networks. *Psychological Science*, 31(2): 202–213. PMID: 31877069.

Li, C.; and Goldwasser, D. 2019. Encoding Social Information with Graph Convolutional Networks forPolitical Perspective Detection in News Media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2594–2604. Florence, Italy: Association for Computational Linguistics.

Liang, B.; Fu, Y.; Gui, L.; Yang, M.; Du, J.; He, Y.; and Xu, R. 2021. Target-Adaptive Graph for Cross-Target Stance Detection. In *Proceedings of the Web Conference 2021*, WWW '21, 3453–3464. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383127.

Magdy, W.; Darwish, K.; Abokhodair, N.; Rahimi, A.; and Baldwin, T. 2016. #ISISisNotIslam or #DeportAllMuslims? Predicting Unspoken Views. In *Proceedings of the 8th ACM Conference on Web Science*, WebSci '16, 95–106. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342087.

Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 31–41. San Diego, California: Association for Computational Linguistics.

Pennacchiotti, M.; and Popescu, A.-M. 2011. Democrats, Republicans and Starbucks Afficionados: User Classification in Twitter. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, 430–438. New York, NY, USA: Association for Computing Machinery. ISBN 9781450308137.

Pick, R. K.; Kozhukhov, V.; Vilenchik, D.; and Tsur, O. 2022. STEM: Unsupervised STructural EMbedding for Stance Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 11174–11182.

Preoţiuc-Pietro, D.; Liu, Y.; Hopkins, D.; and Ungar, L. 2017. Beyond Binary Labels: Political Ideology Prediction of Twitter Users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 729–740. Vancouver, Canada: Association for Computational Linguistics.

Rao, D.; Yarowsky, D.; Shreevats, A.; and Gupta, M. 2010. Classifying Latent User Attributes in Twitter. In *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents*, SMUC '10, 37–44. New York, NY, USA: Association for Computing Machinery. ISBN 9781450303866.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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)*, 3982–3992. Hong Kong, China: Association for Computational Linguistics.

Salloum, S. A.; Al-Emran, M.; Monem, A. A.; and Shaalan, K. 2017. A survey of text mining in social media: Facebook and Twitter perspectives. *Advances in Science, Technology and Engineering Systems Journal*, 2(1): 127–133.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling Relational Data with Graph Convolutional Networks. volume 10843 of *ESWC 18'*, 593–607. Cham: Springer International Publishing. ISBN 978-3-319-93417-4.

Tran, O.; Phung, A. C.; and Ngo, B. X. 2022. Using Convolution Neural Network with BERT for Stance Detection in Vietnamese. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 7220–7225. Marseille, France: European Language Resources Association.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2017. Graph Attention Networks. *6th International Conference on Learning Representations*.

Weber, I.; Garimella, V. R. K.; and Batayneh, A. 2013. Secular vs. Islamist Polarization in Egypt on Twitter. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, 290–297. New York, NY, USA: Association for Computing Machinery. ISBN 9781450322409.

Wong, F. M. F.; Tan, C. W.; Sen, S.; and Chiang, M. 2021. Quantifying Political Leaning from Tweets and Retweets. *Proceedings of the International AAAI Conference on Web and Social Media*, 7(1): 640–649.

Wu, F.; Zhang, T.; Jr., A. H. S.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. *CoRR*, abs/1902.07153.

Xie, T.; He, C.; Ren, X.; Shahabi, C.; and Kuo, C.-C. J. 2022. L-BGNN: Layerwise Trained Bipartite Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yang, Y.; and Eisenstein, J. 2017. Overcoming Language Variation in Sentiment Analysis with Social Attention. *Transactions of the Association for Computational Linguistics*, 5: 295–307.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, 974–983. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355520.

Zhou, Z.; Serafino, M.; Cohan, L.; Caldarelli, G.; and Makse, H. A. 2021. Why polls fail to predict elections. *Journal of Big Data*, 8(1): 1–28.

Zhu, H.; and Koniusz, P. 2021. Simple Spectral Graph Convolution. In *International Conference on Learning Representation 2021*.

Paper Checklist

1. For most authors...
   (a) Would answering this research question advance science without violating social contracts, such as violating privacy norms, perpetuating unfair profiling, exacerbating the socio-economic divide, or implying disrespect to societies or cultures? Yes.
   (b) Do your main claims in the abstract and introduction accurately reflect the paper's contributions and scope? Yes, see the second half of the abstract and the end of the Introduction section.
   (c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes, the Method section presents our claims in detail.
   (d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes.
   (e) Did you describe the limitations of your work? Yes, see the second paragraph in Conclusion section.
   (f) Did you discuss any potential negative societal impacts of your work? Yes, see our notes at the end of the first paragraph of this section.
   (g) Did you discuss any potential misuse of your work? Yes, see our elaboration in the first two paragraphs of this section.
   (h) Did you describe steps taken to prevent or mitigate potential negative outcomes of the research, such as data and model documentation, data anonymization, responsible release, access control, and the reproducibility of findings? Yes, see the second paragraph of this section.
   (i) Have you read the ethics review guidelines and ensured that your paper conforms to them? Yes.
2. Additionally, if your study involves hypotheses testing...
   (a) Did you clearly state the assumptions underlying all theoretical results? NA
   (b) Have you provided justifications for all theoretical results? NA
   (c) Did you discuss competing hypotheses or theories that might challenge or complement your theoretical results? NA
   (d) Have you considered alternative mechanisms or explanations that might account for the same outcomes observed in your study? NA
   (e) Did you address potential biases or limitations in your theoretical framework? NA
   (f) Have you related your theoretical results to the existing literature in social science? NA
   (g) Did you discuss the implications of your theoretical results for policy, practice, or further research in the social science domain? NA
3. Additionally, if you are including theoretical proofs...
   (a) Did you state the full set of assumptions of all theoretical results? NA
   (b) Did you include complete proofs of all theoretical results? NA
4. Additionally, if you ran machine learning experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? Yes, we plan to provide the URL of the code in the footnote on page 6. To maintain anonymity, we will make the code public after the paper is accepted.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes, see Setup in the Experiments section.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes, see Setup and Experimental Result in the Experiments section.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? Yes, see Setup in the Experiments section.
   (e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes, we have proven the effectiveness of the proposed method through comparative experiments with baseline, ablation learning, parameter analysis and efficiency analysis, see the Experiments section.
   (f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? Yes, see the second paragraph of this section.
5. Additionally, if you are using existing assets (e.g., code, data, models) or curating/releasing new assets, **without compromising anonymity**...
   (a) If your work uses existing assets, did you cite the creators? NA
   (b) Did you mention the license of the assets? NA
   (c) Did you include any new assets in the supplemental material or as a URL? NA
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? NA
   (f) If you are curating or releasing new datasets, did you discuss how you intend to make your datasets FAIR (see **?**)? NA
   (g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset (see **?**)? NA
6. Additionally, if you used crowdsourcing or conducted research with human subjects, **without compromising anonymity**...
   (a) Did you include the full text of instructions given to participants and screenshots? Yes.

(b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? Yes.
(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? Yes.
(d) Did you discuss how data is stored, shared, and deidentified? Yes, see the second paragraph of this section.