GUEST - Graph of User Encoder and Semantic Attention Network for Fake News Detection

Yen-Wen Lu
Academia Sinica
Taipei, Taiwan
ywlu2@iis.sinica.edu.tw

Cheng-Te Li
Academia Sinica
Taipei, Taiwan
chengte@mail.ncku.edu.tw

ABSTRACT

Recently, research works of fake news detection are devoted to studying explainable neural network frameworks to fulfill the task. However, most frameworks are focus on generating interpretable evidences from language viewpoint while other information like user profile or information about news structure and time are able to offer firmer evidences. Hence, we propose a new framework Graph of User Encoder and Semantic Attention Network (GUEST) to explore discover explainable evidences through social media contexts, user profiles, structure and time information of social media conversation. Experiments from two public datasets indicates that GUEST simultaneously outperforms state-of-art models and provides interpretable evidences from semantics and user characteristics.

CCS CONCEPTS

• **Security and privacy → Social aspects of security and privacy**.

KEYWORDS

Fake news; explainable machine learning; social network

ACM Reference Format:
Yen-Wen Lu and Cheng-Te Li. 2021. GUEST - Graph of User Encoder and Semantic Attention Network for Fake News Detection. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Due to the ubiquity and loose supervision, serious dissemination of fake news and false claims becomes a challenge needed to be solved. For example, Investigation from Pew Research Center[1] in 2016 reveals that up to 88% adults in U.S. exeprienced a great deal or some confusion about the current facts because of fake news proliferation. In order to cease the fake news propagation, effective fake news detection method is crucial. Although there are websites like Snopes[2] and Politifact[3] to do fake news check by professional journalists, such human fack-checking method is not efficient enough to address huge volume and fast-spreading fake news. Hence, automatic veracity checking methods are the main research trend.

. Current fake news verification is based on machine learning. Typical approaches leverages deep learning network to learn the truthfulness of a claim through capturing key contextual features of claim, relevant comments and articles. [14, 16, 24] In spite of their effectiveness, these approaches are insufficient in interpretablity. To solve this issue, recent research trend is focus on evidence-based neural networks models to improve explainability. Ma et al., Shu et al., Wu and Rao exert attention mechanism to explore trustworthy evidence from relevant replies and articles. Liu et al., Zhong et al., Zhou et al. utilize graph neural network to find evidence from a graph structure composed of claims and associated articles. In addition to language information, Lu and Li and Wu et al. and propose framework of GCAN and DTCA respectively to evaluate whether a source tweet is false or not via semantic and user characteristics together.

. Inspired by GCAN and DTCA, this paper presents a new deep learning model **G**raph of **U**ser **E**ncoder and **S**emantic A**T**tention Network (GUEST) for fake new verification task. Although GCAN and DTCA achieve astonoshing results, there are still several limitation. First, GCAN requires user retweet path, which may not applicable in some situation, such as PHEME [32] and RumourEval [3] whose datasets does not possess retweet information. Second, for DTCA, user characteristics are utilized to computed as thresholds to distinguish whether a comment is able to be an evidence. Nevertheless, thresholds are hyper-parameters rather than a learnable parameter, so model performance heavily depends on the fine-tuning of thresholds. Third, both models does not consider structural and temporal information which are proved to be effective on fact checking [19]. Consequently, we propose an end-to-end model GUEST to solve the above issues. GUEST is designed by four components. 1) Graph Attention Network (GAT) generates sentence level embeddings of comments. 2) Fi-GNN which integrates self-attention and Gated Graph Neural Networks is for learning user credibility indicator. 3) Co-attention fusion first learns co-relation between sentence and corresponding user and then further models the interaction between claim and comments. 4) Bi-LSTM is exerted on encoding whole structural and temporal information of a tweet conversation. The final binary classification is from proper fusion of contexts, user characteristics, structure and time features. Result of Experiment shows that GUEST does not only achieve better performance than state-of-art-models do, but also displays interpretable process of evidence selection. In conclusion, our contribution is as follows: 1)

[1]https://www.journalism.org/2016/12/15/many-americans-believe-fake-news-is-sowing-confusion/
[2]https://www.snopes.com/
[3]https://www.politifact.com/

https://doi.org/10.1145/nnnnnnn.nnnnnnn

we develop a new model to successfully fuse heterogeneous information together to generate effective representation for this binary classification task. 2) The evidence selection process is explainable either on language perspective or user credibility perspective. 3) Results of experiments on two widely-used fake news datasets shows competitive performance comparing to state-of-art models.

2 RELATED WORK

We generally categorize previous fake news detection works related to ours by varied methods neural network models implement. For attention based model, DeClarE [15] and HAN [12] applies attention mechanism to explore valid semantic evidence among articles. dEFEND [18] and DTCA [27] implement co-attention to make claim and relevant articles involve deep interaction. For multitask-learning based model, [7] implements LSTM to learn joint representation of veracity, stance and rumour for three tasks. Sifted-MTL [26] employs transformer encoder [4] and gate cells to learn shared information of two tasks. For graph-based model, Ma et al. uses RNN based model to achieve rumour classification from tree structure graph representing claim and following replies. With regard to FEVER datasets [21], GEAR [31] and KGAT [10] leverage graph attention neural network to gain key features from fully-connected evidence graph. DREAM [30] first builds a semantic role labeling (SRL) based graph and then conducts GAT and GCN [6] to accomplish fact checking task. HGAT [17] implements hierarchical graph attention network to model relationship among creator of claim, News article and news subject. GCAN [11] first employs GCN to generate semantic embedding of claim and then co-attention is implemented for involving semantic embedding and user characteristic embedding into interaction. TriFN [20] models interaction among publisher, user and news through matrix factorization method instead of deep neural network.

3 PROBLEM STATEMENT

We assume comments created by reliable users have the priority to be evidences on fake news detection task. In addition, there exists unique tree structure and time relation among source tweet and comments in fake news. Given a Twitter conversation containing a source tweet $\mathbf{c}$, associated comments $comm_i$, users $U = [u_c, u_1, ...u_m]$, and structure-temporal features ST, our goal is to predict veracity $\hat{y} \in [0, 1]$ of the source tweet by an explainable model.

4 GUEST - GRAPH OF USER ENCODER AND SEMANTIC ATTENTION NETWORK

In this section, we introduce our fake news detection model Graph of User Encoder and Semantic Attention Network (GUEST), which contains five components shown in Figure 1. Graph Attention Network (GAT) [23] is for generating sentence representation. Fi-GNN [9] implements self-attention mechanism [22] and GGNN [8] to generate user feature embedding. Co-attention fusion enables interaction between each sentence representation and corresponding user feature embedding. Temporal and structure embedding layer applies biLSTM to capture temporal and structure features of a Twitter conversation. Final prediction layer applies MLP to decide whether a Twitter claim is fake or not. Each component will be described in details in the following subsections.

4.1 Semantic Extraction

we propose a GAT framework for sentence representation. From Figure 1, GAT includes two parts: Pre-trained sentence Encoder and Graph Attention Neural Network (GAT). After Pre-trained sentence Encoder generates claim and comments representation, GAT propagates each comments information in a fully connected graph to obtain graph-based sentence embedding.

4.1.1 Pre-trained Sentence Encoder. Similar to previous work [31], We employ pre-trained BERT [22] as sentence encoder. Given a word sequence which could be a claim (c) or a claim-comment pair $[c \,||\, comm_i]$ which is a concatenation of a claim a comment, we feed it into pre-trained BERT and utilize the final state of BERT as sentence representation. That is,

$$\begin{aligned} \mathbf{c} &= BERT(c) \\ e_i &= BERT([c \;||\; comm_i]) \end{aligned} \tag{1}$$

where || is concatenation operation. Note that for both claim and claim-comment pair, a $[CLS]$ token is inserted in the beginning of the sentence. For claim-comment pair, a $[SEP]$ token is inserted between claim and comment.

4.1.2 Graph Attention Neural Network (GAT). In order to make further interaction among each evidence, a graph $\mathfrak{G} = (N, E)$ is constructed for information propagation of each comment, where $n_i \in N$ corresponds to nodes and $\varepsilon_{ij} \in E$ corresponds to graph edge. We design $\mathfrak{G}$ as fully connected and self-loop. For node set N, it is assigned as a set of hidden vector $H^t = [h_1^t, h_2^t, , ..., h_m^t]$, where $h_i^t \in \mathbb{R}^{L\times 1}$, t is the layer number and m is the number of comments. Initial hidden vector is assigned as evidence, i.e., $h_i^0 = \mathbf{e_i}$. For edge set N, we obtain edge attention w_{ij} via node n_i and its neighbor n_j.

$$w_{ij} = \frac{exp(W_1^{t-1}(\sigma(W_0^{t-1}(h_i^{t-1}||h_j^{t-1}))))}{\sum_{k\in N} exp(W_1^{t-1}(\sigma(W_0^{t-1}(h_i^{t-1}||h_k^{t-1}))))} \tag{2}$$

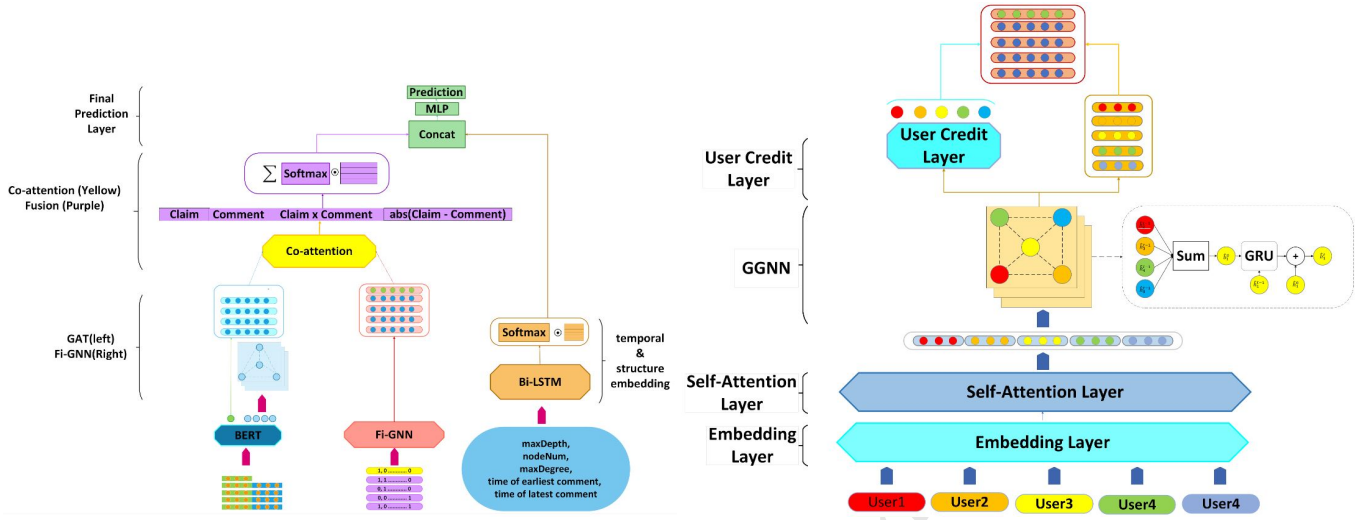
where σ is activation function, $W_0^{t-1} \in \mathbb{R}^{H\times 2L}$ and $W_1^{t-1} \in \mathbb{R}^{1\times H}$.

. Therefore, a new hidden vector h_i^t at layer t will be generated via aggregating its neighbor nodes through edges with specified attention weights,

$$h_i^t = \sum_{j\in m} w_{ij} h_i^{t-1}. \tag{3}$$

4.2 User Characteristic and Text Style Extraction

Inspired by previous research [1, 11, 27], user characteristic and text style are prove to be effective on fake news detection. Therefore, we proposed a Gated Graph Neural Networks(GGNN) based model Fi-GNN [9] which is shown as figure 1 to yield combined User style and writing style embedding. Fi-GNN is originally used on recommendation system to predict CTR, so we believe that Fi-GNN has the ability to give higher score to more reliable user. Fi-GNN includes four components: **Embedding layer**, **self-attention layer**, **GGNN** and **attention credit layer** as shown in Figure 2. The following subsections will describe each components further.



(a) GUEST (b) Fi-GNN

Figure 1: Architecture of GUEST

4.2.1 Embedding layer. In order to describe a social media user and their text style, we choose features [27] from Table 6. and Table 7 in Appendix A. User and text style are transformed into one-hot vectors and concatenated as input,

$$u_i = \overbrace{[1, 0, ..., 1]}^{User} \,||\, \overbrace{[0, 1, ..., 1]}^{TextStyle} \tag{4}$$

where $u_i \in \mathbb{R}^{T\times 1}$. Then we implement MLP as Embedding layer to convert the one-hot vectors into low dimensional user embedding vectors,

$$hu_i = \sigma W_0(u_i), \tag{5}$$

where $W_0 \in \mathbb{R}^{F\times T}$ and σ is activation function. Note that for simplicity, we denote joint embedding of user and text style as user embedding.

4.2.2 Self-Attention Layer. We employ multi-head self attention mechanism to learn dependency of each user embedding. Given user embedding set U where $hu_i \in U$, Self-Attention Mechanism is described as,

$$\begin{aligned} O_i &= Attention(Q_i, K_i, V_i) \\ Attention(Q_i, K_i, V_i) &= softmax(\frac{Q_i K_i^\top}{\sqrt{d}})V_i \\ Q_i = W_i^Q U, K_i &= W_i^K U, V_i = W_i^V U \end{aligned} \tag{6}$$

where $O_i \in \mathbb{R}^{d_i\times N}$ and $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_i\times F}$ are learned weight of head i. N is number of users. Q_i, K_i, V_i are query, key and value. Once calculating all head attention, we concatenate them to obtain a user feature representation,

$$\hat{O} = [O_1||O_2||...||O_M], \tag{7}$$

where $\hat{O} \in \mathbb{R}^{D\times N}$, $D = \sum_{i\in M} d_i$ and M is the number of heads.

4.2.3 Gated Graph Neural Networks (GGNN). To make further interaction between each user feature vector $\hat{o} \in \hat{O}$, we utilize GGNN which is composed of two parts: **GAT** and **GRU with residual connection**. GAT is implemented for node propagation while GRU with residual connection is for relieving vanishing gradient issue and reusing low-order feature.

GGNN. First, a graph $\mathfrak{G} = (N, E)$ where node set N corresponds to all users and E represents graph edge. The graph is designed as fully-connected since we would like every node to be completely interacted with others. For each node $n_i \in N$, it is assigned as a hidden state vector $\hat{H}^t = [\hat{h}_1^t, \hat{h}_2^t, ..., \hat{h}_N^t]$, where t denotes the graph layer and N is the number of source tweet user and comment users. Initial state of node is assigned as user feature vector $\hat{O}$, i.e. $\hat{H}^0 = \hat{O}$.

Node propagation. As information propagation from neighbor nodes, Fi-GNN utilizes two techniques to achieve: **Attention-Edge Weights** and **Edge-wise Transformation**. Details will be further discussed in the next paragraph.

- **Attention-Edge Weights** for each edge $\varepsilon \in E$, attention weights α_{ij} is calculated as,

$$\alpha_{ij} = \frac{exp(\sigma(W_0^{t-1}(\hat{h}_1^{t-1}||\hat{h}_j^{t-1})))}{\sum_{k\in m} exp(\sigma(W_0^{t-1}(\hat{h}_i^{t-1}||\hat{h}_k^{t-1})))} \tag{8}$$

  where $\alpha_{ij} = 0$, if $i = j$. $W_0^{t-1} \in \mathbb{R}^{1\times 2D}$.
- **Edge-Wise Transformation** [9] indicate that a fixed transformed function on all the edges is unable to model the flexible node interactions. Therefore, edge-wise transformation is proposed to achieve different transformation on different edge. We apply input matrix W_{in}^i and an output matrix W_{out}^i on each node n_i. If a node n_j propagates its information to node n_i, its representation vector will be double transformed by W_{out}^j and W_{in}^i before arriving node n_j.

Via two techniques mentioned above, node information propagation and aggregation is done as

$$b_i^t = \sum_{\varepsilon_{ij} \in E} (\alpha_{ij} W_{out}^j W_{in}^i) \hat{h}_j^{t-1} + \beta \tag{9}$$

where b_i^t is the output of GNN, W_{out}^j, $W_{in}^i \in \mathbb{R}^{D \times D}$ and $\beta \in \mathbb{R}^{1 \times D}$ are learned parameters.

GRU with residual connection. After generating new vectors from node propagation, each node representation will be updated through GGNN connected with a residual vector, which is depicted as,

$$\hat{h}_i^t = GRU(\hat{h}_i^{t-1}, b_i^t) + \hat{h}_0^t, \tag{10}$$

Detail of *GRU* is formalized as,

$$\begin{aligned} z_i^t &= \sigma(W_z b_i^t + U_z \hat{h}_i^{t-1} \beta_z) \\ r_i^t &= \sigma(W_r b_i^t + U_r \hat{h}_i^{t-1} \beta_r) \\ \tilde{h}_i^t &= tanh(W_h b_i^t + U_h (r_i^t \odot \hat{h}_i^{t-1}) + \beta_h) \\ \hat{h}_i^t &= \tilde{h}_i^t \odot z_i^t + (1 - z_i^t) \end{aligned} \tag{11}$$

where W_z, W_r, W_h, U_z, U_r, $U_h \in \mathbb{R}^{D \times D}$ are learned weights. b_z, b_r, $b_h \in \mathbb{R}^{1 \times D}$ are learned bias. z_i^t is update gate and r_i^t is reset gate.

4.2.4 Attention Credit Layer. Through Fi-GNN, we obtain final node representation $\hat{H}^T = [\hat{h}_1^T, \hat{h}_2^T, ..., \hat{h}_N^T]$. In attention credit layer layer, the purpose is to generate user credit, which can be seen as attention for each node representation. Two multilayer perceptron networks (MLP) are applied for calculating attention weights and initial score respectively, which is denoted as,

$$\begin{aligned} \widetilde{cred}_i &= MLP_1(\hat{h}_i^T) \\ \tilde{w}_i &= MLP_2(\hat{h}_i^T) \\ cred_i &= \sum_{i \in m} \tilde{w}_i \widetilde{cred}_i \end{aligned} \tag{12}$$

where $\widetilde{cred}_i$, $\tilde{w}_i$ and $cred_i \in \mathbb{R}^{1 \times 1}$ are initial credit, attention and final credit respectively.

Different from original Fi-GNN, we have to feed node representation in the following components. Thus, final credit and node representation are multiplied to generate final user feature representation, which denoted as,

$$\hat{f}_i = cred_i \odot \hat{h}_i^T \tag{13}$$

4.3 Co-Attention Fusion

[11] indicates that fake news could be detected by capturing the reciprocal effects between semantics of claims and user features via co-attention mechanism. Thus, we employ co-attention mechanism to model the interaction between sentence representation $\{\mathbf{c}, e_i\} \in S$ and user feature representation $\hat{f}_i \in \hat{F}$. First, an affinity matrix $\mathbf{T}$ is constructed as denoted,

$$\mathbf{T} = S^\top \mathbf{W_a} \hat{F} \tag{14}$$

where $\mathbf{W_a} \in R^{N \times N}$ is a learned weight and affinity matrix is used to yield interaction attention map as,

$$\begin{aligned} H_s &= tanh(\mathbf{W_s} S + (\mathbf{W_f} \hat{F}) \mathbf{W_a}^\top) \\ H_f &= tanh(\mathbf{W_f} \hat{F} + (\mathbf{W_s} S) \mathbf{W_a}) \\ \alpha^{\mathbf{s}} &= softmax(\mathbf{W_{sf}}^\top H_s) \\ \alpha^{\mathbf{f}} &= softmax(\mathbf{W_{fs}}^\top H_f) \end{aligned} \tag{15}$$

where $\mathbf{W_s} \in \mathbb{R}^{k \times L}$, $\mathbf{W_f} \in \mathbb{R}^{k \times D}$. $\alpha^s \in \mathbb{R}^{1 \times m}$ and $\alpha^f \in \mathbb{R}^{1 \times m}$ are the attention for each sentence and user embedding. We multiply attention and corresponded vectors as follows,

$$\begin{aligned} \hat{\mathbf{c}} &= \alpha_1^s \mathbf{c}, \\ \hat{\mathbf{e}}_i &= \alpha_i^s e_i, \\ \hat{f}_i &= \alpha_i^f \hat{f}_i \end{aligned} \tag{16}$$

where $i \in m$ and $i \neq 1$, and then semantic vectors and user-text vectors are concatenated as follows,

$$\begin{aligned} \widehat{\mathbf{fc}}_1 &= [\ \hat{\mathbf{c}} \ || \ \hat{f}_i\ ] \\ \widehat{\mathbf{fe}}_i &= [\ \hat{\mathbf{e}}_i \ || \ \hat{f}_i\ ] \end{aligned} \tag{17}$$

where $\widehat{\mathbf{fc}}_1$ and $\widehat{\mathbf{fe}}_i \in \mathbb{R}^{1 \times 2k}$. Since we would like to model the interaction between claim and its following comments, a fusion technique [12] is implemented to fuse claim vector $\widehat{\mathbf{fc}}_1$ and comment vectors $\widehat{\mathbf{fe}}_i$ as follows.

$$\begin{aligned} \widetilde{fa}_i &= tanh(\mathbf{W_{fa}}[\widehat{\mathbf{fc}}_1, \widehat{\mathbf{fe}}_i, (\widehat{\mathbf{fc}}_1 \odot \widehat{\mathbf{fe}}_i), |\widehat{\mathbf{fc}}_1 - \widehat{\mathbf{fe}}_i|\ ]) \\ \beta_i &= \frac{tanh(\mathbf{W}_a \widetilde{fa}_i + \mathbf{b})}{\sum_{k \in m} tanh(\mathbf{W}_a \widetilde{fa}_k + \mathbf{b})} \\ \widehat{fa} &= \sum_{i \in m} \beta_i \widetilde{fa}_i \end{aligned} \tag{18}$$

where $\mathbf{W_{fa}} \in \mathbb{R}^{p \times 4k}$, $\mathbf{W_a} \in \mathbb{R}^{1 \times p}$, $\mathbf{b} \in \mathbb{R}^{1 \times 1}$ are learned parameters. We first concatenate four different schemes shown above to get joint representation vectors $\widetilde{fa}_i$. Then attention weighted sum is applied over joint representation vectors to generate final semantics-user representation vector $\widehat{fa}$.

4.4 Structural and Temporal Information Extraction

Previous research [19] shows that structural and temporal features from hierarchical propagation networks of social media have significant effect on fake news detection. Hence, we select three structural features and two temporal features as input vector $ST = [st_1, st_2, st_3, st_4, st_5] \in \mathbb{R}^{5 \times 1}$, which shows in Table 8 in A. Then the input vector will be encoded through Bi-LSTM layer to generate structure-temporal embedding $\bar{h}_i$, which is as follows,

$$\begin{aligned} \overrightarrow{h}_i &= \overrightarrow{LSTM}(\overrightarrow{h}_{i-1}, st_i) \\ \overleftarrow{h}_i &= \overleftarrow{LSTM}(\overleftarrow{h}_{i-1}, st_i) \\ \bar{h}_i &= [\overrightarrow{h}_i || \overleftarrow{h}_i] \end{aligned} \tag{19}$$

where $\bar{h}_i \in \mathbb{R}^{2h \times 1}$ is the concatenation of hidden vectors $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ which are generated by $\overrightarrow{LSTM}$ and $\overleftarrow{LSTM}$ respectively. Once

obtaining hidden vector $\bar{h}_i$, we implement weighted sum of hidden vectors with attention as,

$$\alpha_{\mathbf{i}}^{\mathbf{sf}} = softmax(\mathbf{W_{sf}}\bar{h}_i)$$
$$\widehat{SF} = \sum_{i=1}^{5}(\alpha_{\mathbf{i}}^{\mathbf{sf}})\bar{h}_i \tag{20}$$

where $\mathbf{W_{sf}} \in \mathbb{R}^{1\times 2h}$ is a learned weight and $\widehat{SF} \in \mathbb{R}^{2h\times 1}$ is the final structure-time representation vector.

4.5 Final Prediction Layer

Purpose of this layer is to fuse semantics, user characteristics, text style, structure and time information extracted from previous components to achieve fake news detection, so We first concatenate semantics-user representation vector $\widehat{fa}$ and structure-time representation vector $\widehat{SF}$ and then employ MLP to do final prediction, which is denoted as,

$$\hat{Y} = softmax(\mathbf{W}_f[\ \widehat{fa}\ ||\ \widehat{SF}\ ] + \mathbf{b}_f) \tag{21}$$

where $\hat{Y} = [\hat{y}_0, \hat{y}_1]$ is the final prediction. $\widehat{fa}$ and $\mathbf{b}_f$ are learned parameters.

. The model is trained to minimize the cross-entropy error via loss function as,

$$Loss = -(\ ylog(\hat{y}_0) + (1-y)log(\hat{y}_1)\ )$$

where y is labeled veracity and $\hat{y}_0, \hat{y}_1 \in [0, 1]$.

5 EXPERIMENTS

As the main goal of this work is to verify truthfulness of claims through heterogeneous information and provide explainable evidence, we design experiments to answer following questions:

(1) **Can GUEST achieve equal or better performance compared to state of art models?**
(2) **How is the influence of different comment numbers on GUEST?**
(3) **What is the effectiveness of each component for addressing embedding in various spaces?**
(4) **Is GUEST able to provide proper explanation why a claim is fake?**

5.1 Datasets

We use two well-known datasets, PHEME [32] and RumourEval [3] to evaluate GUEST. Both datasets contain Twitter conversation threads discussing different events including Ferguson unrest, the crash of a Germanwings plane and etc. A Twitter conversation thread comprise a source tweet (i.e. claim), following tree structure replies (i.e. comments) and a veracity label. Since we aim to verify whether a claim is true or false, unverified label threads are eliminated. Table 1. shows details of two datasets. Note that for RumourEval, data has been split already. For PHEME, we first sort each thread into chronological order in each event since we would not like our model to use conversation threads happening later to predict a veracity of a claim happening earlier. and then separate training, development and testing data through merging each subdataset in each event split by proportion of 80%, 10%, 10%, which is follow the proportion of RumpourEval.

5.2 Settings

For both dataset, number of training epochs is set as 200, sentence embedding as 768, user-text feature embedding as 32 and structure-temporal embedding as 16. In addition, we use early stopping method depending on development data's accuracy with a patience of 30 epochs [31]. Different hyperparamters setting and training strategies are depicted as following. For **RumourEval**, learning rate is set as 0.00005, L2 regularization as 0.05 and batch size as 32. Since veracity class in RumorEval is unbalanced (approximately 7 : 3), over-sampling method [28] is implemented to soothe class imbalanced issue. For **PHEME**, learning rate is set as 0.00005, L2 regularization as 0.00005 and batch size as 64.

5.3 Performance Comparison (Q1.)

5.3.1 Baselines.

SVM. [5] This model leverages linear-SVM to do fake news classification and its inputs include bag of words and other manually extracted features.

CNN. [2]This model captures local semantic features through filter matrix and extract most important information via max pooling.

DeClarE. [15] This model first utilizes bi-LSTM attention to obtain article representation and concatenate it with embedding of claim, claim souce and article source for binary classification.

RvNN-BU. [13] This model is based on RNN to construct a tree structure neural networks which propagates information from comment branches to claim root.

RvNN-TD. [13] This model is similar to RvNN-BU. Nonetheless, Information propagates from claim root to comment branches.

BaysienDL. [29] This model implements Bayesian model to generate a distribution to represent the prediction and uncertainty. And then it encodes comments via Bi-LSTm as auxiliary information.

AIFN. [25]This model implements self-attention and gated adaptive interaction networks to fulfill deep fusion of claim and comments represented by semantic and sentimental embedding jointly.

5.3.2 Comparison Analysis. The result of performance comparison is shown in Table 2. All baseline model hyperparameters are set by defaults or values suggested from their papers. First, All model performance in PHEME is better than RumourEval and among them. There are two possible reason. First possible reason is that training data in PHEME is 7 times more than training data in RumourEval. Moreover, training data in RumourEval does not contain events of Marina Joyce, the health condition of Hillary Clinton and Ferguson, which are occupied half volume of testing data. However, For GUEST, performance on two datasets are nearly same, which show that GUEST has the ability to address data scarcity issue. Second, in RumourEval, models leveraging comments as one of inputs (i.e. DeClarE, RvNN, BURvNN-TD, BaysienDL, AIFN, GUEST) achieve at least 20% higher accuracy than models without using comments (i.e. SVM, CNN), which indicates that comments information is effective for fake news detection. Third, DeCarE gets the second highest accuracy in both datasets, which may because DeClareE considers claim source and article source as well. Therefore, the credibility of sources is also considered as a criteria to assist model in classifying fake news. Fourth, in RumourEval and PHEME, our model exceeds state of art models by 15.0% and 4.4% in accuracy which infers two

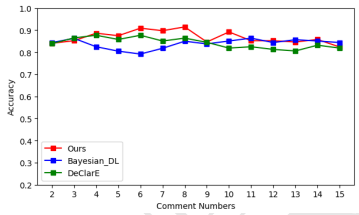
Datasets	Data Split	Threads	Replies	True	False
RumorEval	Training	177	2530	127	50
	Development	22	198	10	12
	Testing	20	822	8	10
Pheme	Training	1369	15883	826	543
	Development	160	2068	105	55
	Testing	176	2040	136	40

Table 1: data distribution of RumorEval and PHEME

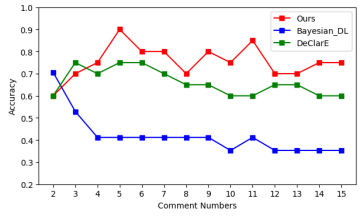
insights. First, the result demonstrates effectiveness in graph embedding on both semantics and user character features. Second, we find an useful way to exert multiple heterogenious features vector on fake news detection news task.

5.4 Comment Numbers Analysis (Q2.)

Figure 2 reveals the result of different comments for GUEST in PHEME and RumourEval. For PHEME, accuracy is more stable as number of comments increases comparing with accuracy in RumourEval. For RumourEval, accuracy improves in consistency from 2 to 5 comments, but it starts to oscillate between 75% and 90% afterwards. The reason of this oscillation may because RumourEval only possesses one-ninth as much data as PHEME possesses. Hence, redundant comments bring extra noise and then cause over-fitting problem. However, comparing with DeClarE and BayseinDL, oscillation is less dramatic, which infers that our model is capable of filtering out useful comments as evidences.



(a) PHEME



(b) RumorEval

Figure 2: Comment Numbers Comparison

5.5 Ablation Study (Q3.)

Table 3 demonstrates ablation study experiment and acronym of each component (i.e. *G/FG/CA/AF/TS/AA*). from *G*, *FG* and *G/FG/AA*, it indicates that simple weighted sum of semantics and user embedding (i.e. *G/FG/AA*) will gain worse result than only leveraging *G* or *FG*. Therefore, we will discuss effectiveness of *CA*, *AF* and *ST*.

Co-Attention (CA). Comparing *G/FG/AA* with *G/FG/CA*, accuracy improves 18% and 10% respectively, which shows that *CA* has the ability to capture significant features of sentence and user-text information.

Attention-Fusion (AF). First we compare *G/FG/AF* with *G/FG/CA* and find that the model with *AF* is higher than model with *CA* in both dataset. Furthermore, in both datasets, Adding *AF* after *CA* increases accuracy of 4% and 10%. As a result, *Attention-Fusion* is proved to be effective on enhancing interaction between claim and comments.

Sructural-Temporal Embedding (ST). The result in PHEME indicates that *G/FG/CA/AF/ST* reaches higher accuracy (91.4%) than accuracy (85.7%) from *G/FG/CA/AF*. From this result, we are able to conclude that structural and temporal information is beneficial for fake news detection task.

5.6 Explainability Analysis (Q4.)

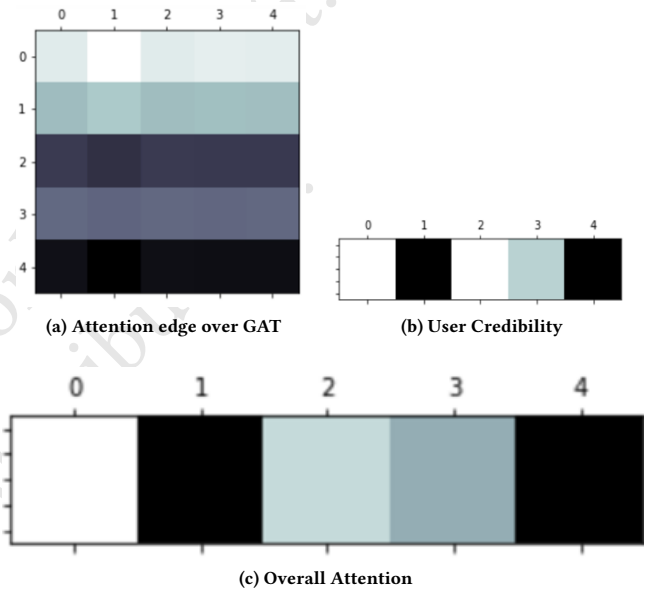
Table 4 and Table 5 illustrate cases of false claims and their related comments while Figure 3 and Figure 4 are the corresponding visualization of varied attention. From Table 4 and Figure 3, We can see that 1) From semantic perspective in Figure 3(a), **[*Comment 0*]** and **[*Comment 1*]** which respectively express denying and querying stance gain first and second highest attention. This shows that GUEST considers these two comments as strong evidences to be against the veracity of the claim. 2) From user perspective in Figure 3(b), GUEST gives highest score to **[*User 2*]**, while **[*User 4*]** obtains the lowest score. It is because **[*User 2*]** is the only one contains user-Geo information and text length of **[*User 4*]** is too short. 3) in Figure 3(c), GUEST considers semantics and user information to generate overall attention. As obtaining the highest semantic attention and user credibility, **[*Comment 0*]** is regarded as a strong evidence to decide that the claim is false, while **[*Comment 4*]** is a weak evidence since it obtains the least semantic attention and user credibility. From Table 5, Figure 4, we can discover similar results. **[*Comment 4*]** and **[*Comment 2*]** receives high semantic attention and user credibility. Hence, they are considered as evidences to query the veracity. of claim.

Table 2: Performance Comparison

		Model							
Datasets	Metrics	SVM	CNN	DeClarE	RvNN-BU	RvNN-TD	BaysienDL	AIFN	GUEST
RumorEval	Accuracy	0.500	0.450	0.750	0.700	0.750	0.706	0.750	**0.900**
	Precision	0.720	0.710	0.807	0.647	0.807	0.7153	0.807	**0.930**
	Recall	0.580	0.540	0.791	0.750	0.791	0.734	0.791	**0.880**
	F1-score	0.670	0.370	0.749	0.700	0.749	0.580	0.749	**0.890**
Pheme	Accuracy	0.834	0.835	0.870	0.785	0.806	0.863	0.763	**0.914**
	Precision	0.837	0.767	0.809	0.647	0.753	**0.890**	0.656	0.881
	Recall	0.818	0.814	0.779	0.588	0.791	0.683	0.629	**0.874**
	F1-score	0.825	0.785	0.793	0.672	0.749	0.726	0.683	**0.877**

Table 3: Component Analysis. Acronym of Component: *G*:GAT, *FG*:Fi-GNN, *CA*:Co-Attention, *AF*:Attention-Fusion, *TS*:Temporal-Structure Embedding, *AA*:Attention-Aggregator [31]

Dataset	Compoment	Accuracy
PHEME	G/FG/CA/AF/TS	0.914
	G/FG/CA/AF	0.857
	G/FG/AF	0.840
	G/FG/CA	0.823
	G/FG/AA	0.647
	FG	0.704
	G	0.727
RumourEval	G/FG/CA/AF/TS	0.900
	G/FG/CA/AF	0.750
	G/FG/AF	0.650
	G/FG/CA	0.650
	G/FG/AA	0.550
	FG	0.600
	G	0.400



(a) Attention edge over GAT (b) User Credibility

(c) Overall Attention

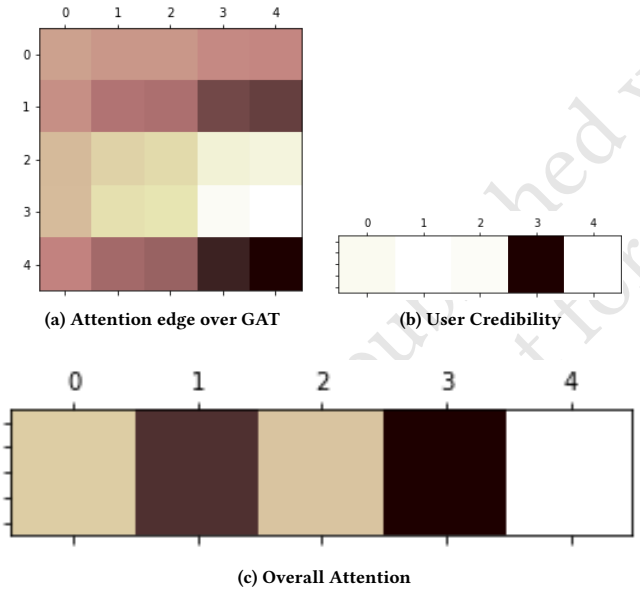
Figure 3: Visualization of attention map, User Credibility and overall comment attention. These three visualization map are respectively generated by GAT, Fi-GNN and Co-attention fusion.

Table 4: A case of false claim with following comments. [*User*] means user features encoded by one-hot vector according to table 6 and table 7.

Veracity: False
[*Claim*]: GERMAN NEWS: Co-Pilot of Germanwings Airbus Was MUSLIM CONVERT …'Hero of Islamic
[*Comment 0*]: true. But I'm not buying the story put forth thus far. There's more to this. [*User 0*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 1, 0, 0, 1, 1]
[*Comment 1*]: Then, it really doesn't mean anything right. [*User 1*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 1, 0, 0, 1, 1]
[*Comment 2*]: won't be hearing about this in the #MSM. MSM is full of twisted, sick, and liberals willing to defend like minded [*User 2*] : [0, 1, 1, 2, 2, 2, 0, 1, 0, 0, 1, 0, 0, 1, 1]
[*Comment 3*]: No he wasn't, stop linking to Pam Geller. [*User 3*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 1, 0, 0, 1, 1]
[*Comment 4*]: yes, yes they do... [*User 4*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 0, 0, 0, 1, 1]

Table 5: A case of false claim with following comments. [*User*] means user features encoded by one-hot vector according to table 6 and table 7.

Veracity: False
[*Claim*]: BREAKING: Prince confirms he is playing a surprise show tonight in the pillows section of Toronto's former Big Bop concert hall.
[*Comment 0*]: :([*User 0*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 0, 0, 0, 1, 1]
[*Comment 1*]: It was too good to pass up! Besides we all know if there's ever a show at the Big Bop, I will be headlining with all my friends. [*User 1*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 1, 0, 0, 1, 1]
[*Comment 2*]: For real? [*User 2*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 0, 0, 0, 1, 1]
[*Comment 3*]: no, that tweet was a joke. [*User 3*] : [0, 0, 1, 2, 2, 2, 0, 1, 0, 0, 0, 0, 0, 1, 1]
[*Comment 4*]: quit playing games with my heart, rob. [*User 4*] : [0, 1, 1, 2, 2, 2, 0, 1, 0, 0, 0, 0, 0, 1, 1]



(a) Attention edge over GAT

(b) User Credibility

(c) Overall Attention

Figure 4: Visualization of attention map 2, User Credibility and overall comment attention. These three visualization map are respectively generated by GAT, Fi-GNN and Co-attention fusion.

6 CONCLUSION

In conclusion, we propose a novel neural network model GUEST for fake news detection based on graph network and attention mechanism. GUEST first obtains sentence embedding and user feature embedding via two graph encoder and then exerts co-attention method to involve this two kind embeddings into interaction. In the end, we fuse sentence, user feature and structural-temporal embeddings to predict the truthfulness of a Twitter claim. Experiments results confirm the effectiveness and expainability of our model. For future work, GUEST will be tried on other tasks, such as stance detection and sentimental analysis. Furthermore, we will extend our model through how to eliminate re-sampling data preprocessing while maintain or exceed current performance.

7 ACKNOWLEDGMENTS

REFERENCES

[1] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011*, Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar (Eds.). ACM, 675–684. https://doi.org/10.1145/1963405.1963500

[2] Yi-Chin Chen, Zhao-Yang Liu, and Hung-Yu Kao. 2017. IKM at SemEval-2017 Task 8: Convolutional Neural Networks for stance detection and rumor verification. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 465–469. https://doi.org/10.18653/v1/S17-2081

[3] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. arXiv:1704.05972 [cs.CL]

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] Omar Enayet and Samhaa R. El-Beltagy. 2017. NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter.. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 470–474. https://doi.org/10.18653/v1/S17-2082

[6] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG]

[7] Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task Learning for Rumour Verification. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3402–3413. https://www.aclweb.org/anthology/C18-1288

[8] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2017. Gated Graph Sequence Neural Networks. arXiv:1511.05493 [cs.LG]

[9] Zekun Li, Zeyu Cui, Shu Wu, Xiaoyu Zhang, and Liang Wang. 2019. Fi-GNN: Modeling Feature Interactions via Graph Neural Networks for CTR Prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 539–548. https://doi.org/10.1145/3357384.3357951

[10] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained Fact Verification with Kernel Graph Attention Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7342–7351. https://doi.org/10.18653/v1/2020.acl-main.655

[11] Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 505–514. https://doi.org/10.18653/v1/2020.acl-main.48

[12] Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. Sentence-Level Evidence Embedding for Claim Verification with Hierarchical Attention Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2561–2571. https://doi.org/10.18653/v1/P19-1244

[13] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1980–1989. https://doi.org/10.18653/v1/P18-1184

[14] Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2016. Credibility Assessment of Textual Claims on the Web. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery,

New York, NY, USA, 2173–2178. https://doi.org/10.1145/2983323.2983661

[15] Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 22–32. https://doi.org/10.18653/v1/D18-1003

[16] Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2931–2937. https://doi.org/10.18653/v1/D17-1317

[17] Yuxiang Ren and Jiawei Zhang. 2020. HGAT: Hierarchical Graph Attention Network for Fake News Detection. arXiv:2002.04397 [cs.SI]

[18] Kai Shu, Limeng Cui, Suhang Wang, Dongwon Lee, and Huan Liu. 2019. DEFEND: Explainable Fake News Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 395–405. https://doi.org/10.1145/3292500.3330935

[19] Kai Shu, Deepak Mahudeswaran, Suhang Wang, and Huan Liu. 2019. Hierarchical Propagation Networks for Fake News Detection: Investigation and Exploitation. arXiv:1903.09196 [cs.SI]

[20] Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond News Contents: The Role of Social Context for Fake News Detection. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (Melbourne VIC, Australia) *(WSDM '19)*. Association for Computing Machinery, New York, NY, USA, 312–320. https://doi.org/10.1145/3289600.3290994

[21] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 809–819. https://doi.org/10.18653/v1/N18-1074

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[23] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML]

[24] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, 647–653. https://doi.org/10.18653/v1/P17-2102

[25] Lianwei Wu and Yuan Rao. 2020. Adaptive Interaction Fusion Networks for Fake News Detection. arXiv:2004.10009 [cs.CL]

[26] Lianwei Wu, Yuan Rao, Haolin Jin, Ambreen Nazir, and Ling Sun. 2019. Different Absorption from the Same Sharing: Sifted Multi-task Learning for Fake News Detection. arXiv:1909.01720 [cs.CL]

[27] Lianwei Wu, Yuan Rao, Yongqiang Zhao, Hao Liang, and Ambreen Nazir. 2020. DTCA: Decision Tree-based Co-Attention Networks for Explainable Claim Verification. arXiv:2004.13455 [cs.CL]

[28] Yuzhe Yang and Zhi Xu. 2020. Rethinking the Value of Labels for Improving Class-Imbalanced Learning. arXiv:2006.07529 [cs.LG]

[29] Qiang Zhang, Aldo Lipani, Shangsong Liang, and E. Yilmaz. 2019. Reply-Aided Detection of Misinformation via Bayesian Deep Learning. *The World Wide Web Conference* (2019).

[30] Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning Over Semantic-Level Graph for Fact Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 6170–6180. https://doi.org/10.18653/v1/2020.acl-main.549

[31] Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 892–901. https://doi.org/10.18653/v1/P19-1085

[32] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PLOS ONE* 11, 3 (Mar 2016), e0150989. https://doi.org/10.1371/journal.pone.0150989

A APPENDIX

User Character	Criterion	Score
verified	True	1
	False	0
geo_enabled	True	1
	False	0
screen_name	True	1
	False	0
profile_use	True	1
_background_image	False	0
followers_count	[0, 100)	0
	[100, 500)	1
	[500)	2
friends_count	[0, 100)	0
	[100, 200)	1
	[200)	2
favourites_count	[0, 100)	0
	[100, 200)	1
	[200)	2

Table 6: User Characteristics

Writing Style	Criterion	Score
Geo	True	1
	False	0
Source	True	1
	False	0
favorite	True	1
	False	0
favorite_count	[0, 100)	0
	[100)	1
text_length	[0, 10)	0
	[10)	1
url	True	1
	False	0
media	True	1
	False	0
hashtag	True	1
	False	0
user mentioned	True	1
	False	0

Table 7: Text Style

Perspective	Network feature
Structural	Tree depth
	Number of nodes
	Maximum Outdegree
Temporal	Time between claim and the earliest reply node
	Time between claim and the earliest reply node

Table 8: structural and temporal features
Structural features involves patterns of conversation among claim and following comments. Temporal features are for capturing exchanging opinion behavior in terms of time.