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

Online Social Behavior Enhanced Detection of Political Stances in Tweets

Xingyu Peng[1], Zhenkun Zhou[2*], Chong Zhang[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
xypeng@buaa.edu.cn, zhenkun@cueb.edu.cn, chongzh@buaa.edu.cn, kexu@buaa.edu.cn

Abstract

Public opinion plays a pivotal role in politics, influencing political leaders' decisions, shaping election outcomes, and impacting policy-making processes. In today's digital age, the abundance of political discourse available on social media platforms has become an invaluable resource for analyzing public opinion. This paper focuses on the task of detecting political stances in the context of the 2020 US presidential election. To facilitate this research, we curate a substantial dataset sourced from Twitter, annotated using hashtags as indicators of political polarity. In our approach, we construct a bipartite graph that explicitly models user-tweet interactions, which provides a comprehensive contextual understanding of the election. To effectively leverage the wealth of user behavioral information encoded in this graph, we adopt graph convolution and introduce a novel skip aggregation mechanism. This mechanism enables tweet nodes to aggregate information from their second-order neighbors, which are also tweet nodes due to the graph's bipartite nature. Our experimental results demonstrate that our proposed model outperforms a range of competitive baseline models. Furthermore, our in-depth analyses highlight the importance of user behavioral information and the effectiveness of skip aggregation.

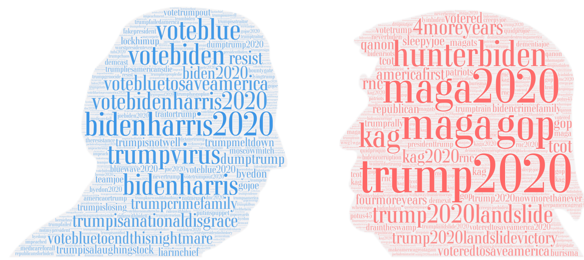


Figure 1: Hashtag clouds. The words or multi-word phrases in the clouds are the most frequently occurring hashtags related to the 2020 US presidential election. The hashtags in blue express support for Biden or opposition to Trump (pro-Biden), while the hashtags in red express support for Trump or opposition to Biden (pro-Trump). The dimension of the hashtags is proportional to their frequency.

Introduction

Nowadays, social media has emerged as an indispensable tool for the general public to engage in discussions on political issues, such as elections, taxes, education, and regulations. As was seen in the 2016 US presidential election (Bovet, Morone, and Makse 2018) and the 2019 Argentina presidential election (Zhou et al. 2021), social media provided a platform for candidates to connect with potential voters, share their policies, and establish their brand during the pre-election phase. Likewise, voters turned to social media to acquaint themselves with the candidates, their policies, and their positions on important issues, while also expressing their own views on the elections. As a result, such political events generated a significant volume of reviews and responses online, offering researchers a valuable avenue to capture public opinion and discern societal trends. However, manually analyzing such a massive amount of textual data is extremely time-consuming and costly, underscoring the pressing need for automated analysis.

Stance detection has evolved into a pivotal and dynamic area of research, extending beyond its traditional applications in customer reviews, product feedback, and movie preferences. In recent years, there has been a notable shift towards analyzing stances in a political context, as it provides valuable insights into public opinion. However, identifying political stances is inherently challenging, primarily due to the presence of nuanced language, ambiguous sentiment, and complex contextual factors. In this study, we specifically focus on the 2020 US presidential election, intending to ascertain the political polarities of election-related tweets posted during the pre-election period. Unlike news articles (Li and Goldwasser 2019, 2021b; Zhang et al. 2022), debate transcripts (Iyyer et al. 2014; Chen et al. 2017; Lai et al. 2020), and tweets from official sources like legislators, news agencies, and politicians (Xiao et al. 2023), which typically adhere to strict grammatical rules, analyzing tweets from ordinary users presents unique challenges. These challenges include the brevity of the text, the absence of contextual information, and the frequent use of emoticons, abbreviations, and hashtags.

Additionally, obtaining labels for social media data of such magnitude poses a formidable challenge. Given the sheer volume of data, traditional manual annotation meth-

*Corresponding author: Zhenkun Zhou.

ods like crowdsourcing and editorial review (Li and Goldwasser 2019) become impractical. Some studies (Xiao et al. 2023; Iyyer et al. 2014; Chen et al. 2017) have resorted to annotating texts based on the ideological positions of authors or speakers. However, for personal Twitter accounts, political ideologies are often undisclosed. Hashtags, on the other hand, are frequently used by Twitter users to provide context and convey their sentiments or opinions on specific topics or issues. For example, hashtags like *#EndGunViolence* advocate for stricter gun control measures and an end to gun violence in communities, while *#StopAsianHate* expresses support for ending anti-Asian hate and fostering a more inclusive and equitable society for all, regardless of their racial or ethnic background. Generally, hashtags align with the overall stance of the tweets in which they are included. Therefore, we adopt a manual approach to compile a set of hashtags expressing support or opposition to one of the candidates, categorizing them as either pro-Biden or pro-Trump. If a hashtag is found to oppose one candidate, we interpret it as supporting the other candidate (e.g., considering *#VoteTrumpOut* as pro-Biden). Subsequently, we annotate the tweets according to the polarities of the hashtags within them.

Our annotation approach enables the acquisition of a substantial set of labeled tweets, empowering the model to effectively tackle the challenges posed by informal language and comprehend the complex context of the election. Instead of relying solely on textual analysis to infer stances from these tweets, we propose leveraging the abundance of behavioral data available on social media platforms. Our approach involves constructing a user-tweet bipartite graph, where users and tweets are represented as nodes, and edges denote users' behaviors towards tweets, such as posting and retweeting. In this manner, we encode user behavioral information into the graph, transforming the task of political stance detection into a Graph Neural Network (GNN)-based node classification problem. While we initialize tweet node features using a pre-trained language model (e.g., BERT (Devlin et al. 2019)), we encounter a challenge when initializing the features of user nodes. This challenge stems from the absence of textual information for users, making it difficult to assign appropriate initial features that align with tweet nodes. To address this issue, we propose a novel skip aggregation mechanism that leverages the bipartite nature of the graph. Each tweet node aggregates information from its second-order neighbors, which comprise other tweets posted or retweeted by its author or retweeters and tend to express similar stances (Wong et al. 2016). User nodes, on the other hand, play a bridging role during neighborhood sampling and do not require explicit characterization. In addition, we incorporate relational information into the model to make more effective use of behavioral data, resulting in further performance improvements.

We conduct a comprehensive comparative analysis of our model against several state-of-the-art text classification and node classification models, using our annotated dataset. The results of our experiments consistently demonstrate the superior performance of our model, outperforming the competing models across various metrics. Our further analyses emphasize the significance of integrating user behavioral information into political stance detection and highlight the effectiveness of skip aggregation in leveraging this valuable information. Additionally, we provide evidence of our model's robustness in handling short texts through performance analysis. We also employ word cloud visualization to exhibit the prevalent language in tweets supporting or opposing each candidate. Our contributions can be summarized as follows:

- We collect a large amount of data from Twitter pertaining to the 2020 US presidential election, including tweets and associated metadata such as authors and retweeters, and automatically annotate the tweets using manually labeled hashtags.
- We propose a GNN-based framework for detecting political stances in tweets. Our framework adopts a novel skip aggregation mechanism to effectively learn behavior-aware tweet representations from a bipartite graph consisting of user-tweet interactions.
- We conduct an extensive series of experiments and analyses to demonstrate the importance of user behavioral information and the effectiveness of skip aggregation, both of which significantly contribute to the superior performance of our model over baseline models.

Related Work

Political stance detection falls within the domain of Natural Language Processing (NLP), involving the analysis of political texts to discern the author's position on specific political events. Previous research in this area has primarily focused on several key aspects.

Numerous studies have concentrated on measuring political leanings in tweets. For instance, Maynard and Funk (2011) integrated contextual information to extract political opinions from pre-election tweets. Lai et al. (2017) considered the context surrounding a target of interest to predict stances towards those mentioned targets. Chen et al. (2017) proposed the creation of an opinion-aware knowledge graph for inferring political ideologies. Kamal, Gullic, and Bagavathi (2022) investigated various text representation learning frameworks and supervised machine learning models to analyze the political leanings expressed in social media posts. Ng and Carley (2022) explored several crucial aspects related to cross-dataset model generalization in stance detection, providing practical guidance on effectively generalizing models to new data in real-world applications. Xiao et al. (2023) introduced a method to quantify political polarities in tweets by assigning polarity scores to entities and hashtags.

Research has also aimed to infer the political alignment of users on social media platforms. Wong et al. (2016) incorporated tweets and retweets, formulating the inference of Twitter users' political leanings as a convex optimization problem. Manickam et al. (2019) introduced a framework for jointly estimating the ideology of social media users and news websites. Darwish et al. (2020) developed an unsupervised framework for detecting the stance of prolific Twitter users on controversial topics, employing dimensionality reduction and clustering techniques. Xiao et al. (2020)

proposed a framework that involves a multi-relational encoder and a multi-task decoder for ideology detection on Twitter. Fagni and Cresci (2022) presented an unsupervised method for extracting fine-grained political leanings from social media posts. Jiang, Ren, and Ferrara (2023) leveraged the retweet network structure and the language used in users' profile descriptions to estimate the political leanings of social media users.

Some researchers have worked on predicting voting intentions or election outcomes. Lampos, Preoţiuc-Pietro, and Cohn (2013) analyzed tweets from the UK and Austria, successfully predicting voting intentions in numerous polls. Khatua, Khatua, and Cambria (2020) examined tweets mentioning multiple political parties to predict the political opinions of voters in the context of the 2014 Indian General Election. Zhou et al. (2021) proposed an opinion tracking method that leveraged machine learning models and social network big data analysis to achieve accurate results in the 2019 Argentina elections. Islam, Roy, and Goldwasser (2023) introduced a weakly supervised approach to identify the stance and issue of political advertisements on Facebook. Their study also delved into the temporal dynamics of these political advertisements in relation to election polls.

Another line of research has focused on detecting political perspectives in news articles. Li and Goldwasser (2019) proposed inferring political perspectives expressed in news articles by examining document dissemination patterns and the characteristics of endorsing users, using Graph Convolutional Networks (GCN). Li and Goldwasser (2021b) introduced a framework for pre-training models that leverages social and linguistic context. They also proposed an entity-centric framework (Li and Goldwasser 2021a) that incorporates entity and relation representations learned from external knowledge sources and text corpus, utilizing attention mechanisms to evaluate the importance of article aspects. Zhang et al. (2022) presented an approach that uses textual cues as paragraph-level labels and integrates multi-hop knowledge reasoning for inference.

At present, the integration of Large Language Models (LLMs) in stance detection is an active area of exploration with considerable promise. Experiments conducted by Zhang, Ding, and Jing (2022) revealed that ChatGPT exhibits impressive performance on commonly used datasets while providing explanations for its predictions. However, Aiyappa et al. (2023) noted improved stance detection with ChatGPT while expressing concerns about potential data contamination from its extensive training dataset, thereby casting doubts on the reliability of evaluations. On the other hand, Cruickshank and Ng (2023) argued that, although LLMs may not definitively outperform existing supervised methods, they offer adaptable stance detection outputs without the need for extensive human annotation. This adaptability opens up new avenues for the application of stance detection techniques across a diverse range of contexts.

Data

In this study, we focus on the 2020 United States presidential election that took place on November 3, 2020. We

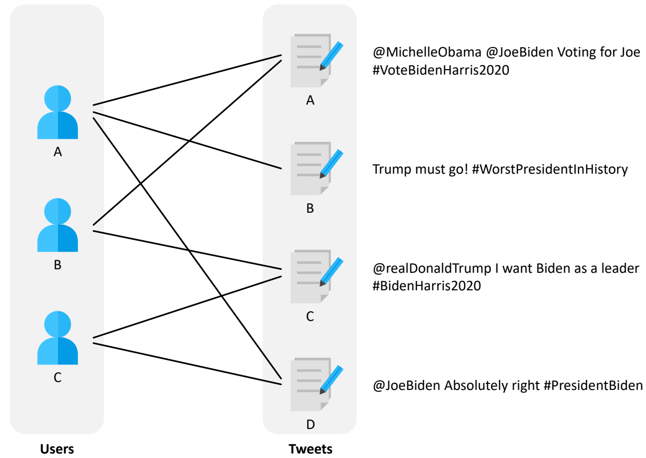


Figure 2: Example of a user-tweet bipartite graph. The links represent posting and retweeting behaviors.

crawl tweets posted between October 1, 2020, and November 2, 2020, using the Twitter Streaming API. We filter these tweets with the query: *biden* OR *trump*, corresponding to the two main candidates from the Democratic Party (Joe Biden) and the Republican Party (Donald Trump). In total, we gather 138.9 million tweets in English, along with their associated metadata.

We annotate the tweets by leveraging the prevalent usage of hashtags in tweets, which tend to reflect the stances expressed throughout the entire tweet. Initially, we identify the most frequently used hashtags and manually categorize those explicitly expressing support or opposition to one of the candidates as either pro-Biden or pro-Trump. Hashtags opposing one candidate are considered as supporting the other (e.g., *#VoteTrumpOut* is categorized as pro-Biden). We then iteratively expand the hashtag set by discovering new hashtags significantly related to the initial set through co-occurrence analysis. After the expansion, the hashtag set comprises a total of 221 labeled hashtags, with 144 in support of Biden and 77 in support of Trump. To ensure the consistency of our hashtag classification, we validate it through a hashtag co-occurrence network (Martinez-Romo et al. 2011; Bovet, Morone, and Makse 2018; Zhou et al. 2021). In Figure 1, we showcase a subset of these hashtags selected based on their frequency of occurrence. Next, we filter tweets containing at least one hashtag from the hashtag set. In cases where a tweet contains multiple hashtags, we retain it only if all the hashtags support the same candidate. Finally, we annotate each tweet according to the polarity of the hashtags within it.

To avoid duplicates, we exclude retweets (tweets starting with "RT @username") and record retweet information as metadata for the corresponding original tweets. Additionally, we remove the hashtags used for annotation from tweets to avoid target leakage, while retaining other hashtags as they may contain crucial information about the stance. In the end, our dataset consists of 1,123,749 labeled tweets, with 724,057 supporting Biden and 399,692 supporting Trump.

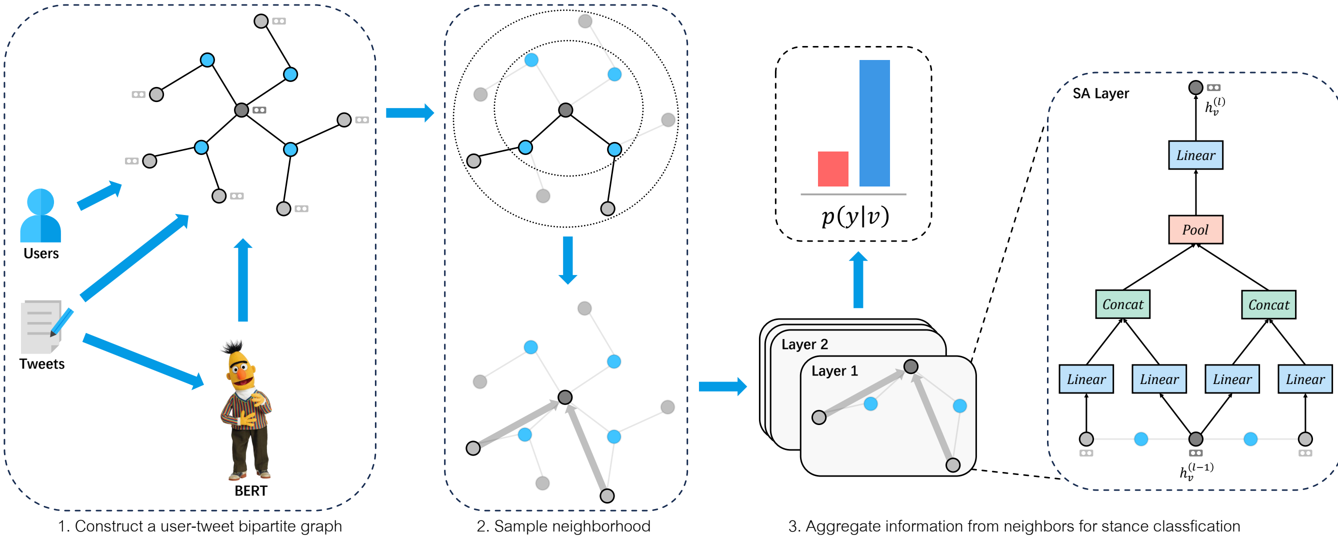


Figure 3: Overview of our proposed approach. We illustrate how a target node aggregates information from two sampled second-order neighbors. The arrows represent the flow of information.

Methodology

User-Tweet Bipartite Graph Construction

In assessing the stance of a tweet, it is essential to consider not only its content but also its social context (Lai et al. 2017). Therefore, we construct a bipartite graph to model interactions between users and tweets, where each edge signifies a user's behavior towards a tweet. We take into account both posting and retweeting behaviors, as users often retweet messages that align with their ideological preferences (Conover et al. 2011; Wong et al. 2016). The resulting graph comprises 1,123,749 tweet nodes, 700,507 user nodes, 1,123,749 post links, and 1,473,818 retweet links. Since this framework can also accommodate other user behaviors such as liking and pinning, we denote the graph as $\mathcal{G} = (\mathcal{U}, \mathcal{T}, \mathcal{E}, \mathcal{R})$, where $\mathcal{U}$ and $\mathcal{T}$ represent sets of user nodes and tweet nodes, respectively. $\mathcal{E}$ is the set of edges connecting nodes from $\mathcal{U}$ to $\mathcal{T}$. Additionally, each edge is assigned a relation type according to the user behavior it signifies, denoted by $\mathcal{R}$.

In recent times, GNNs have gained considerable traction for effectively processing such graph-structured data, exhibiting remarkable performance across various domains (Lv et al. 2023; Luo et al. 2023). Inspired by this success, we intend to develop a GNN to derive tweet representations from the graph and subsequently determine their stances. To this end, our initial step involves assigning initial features to the nodes in the graph. Initializing tweet node features is straightforward, as we can leverage a pre-trained language model. However, initializing user node features poses a challenge due to the absence of textual information. To uphold ethical considerations (Allein, Moens, and Perrotta 2023), we refrain from utilizing users' profile descriptions as the initial features in our approach. Consequently, it becomes difficult to provide user nodes with appropriate initial features that align with tweet nodes.

In a bipartite graph, each node's first-order neighbors belong to a different type than the node itself, while second-order neighbors share the same type. In simpler terms, a tweet node's second-order neighbors in the user-tweet bipartite graph are also tweet nodes. Notably, these tweets tend to share the same stance as the original tweet. As illustrated in Figure 2, using users A and B as bridges, tweet A's second-order neighbors include tweets B, C, and D, all of which share similar stances. To leverage this insight, we introduce a skip aggregation mechanism that enables each tweet node to aggregate information from its second-order neighbors. User nodes only serve as bridges during neighborhood sampling, bypassing the initialization problem.

Skip Aggregation Graph Convolution Layer

Modern GNNs commonly employ a two-phase scheme involving aggregation and combination to update node features in a graph (Xu et al. 2019). Formally, the l-th layer of a GNN can be defined as follows:

$$\begin{aligned} a_v^{(l)} &= \mathrm{AGGREGATE}^{(l)}\left(\left\{h_u^{(l-1)} : u \in \mathcal{N}(v)\right\}\right), \\ h_v^{(l)} &= \mathrm{COMBINE}^{(l)}\left(h_v^{(l-1)}, a_v^{(l)}\right). \end{aligned} \tag{1}$$

Here, $a_v^{(l)}$ represents the aggregated feature of node v's neighborhood, $\mathcal{N}(v)$ denotes the set of neighboring nodes of node v, and $h_v^{(l)}$ is the feature of node v at the l-th iteration. We initialize $h_v^{(0)}$ as $\mathbf{x}_v$, where $\mathbf{x}_v$ is the initial feature of node v.

The choice of functions $\mathcal{N}(\cdot)$, $\mathrm{AGGREGATE}^{(l)}(\cdot)$, and $\mathrm{COMBINE}^{(l)}(\cdot)$ is a critical aspect of GNN design. To efficiently train on the large-scale graph, we adopt a random walk-based neighborhood sampling strategy. Specifically, to capture second-order neighbors, we simulate n random walks of length 2 for each tweet node in the graph.

During these walks, we record the visited tweet nodes along with their corresponding visit frequencies. Consequently, we define the neighborhood of a tweet node as the top k most frequently visited tweet nodes across the random walks initiated from that node.

This approach yields an informative and representative set of neighboring tweet nodes for each tweet node. Notably, a tweet node v and its k neighboring tweet nodes form k unique tweet node pairs $\{(v, u) : u \in \mathcal{N}(v)\}$. Unlike typical scenarios where only a single edge connects each pair of nodes, our case involves a more complex structure. Specifically, each tweet node pair is connected by a user node and two edges: one edge $e^v_{v,u}$ links the target tweet node v to the user node, and the other edge $e^u_{v,u}$ links the neighboring tweet node u to the user node. Importantly, due to the heterogeneity of the user-tweet bipartite graph, these edges carry specific relation types, such as posting and retweeting. Different tweet node pairs may exhibit distinct types of $e^v_{v,u}$ and $e^u_{v,u}$. Hence, effectively leveraging this relational information is crucial for enhancing the representation learning of tweet nodes.

To achieve this, instead of applying a uniform transformation to all edges, our proposed Skip Aggregation (SA) graph convolution layer incorporates relation-specific transformations. For each pair of tweet nodes, a tweet node's feature is transformed according to the relation type of the connecting edge, allowing the model to adapt its behavior according to the edge semantics. Specifically, the SA layer operates as follows for each tweet node:

$$a_v^{(l)} = \mathrm{Pool}\left(\left\{\sigma\left(\left[\mathbf{W}^{(l)}_{v,\phi(e^v_{v,u})} \cdot h_v^{(l-1)} \| \mathbf{W}^{(l)}_{u,\phi(e^u_{v,u})} \cdot h_u^{(l-1)}\right]\right) : u \in \mathcal{N}(v)\right\}\right), \quad (2)$$

$$h_v^{(l)} = \sigma\left(\mathbf{W}_c^{(l)} \cdot a_v^{(l)}\right), \quad (3)$$

where $\|$ represents concatenation, $\sigma(\cdot)$ denotes a nonlinear activation function, and $\phi : \mathcal{E} \to \mathcal{R}$ is a relation type mapping function. $\mathbf{W}^{(l)}_{v,\phi(e^v_{v,u})}, \mathbf{W}^{(l)}_{u,\phi(e^u_{v,u})} \in \mathbb{R}^{d\times d}$ are relation-specific weight matrices that transform the features of the target tweet node v and the neighboring tweet node u according to the relation types of edges $e^v_{v,u}$ and $e^u_{v,u}$, respectively. $\mathbf{W}_c^{(l)} \in \mathbb{R}^{2d\times d}$ combines the concatenated features for use in the next layer. The Pool operator offers various pooling options for the representations of tweet node pairs, including mean, max, sum, weighted sum (Ying et al. 2018), and LSTM (Hamilton, Ying, and Leskovec 2017).

Figure 3 provides an overview of our proposed approach, along with details of the SA layer. The SA layer ensures that the aggregation process incorporates the relational information present in the two edges between the target tweet node and each of its neighboring tweet nodes, enabling more expressive and accurate modeling of the bipartite graph structure.

We refer to the GNN model formed by stacking multiple SA layers as SA-GNN. Algorithm 1 describes the embedding generation process of an L-Layer SA-GNN on a labeled bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{T}, \mathcal{E}, \mathcal{R})$. The normalization in Line 10 makes training more stable.

Algorithm 1: SA-GNN embedding generation (i.e., forward propagation) algorithm

---

Input: Labeled bipartite graph $\mathcal{G}(\mathcal{U}, \mathcal{T}, \mathcal{E}, \mathcal{R})$; input features $\{\mathbf{x}_v : v \in \mathcal{V}\}$; depth L
Output: Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

```
1:  h_v^(0) ← x_v, ∀v ∈ V;
2:  for l = 1 ... L do
3:      for v ∈ V do
4:          for u ∈ N(v) do
5:              a_{v,u}^(l) ← σ([W^(l)_{v,φ(e^v_{v,u})} · h_v^(l−1) ‖ W^(l)_{u,φ(e^u_{v,u})} · h_u^(l−1)]);
6:          end for
7:          a_v^(l) ← Pool({a_{v,u}^(l) : u ∈ N(v)});
8:          h_v^(l) ← σ(W_c^(l) · a_v^(l));
9:      end for
10:     h_v^(l) ← h_v^(l) / ‖h_v^(l)‖_2, ∀v ∈ V;
11: end for
12: z_v ← h_v^(L), ∀v ∈ V;
```

Stance Classification and Model Training

Thus far, SA-GNN enables the generation of behavior-aware representations for each tweet node in the graph. Following this, to ascertain the political stances of the tweets, these representations are processed through a fully connected layer followed by a sigmoid function to compute the predicted label probabilities:

$$\hat{y}_i = \mathrm{Sigmoid}\left(\mathbf{W} \cdot \mathbf{z}_i + \mathbf{b}\right), \quad (4)$$

where $\mathbf{z}_i$ is the new representation of tweet node v_i generated by SA-GNN.

In terms of model training, we employ the binary cross-entropy loss as the objective function:

$$\mathcal{L} = -\sum_{v_i \in \mathcal{V}} y_i \log \hat{y}_i + (1 - y_i) \log\left(1 - \hat{y}_i\right), \quad (5)$$

where $\mathcal{V}$ represents the set of tweet nodes in the graph.

Experiments

Experimental Settings

Baselines We compare our proposed model[1] against a range of baseline models, which can be broadly classified into two main categories: **text-based methods** and **graph-based methods**. Within the realm of graph-based methods, we further distinguish between two subcategories: **random walk-based methods** and **GNN-based methods**.

- **TextCNN** (Kim 2014) is a Convolutional Neural Network (CNN) architecture used for text classification and sentiment analysis, utilizing convolutional filters to extract meaningful features from textual data.
- **fastText** (Bojanowski et al. 2017) is a library for text classification and word embedding that uses character-level embeddings and employs a shallow neural network.

[1]https://github.com/Crysta1ovo/SA-GNN

- **BERT** (Devlin et al. 2019) is a pre-trained language model that has revolutionized natural language understanding and generation tasks by capturing contextual information bi-directionally.
- **DeepWalk** (Perozzi, Al-Rfou, and Skiena 2014) learns node embeddings by treating random walks on a graph as sentences.
- **node2vec** (Grover and Leskovec 2016) extends DeepWalk by introducing a flexible exploration strategy, allowing nodes to be embedded based on both breadth-first and depth-first random walks.
- **metapath2vec++** (Dong, Chawla, and Swami 2017) is designed for heterogeneous information networks and learns node embeddings by defining metapaths that capture the rich semantics of connections between different types of nodes.
- **FeatWalk** (Huang et al. 2019) generates node embeddings by considering both network structure and node content information through random walks and feature learning.
- **BiNE** (Gao et al. 2018) embeds node in a bipartite graph by considering both explicit relations and high-order implicit relations.
- **GCN** (Kipf and Welling 2017) is a neural network architecture that operates directly on graph-structured data by aggregating and updating node representations based on their local neighborhood.
- **GAT** (Velickovic et al. 2018) extends GCN by introducing attention mechanisms to assign different importance scores to neighboring nodes during the aggregation step.
- **GraphSAGE** (Hamilton, Ying, and Leskovec 2017) is a graph representation learning method that samples and aggregates information from a node's neighborhood to generate embeddings that capture both local and global graph structure.
- **GIN** (Xu et al. 2019) incorporates an isomorphism-insensitive readout function, making it particularly effective for tasks where graph structure matters more than node attributes.
- **RGCN** (Schlichtkrull et al. 2018) extends GCN to handle graphs with multiple types of relationships by incorporating relational information into the convolutional operations, allowing for more expressive graph representations.
- **RGAT** (Busbridge et al. 2019) is a variation of GAT that incorporates relational information into the attention mechanism, enabling it to capture complex patterns in graphs with multiple types of edges and relationships.
- **HGT** (Hu et al. 2020) is a transformer-based model specifically designed for handling heterogeneous graphs, employing node- and edge-type specific parameters to effectively capture diverse attention patterns across different edges in the graph.
- **L-BGNN** (Xie et al. 2022) aggregates information from both partitions of a bipartite graph by adopting inter-domain message passing and intra-domain alignment.

Implementation Details For the BERT baseline, we fine-tune the pre-trained $\text{BERT}_{\text{base}}$ model for 5 epochs, using a learning rate of 5e-5. The $\text{BERT}_{\text{base}}$ model consists of 12 layers, each with 768 hidden units. We train the random walk-based baselines with an embedding size of 128 and a learning rate of 1e-2. The context window size around each node in a random walk is set to 5, and we use 5 negative samples for each positive sample. Both the GNN-based baselines and SA-GNN have a hidden size of 768 and are trained for 5 epochs with a learning rate of 1e-3.

In terms of node initialization, both the GNN-based baselines and our proposed SA-GNN require the initialization of tweet node features. To achieve this, we employ a pre-trained $\text{BERT}_{\text{base}}$ model to encode the content of tweets. The resulting encoded representations serve as the initial features for the corresponding tweet nodes in the graph. Notably, since SA-GNN omits the use of user node features, only the GNN-based baselines require the initialization of user node features. In these instances, the feature of each user node is initialized with the average of the features of tweet nodes in its neighborhood, which has been experimentally verified to be superior to random initialization.

All models are trained on a Tesla V100. Due to the enormity of the graph, it is infeasible to fit the features of all nodes into the GPU. Therefore, we perform stochastic mini-batch training using the neighborhood sampling technique. For the GNN-based baselines, we randomly select 5 neighbors per layer. As for SA-GNN, we sample neighbors using a random walk-based strategy, as previously described. Specifically, we simulate 10 random walks and select the top 5 most frequently visited tweet nodes per layer.

We consider Accuracy, F1 score, and AUC as evaluation metrics to quantitatively evaluate model performance. The training process involves training the model on the training set and selecting the best-performing model based on its performance on the validation set. We allocate 80% of the instances for training, 10% for validation, and another 10% for testing. To ensure the reliability and robustness of our results, we repeat the experiment 5 times and report the average performance.

Main Results

In this section, we conduct a comprehensive evaluation of our proposed SA-GNN, comparing it with various baseline methods using different feature sets. The results, summarized in Table 1, showcase the exceptional performance of SA-GNN, surpassing all other methods with the incorporation of all available features. SA-GNN demonstrates remarkable results, achieving an accuracy of 92.57%, an F1 score of 89.35%, and an AUC of 91.47%.

Among the text-based baselines, BERT performs the best, outperforming TextCNN and fastText. BERT achieves an accuracy of 90.70%, an F1 score of 86.67%, and an AUC of 89.43%. When incorporating textual information, FeatWalk outperforms other random walk-based baselines, with an accuracy of 81.14%, an F1 score of 71.24%, and an AUC of 77.67%.

The GNN-based baselines consistently outperform the random walk-based baselines by leveraging both text and

Method	Feature				Performance		
	Text	Graph	Relation	Bipartition	Acc.	F1	AUC
TextCNN	✓				0.7984 ± 0.05	0.6961 ± 0.14	0.7650 ± 0.10
fastText	✓				0.7420 ± 0.12	0.5724 ± 0.15	0.6846 ± 0.08
BERT	✓				0.9070 ± 0.04	0.8667 ± 0.07	0.8943 ± 0.08
DeepWalk		✓			0.8106 ± 0.03	0.6678 ± 0.12	0.7489 ± 0.06
node2vec		✓			0.8191 ± 0.06	0.6856 ± 0.11	0.7598 ± 0.07
metapath2vec++		✓	✓		0.8072 ± 0.04	0.6493 ± 0.10	0.7388 ± 0.06
FeatWalk	✓	✓			0.8114 ± 0.03	0.7124 ± 0.09	0.7767 ± 0.07
BiNE		✓		✓	0.7850 ± 0.07	0.6572 ± 0.15	0.7390 ± 0.10
GCN	✓	✓			0.9124 ± 0.04	0.8734 ± 0.08	0.8984 ± 0.09
GAT	✓	✓			0.9161 ± 0.05	0.8794 ± 0.07	0.9034 ± 0.07
GraphSAGE	✓	✓			0.9151 ± 0.06	0.8780 ± 0.07	0.9026 ± 0.06
GIN	✓	✓			0.9148 ± 0.06	0.8775 ± 0.08	0.9020 ± 0.07
RGCN	✓	✓	✓		0.9177 ± 0.08	0.8820 ± 0.05	0.9057 ± 0.06
RGAT	✓	✓	✓		0.9180 ± 0.06	0.8830 ± 0.07	0.9072 ± 0.09
HGT	✓	✓	✓		0.9183 ± 0.04	0.8826 ± 0.07	0.9060 ± 0.06
L-BGNN	✓	✓		✓	0.9128 ± 0.06	0.8744 ± 0.10	0.8994 ± 0.08
SA-GNN	✓	✓	✓	✓	$\mathbf{0.9257 \pm 0.04}$	$\mathbf{0.8935 \pm 0.08}$	$\mathbf{0.9147 \pm 0.05}$
SA-GNN$_{\text{w/o relation}}$	✓	✓		✓	$\underline{0.9240 \pm 0.06}$	$\underline{0.8911 \pm 0.07}$	$\underline{0.9129 \pm 0.07}$

Table 1: Performance comparison of various methods. We conduct five trials with different random seeds and report the mean and standard deviation of the results on the test set. The best result in each metric is indicated in bold, while the second-best result is underlined. The checkmark (✓) indicates the presence of a feature in a particular method.

Method	Acc.	F1	AUC
SA-GNN$_{\text{max}}$	$\mathbf{0.9257 \pm 0.04}$	$\mathbf{0.8935 \pm 0.08}$	$\mathbf{0.9147 \pm 0.05}$
SA-GNN$_{\text{mean}}$	0.9218 ± 0.05	0.8880 ± 0.06	0.9106 ± 0.08
SA-GNN$_{\text{sum}}$	0.9232 ± 0.06	0.8897 ± 0.10	0.9115 ± 0.06
SA-GNN$_{\text{wsum}}$	0.9211 ± 0.04	0.8867 ± 0.08	0.9092 ± 0.05
SA-GNN$_{\text{LSTM}}$	0.9254 ± 0.02	0.8931 ± 0.05	0.9144 ± 0.07

Table 2: Performance comparison when using different pooling functions.

graph features. Notably, HGT and RGAT, which additionally incorporate relational information, stand out among the GNN-based baselines. HGT achieves an impressive accuracy of 91.83%, while RGAT attains the highest F1 score of 88.30% and AUC of 90.72%.

We also evaluate a variant of our model that does not utilize relational information, aggregating information from second-order neighbors without considering the two edges between each tweet node pair. The results indicate that the ablated model, SA-GNN$_{\text{w/o relation}}$, achieves the second-best performance but still maintains a notable performance gap compared to the complete model in all aspects.

In Table 2, we analyze the performance of our model when using different pooling functions. Our analysis encompasses mean, max, sum, weighted sum, and LSTM pooling techniques. Remarkably, the max function emerges as the most effective among these options, underscoring the significance of selecting appropriate pooling strategies in enhancing our model's performance.

Performance Analysis

Effect of User Behavioral Information To evaluate the effect of user behavioral information on performance, we perform a comparative analysis between GNN-based methods and the BERT baseline. User behavioral information holds significance as it provides valuable contextual cues and insights into user preferences. GNN-based methods leverage this information through a user-tweet bipartite graph, with edges representing users' behaviors towards tweets. In addition to textual data, the GNN-based methods additionally leverage the graph structure to capture interactions between users and tweets, resulting in a noticeable enhancement in performance. Overall, the consistently superior performance of GNN-based methods over the BERT baseline underscores the critical importance of incorporating user behavioral information into the task of political stance detection.

Effect of Skip Aggregation Skip aggregation enables the effective aggregation of information from second-order neighbors in the user-tweet bipartite graph. This addresses the challenge of initializing user node features in the absence of textual information. In this section, we analyze the performance of SA-GNN in comparison to the GNN-based baselines, which initialize user node features heuristically by averaging the features of their neighboring tweet nodes in the graph. The results consistently demonstrate that SA-GNN outperforms all GNN-based baselines across all evaluation metrics, highlighting the effectiveness of skip aggregation in leveraging user behavioral information. Moreover, even when we exclude the utilization of relational informa-

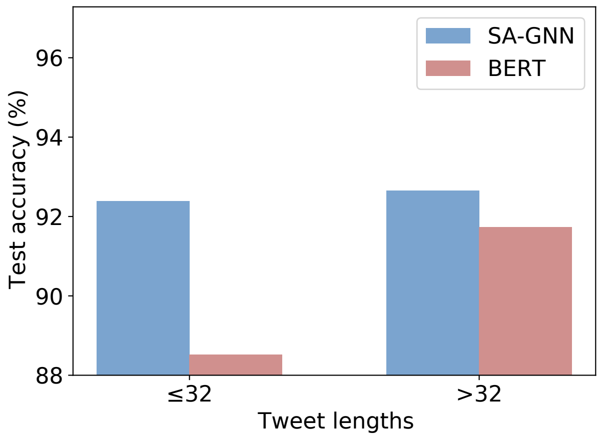


Figure 4: Performance analysis results in terms of tweet length.

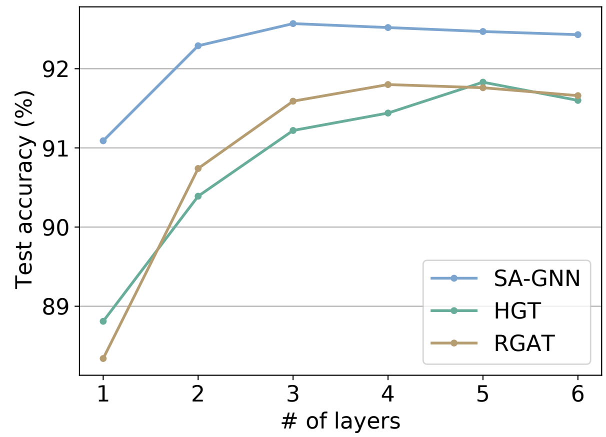


Figure 5: Impact of model depth on performance.

tion (as demonstrated by SA-GNN$_{\text{w/o relation}}$ in Table 1), our model's performance remains superior to that of the GNN-based baselines. These findings underscore the indispensable contribution of skip aggregation to the outstanding performance achieved by SA-GNN.

Effect of Relational Information To assess the effect of relation information and the capacity of SA-GNN to utilize such information, we conduct a comparison involving SA-GNN and its variant SA-GNN$_{\text{w/o relation}}$, as well as RGAT. The results, as presented in Table 1, clearly illustrate a notable disparity between SA-GNN and its variant, which confirms our intuition that the information contained in the two edges between the tweet node pair is indeed valuable. Similar trends are discernible when comparing RGAT to GAT, which further demonstrates that incorporating relational information can lead to certain improvements. Notably, SA-GNN surpasses RGAT by considerable margins, achieving improvements of 0.77, 1.05, and 0.75 points in terms of Accuracy, F1 score, and AUC, respectively. These results indicate that SA-GNN exhibits superior efficacy in leveraging relational information, thereby enabling more effective utilization of user behavioral information.

Performance on Short Texts This section presents a performance analysis to explore the capabilities of SA-GNN specifically when applied to short texts. We begin by examining the distribution of tweet lengths in our dataset. The length of a tweet is defined as the number of sub-word tokens split by the WordPiece tokenizer used in BERT. Statistics indicate that approximately 30% of the tweets in our dataset have a length no greater than 32, which we categorize as short texts. Subsequently, we compare the performance of SA-GNN with that of BERT on these short texts. Figure 4 illustrates the performance trends for both models.

Notably, BERT exhibits a significant decline in performance when dealing with short texts. Conversely, SA-GNN demonstrates remarkable stability, experiencing only a marginal reduction in performance. This noteworthy outcome can be attributed to SA-GNN's effective utilization of user behavioral information. In general, tweets tend to deviate from strict grammatical rules and, when even short, inherently contain limited linguistic features, often lacking the necessary context. Consequently, determining the stances expressed in such tweets based solely on their content becomes challenging, resulting in BERT's underperformance. In contrast, SA-GNN mitigates this challenge by leveraging the graph structure, aggregating information from other relevant tweets through graph convolution. This approach enriches the representation of short texts and empowers the classifier to capture stances based on more comprehensive information.

Impact of Model Depth In this part, we investigate the impact of model depth on the performance of SA-GNN and two GNN-based baseline models: HGT and RGAT. We explore a range of depths, spanning from 1 to 6, and present the accuracy trends of the three models on the test set in Figure 5.

In general, we observe that the performance of all three models improves with increasing model depth, up to a certain point. SA-GNN achieves its peak performance with 3 layers, while HGT and RGAT exhibit their best performance with 5 and 4 layers, respectively. Notably, the performance of 1-layer HGT and RGAT falls significantly short when compared to their performance with multiple layers. This discrepancy can be explained by the fact that 1-layer GNN-based baseline models only aggregate information from first-order neighbors, which primarily consist of user nodes containing limited information. Consequently, their performance is even worse than the BERT baseline.

Furthermore, given that SA-GNN aggregates information from second-order neighbors, the GNN-based baselines require twice as many layers as SA-GNN to capture a comparable amount of relevant tweet information. However, as the number of layers increases, issues such as vanishing gradient and over-smoothing gradually emerge, leading to a decline in performance.

Pro-Biden		Pro-Trump	
Trigram	r	Trigram	r
covid covid covid	0.542	president donald trump	0.485
joe biden president	0.506	trump trump trump	0.562
vote joe biden	0.473	make america great	0.598
removal office legal	0.431	god bless america	0.569
fascist donald trump	0.577	love president trump	0.659
vote biden harris	0.566	love mr president	0.597
health care plan	0.430	god bless president	0.570
vote vote vote	0.539	vote president trump	0.579
super spreader event	0.583	keep america great	0.589
vote life depends	0.565	vote donald trump	0.570

Table 3: Top ten most frequent trigrams in each stance category and their Pearson correlation coefficients with respective categories.

Descriptive Analysis

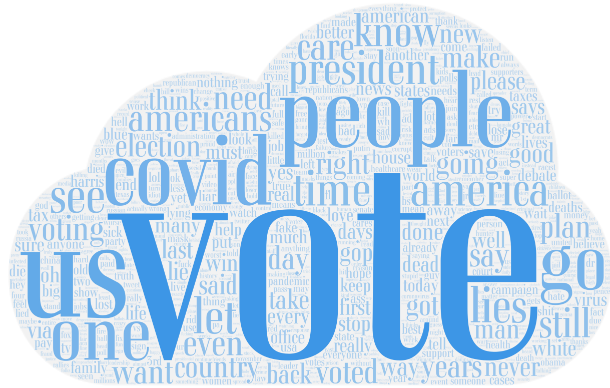
In this section, we provide a descriptive analysis of our dataset, aiming to uncover prevalent language patterns in tweets supporting or opposing each candidate.

To commence, we extract the top ten most frequently occurring trigrams in both pro-Biden and pro-Trump tweets, respectively. Subsequently, we assess the statistical significance of these trigrams by calculating the Pearson correlation coefficient between the tweets within each category and the trigrams generated by the corresponding category of tweets (Islam, Roy, and Goldwasser 2023). This calculation is grounded in BERT embeddings. The summarized results are presented in Table 3.

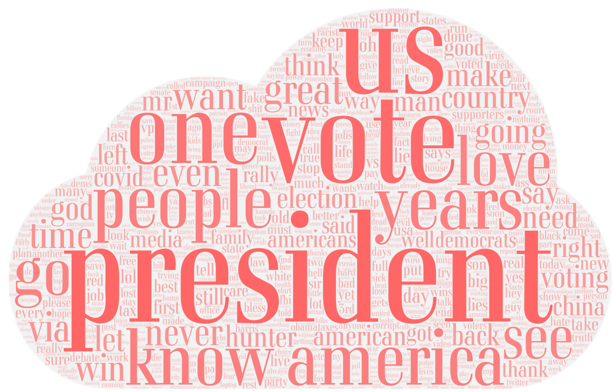
Upon a thorough examination of Table 3, several key findings emerge. Pro-Biden tweets prominently feature trigrams such as *joe biden president*, *vote joe biden*, *vote biden harris*, which indicate strong support for Joe Biden. Additionally, some trigrams express opposition to Donald Trump, as seen in *removal office legal* and *fascist donald trump*. These trigrams reflect the criticism of Trump's presidency. Trump's handling of the COVID-19 pandemic is a frequent target of criticism, evident in trigrams like *covid covid covid* and *super spreader event*.

In contrast, pro-Trump tweets frequently include trigrams like *president donald trump*, *trump trump trump*, *make america great*, and *god bless america*, underscoring enthusiastic support for Donald Trump. Supporters emphasize their desire for Trump's re-election through trigrams like *vote president trump* and *keep america great*. Notably, unlike the pro-Biden category, there are no prominent trigrams expressing opposition to Joe Biden.

To visually represent the qualitative differences between the two categories, we create word clouds in Figure 6 to showcase the most commonly used words in the tweets. In these word clouds, the size of each word corresponds to its frequency of occurrence. The word cloud for pro-Biden tweets prominently features terms like *vote* and *covid*, reflecting the energetic pre-election campaigning by Biden's supporters and their criticism of Trump's pandemic response. Conversely, the word cloud for Pro-Trump tweets emphasizes words such as *president* and *vote*, highlighting the fervent desire among Trump supporters for his re-election.



(a) pro-Biden



(b) pro-Trump

Figure 6: Word clouds for each stance category.

Limitations

While leveraging SA-GNN to derive behavior-aware tweet representations from the user-tweet bipartite graph has yielded substantial improvements in performance, we acknowledge that it has certain limitations that warrant consideration.

Primarily, our SA-GNN model is designed around the construction of the user-tweet bipartite graph, making it less suitable for datasets comprised solely of text. To efficiently annotate large-scale tweets in our dataset, we employ a hashtag-based automatic annotation method, which inevitably introduces some noise. Additionally, a notable limitation arises when dealing with tweets that are less popular or posted by inactive users on the platform. In such scenarios, the model's ability to aggregate information during the graph convolution phase is constrained due to the limited availability of relevant tweets. As a result, the model's performance may suffer as it lacks access to an adequate amount of information. Similarly, for newly posted tweets beyond the scope of our initial data collection, their authors or retweeters may not be included in the previously constructed bipartite graph, posing challenges for our model in

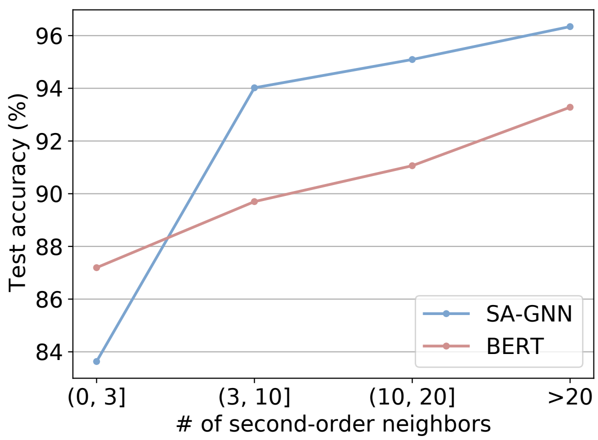


Figure 7: Performance analysis results in terms of the number of second-order neighbors of tweets in the graph.

making precise inferences.

To substantiate our concerns about these limitations, we conduct a performance analysis. Specifically, we compare the performance of SA-GNN and BERT across tweets with varying numbers of second-order neighbors in the graph. As shown in Figure 7, the performance of BERT, unaffected by graph-related factors, demonstrates an upward trend as the number of second-order neighbors increases. This observation suggests that tweets with more neighbors in the graph often exhibit clearer stances or employ more formal language, enabling the model to make more accurate predictions. However, the performance of SA-GNN experiences a notable decline for tweets with no more than 3 second-order neighbors, even falling below that of BERT. This performance deterioration aligns with a common challenge faced by GNNs, which tend to exhibit bias against low-degree nodes in the graph (Tang et al. 2020).

While BERT's performance is inferior to graph-based models, it offers greater flexibility and efficiency as it relies solely on the linguistic information within the tweet. Therefore, we have plans to address these limitations in the future by integrating user behavioral information into BERT. This could be achieved through the design of novel pretraining objectives or the utilization of contrastive learning techniques. By doing so, we aim to develop a more flexible and efficient political stance detection model while maintaining a high level of performance.

Conclusion

In this study, we embark on the task of detecting political stances in tweets related to the 2020 US presidential election. To support our research, we curate a large-scale dataset sourced from Twitter, annotating it using hashtags as indicators of political polarity. Our developed GNN-based framework incorporates a novel skip aggregation mechanism, enabling the extraction of behavior-aware tweet representations from a user-tweet bipartite graph. Through extensive experiments on our annotated dataset, our model consistently demonstrates its superiority over various state-of-the-art models in the field. Additionally, we conduct thorough analyses to underscore the robustness of our framework when applied to short texts. Furthermore, we present compelling visualizations to showcase the prevailing language observed within our dataset. In closing, we discuss the limitations of our work and propose promising avenues for future research.

Ethical Considerations

This paper primarily focuses on the 2020 US presidential election, analyzing the stances expressed in election-related tweets regarding the candidates. Our approach, involving hashtag-based annotation and bipartite graph analysis, is adaptable and can be readily applied to other political events, such as the upcoming 2024 US presidential election. However, it is important to acknowledge that the misclassification made by our approach may affect the prediction of election outcomes and the assessment of the political landscape.

Our dataset is sourced from the Twitter platform. However, Twitter users are not representative of the entire U.S. population. They tend to be younger, more urban, and more politically engaged. This demographic bias may potentially skew the results of downstream tasks including election prediction.

We uphold ethical principles by refraining from exploiting user information, such as profile descriptions, to initialize features for user nodes in the graph. Instead, our model allows tweet nodes to aggregate information from second-order neighbors in the graph, comprised of other tweet nodes. Nevertheless, safeguarding the privacy of the dataset is of utmost importance to prevent the inadvertent disclosure of personal information. To ensure privacy protection, we have implemented stringent measures that include the removal or obfuscation of any personally identifiable information, including usernames and any other sensitive data that could potentially be used to identify individuals.

We strongly emphasize that our research should never be used for malicious intent or inappropriate purposes. Specifically, we discourage any attempt to use our findings to target individuals or groups based on their political beliefs.

Acknowledgements

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

References

Aiyappa, R.; An, J.; Kwak, H.; and Ahn, Y.-Y. 2023. Can we trust the evaluation on ChatGPT? *arXiv preprint arXiv:2303.12767*.

Allein, L.; Moens, M.-F.; and Perrotta, D. 2023. Preventing profiling for ethical fake news detection. *Information Processing & Management*, 60(2): 103206.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5: 135–146.

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.

Busbridge, D.; Sherburn, D.; Cavallo, P.; and Hammerla, N. Y. 2019. Relational graph attention networks. *arXiv preprint arXiv:1904.05811*.

Chen, W.; Zhang, X.; Wang, T.; Yang, B.; and Li, Y. 2017. Opinion-aware Knowledge Graph for Political Ideology Detection. In *IJCAI*, volume 17, 3647–3653.

Conover, M.; Ratkiewicz, J.; Francisco, M.; Gonçalves, B.; Menczer, F.; and Flammini, A. 2011. Political polarization on twitter. In *Proceedings of the international aaai conference on web and social media*, volume 5, 89–96.

Cruickshank, I. J.; and Ng, L. H. X. 2023. Use of Large Language Models for Stance Classification. *arXiv preprint arXiv:2309.13734*.

Darwish, K.; Stefanov, P.; Aupetit, M.; and Nakov, P. 2020. Unsupervised user stance detection on Twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, 141–152.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 135–144.

Fagni, T.; and Cresci, S. 2022. Fine-Grained Prediction of Political Leaning on Social Media with Unsupervised Deep Learning. *Journal of Artificial Intelligence Research*, 73: 633–672.

Gao, M.; Chen, L.; He, X.; and Zhou, A. 2018. Bine: Bipartite network embedding. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 715–724.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 1024–1034.

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

Huang, X.; Song, Q.; Yang, F.; and Hu, X. 2019. Large-scale heterogeneous feature embedding. In *Proceedings of the AAAI conference on artificial intelligence*, 3878–3885.

Islam, T.; Roy, S.; and Goldwasser, D. 2023. Weakly Supervised Learning for Analyzing Political Campaigns on Facebook. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 17, 411–422.

Iyyer, M.; Enns, P.; Boyd-Graber, J.; and Resnik, P. 2014. Political Ideology Detection Using Recursive Neural Networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1113–1122. Baltimore, Maryland: Association for Computational Linguistics.

Jiang, J.; Ren, X.; and Ferrara, E. 2023. Retweet-BERT: political leaning detection using language features and information diffusion on social networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 17, 459–469.

Kamal, S.; Gullic, J.; and Bagavathi, A. 2022. Modeling Polarization on Social Media Posts: A Heuristic Approach Using Media Bias. In *International Symposium on Methodologies for Intelligent Systems*, 35–43. Springer.

Khatua, A.; Khatua, A.; and Cambria, E. 2020. Predicting political sentiments of voters from Twitter in multi-party contexts. *Applied Soft Computing*, 97: 106743.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Association for Computational Linguistics.

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.

Lai, M.; Cignarella, A. T.; Farías, D. I. H.; Bosco, C.; Patti, V.; and Rosso, P. 2020. Multilingual stance detection in social media political debates. *Computer Speech & Language*, 63: 101075.

Lai, M.; Hernández Farías, D. I.; Patti, V.; and Rosso, P. 2017. Friends and enemies of clinton and trump: using context for detecting stance in political tweets. In *Advances in Computational Intelligence: 15th Mexican International Conference on Artificial Intelligence, MICAI 2016, Cancún, Mexico, October 23–28, 2016, Proceedings, Part I 15*, 155–168. Springer.

Lampos, V.; Preoţiuc-Pietro, D.; and Cohn, T. 2013. A user-centric model of voting intention from Social Media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 993–1003. Sofia, Bulgaria: Association for Computational Linguistics.

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.

Li, C.; and Goldwasser, D. 2021a. MEAN: Multi-head Entity Aware Attention Networkfor Political Perspective Detection in News Media. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, 66–75. Online: Association for Computational Linguistics.

Li, C.; and Goldwasser, D. 2021b. Using Social and Linguistic Information to Adapt Pretrained Representations for Political Perspective Identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 4569–4579. Online: Association for Computational Linguistics.

Luo, X.; Ju, W.; Gu, Y.; Qin, Y.; Yi, S.; Wu, D.; Liu, L.; and Zhang, M. 2023. Toward Effective Semi-supervised Node Classification with Hybrid Curriculum Pseudo-labeling. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(3): 1–19.

Lv, J.; Song, K.; Ye, Q.; and Tian, G. 2023. Semi-supervised node classification via fine-grained graph auxiliary augmentation learning. *Pattern Recognition*, 137: 109301.

Manickam, I.; Lan, A. S.; Dasarathy, G.; and Baraniuk, R. G. 2019. IdeoTrace: a framework for ideology tracing with a case study on the 2016 U.S. presidential election. In *ASONAM '19: International Conference on Advances in Social Networks Analysis and Mining, Vancouver, British Columbia, Canada, 27-30 August, 2019*, 274–281. ACM.

Martinez-Romo, J.; Araujo, L.; Borge-Holthoefer, J.; Arenas, A.; Capitán, J. A.; and Cuesta, J. A. 2011. Disentangling categorical relationships through a graph of co-occurrences. *Physical Review E*, 84(4): 046108.

Maynard, D.; and Funk, A. 2011. Automatic detection of political opinions in tweets. In *Extended semantic web conference*, 88–99. Springer.

Ng, L. H. X.; and Carley, K. M. 2022. Is my stance the same as your stance? A cross validation study of stance detection datasets. *Information Processing & Management*, 59(6): 103070.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701–710.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Tang, X.; Yao, H.; Sun, Y.; Wang, Y.; Tang, J.; Aggarwal, C.; Mitra, P.; and Wang, S. 2020. Investigating and mitigating degree-related biases in graph convoltuional networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1435–1444.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Wong, F. M. F.; Tan, C. W.; Sen, S.; and Chiang, M. 2016. Quantifying political leaning from tweets, retweets, and retweeters. *IEEE transactions on knowledge and data engineering*, 28(8): 2158–2172.

Xiao, Z.; Song, W.; Xu, H.; Ren, Z.; and Sun, Y. 2020. TIMME: Twitter Ideology-detection via Multi-task Multi-relational Embedding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 2258–2268. ACM.

Xiao, Z.; Zhu, J.; Wang, Y.; Zhou, P.; Lam, W. H.; Porter, M. A.; and Sun, Y. 2023. Detecting political biases of named entities and hashtags on Twitter. *EPJ Data Science*, 12(1): 20.

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*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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 2018, London, UK, August 19-23, 2018*, 974–983. ACM.

Zhang, B.; Ding, D.; and Jing, L. 2022. How would stance detection techniques evolve after the launch of chatgpt? *arXiv preprint arXiv:2212.14548*.

Zhang, W.; Feng, S.; Chen, Z.; Lei, Z.; Li, J.; and Luo, M. 2022. KCD: Knowledge Walks and Textual Cues Enhanced Political Perspective Detection in News Media. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4129–4140. Seattle, United States: Association for Computational Linguistics.

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.

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 Abstract and the end of the Introduction.

(c) Do you clarify how the proposed methodological approach is appropriate for the claims made? Yes, see the Methodology.

(d) Do you clarify what are possible artifacts in the data used, given population-specific distributions? Yes, see the Ethical Considerations.

(e) Did you describe the limitations of your work? Yes, see the Limitations.

(f) Did you discuss any potential negative societal impacts of your work? Yes, see the Ethical Considerations.

(g) Did you discuss any potential misuse of your work? Yes, see the Ethical Considerations.

(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 Ethical Considerations.

(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, in order to preserve anonymity, our intention is to make our code accessible via a URL upon the acceptance of our paper.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? Yes, see the Implementation Details in the Experiments.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? Yes, see the Table 1.

(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 the Implementation Details in the Experiments.

(e) Do you justify how the proposed evaluation is sufficient and appropriate to the claims made? Yes, see the Main Results and the Performance Analysis in the Experiments. Our evaluation not only demonstrates the superior performance of our model in comparison to baseline models through comparative experiments, but it also provides a thorough assessment of each component's substantial contribution to our model.

(f) Do you discuss what is "the cost" of misclassification and fault (in)tolerance? Yes, see the Ethical Considerations.

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? NA

(g) If you are curating or releasing new datasets, did you create a Datasheet for the Dataset? 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? NA

(b) Did you describe any potential participant risks, with mentions of Institutional Review Board (IRB) approvals? NA

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA

(d) Did you discuss how data is stored, shared, and deidentified? NA