A Progressive Framework for Role-Aware Rumor Resolution

Lei Chen[1], **Guanying Li**[2],**Zhongyu Wei**[1,3]*, **Yang Yang**[4], **Baohua Zhou**[2], **Qi Zhang**[5], **Xuanjing Huang**[5]
[1] School of Data Science, Fudan University, China
[2] School of Journalism, Fudan University, China
[3] Research Institute of Intelligent and Complex Systems, Fudan University, China
[4] College of Computer Science and Technology, Zhejiang University, China
[5] School of Computer Science, Fudan University, China
[1,3,5]{chenl18,zywei,qi_zhang,xjhuang}@fudan.edu.cn
[2]zhoubaohua@yeah.net;[4]yangya@zju.edu.cn

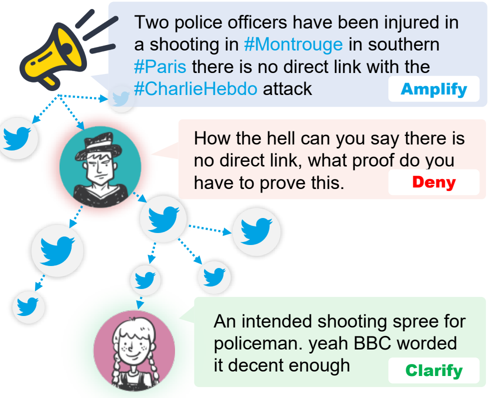
Abstract

Existing works on rumor resolution have shown great potential in recognizing word appearance and user participation. However, they ignore the intrinsic propagation mechanisms of rumors and present poor adaptive ability when unprecedented news emerges. To exploit the fine-grained rumor diffusion patterns and generalize rumor resolution methods, we formulate a predecessor task to identify triggering posts, and then exploit their characteristics to facilitate rumor verification. We design a tree-structured annotation interface and extend PHEME dataset with labels on the message level. Data analysis shows that triggers play a critical role in verifying rumors and present similar lingual patterns across irrelevant events. We propose a graph-based model considering the direction and interaction of information flow to implement role-aware rumor resolution. Experimental results demonstrate the effectiveness of our proposed model and progressive scheme.

1 Introduction

With the expansion of the Internet, online information tends to spread quickly and widely including fake news, misinformation and rumors, the last of which is defined as circulating stories unverifiable or deliberately false (DiFonzo and Bordia, 2007). Especially in current situation with infectious epidemics and intensive international relationships, researchers have witnessed more than 900% growth in the number of English fact-checks during the COVID-19 outbreak. (Brennen et al., 2020). Automatically verifying rumors has become an urgent need for individuals and society.

Conventional methods for rumor resolution depend on exploiting the evolutionary characteristics of content and spreaders (Kwon et al., 2013; Ma et al., 2015). Benefiting from various attention mechanisms, there is a growing tendency to retrieve evidential messages, indicative tokens and critical users to enhance interpretability (Ma et al., 2019; Lu and Li, 2020; Wu et al., 2020a). However, capturing patterns from historical records faces great challenges while transferring to unprecedented events as rumors evolve quickly and recur infrequently. Besides, extracting features of malicious users skips over the immediate indication and allusive roles of the content itself.



Two police officers have been injured in a shooting in #Montrouge in southern #Paris there is no direct link with the #CharlieHebdo attack

How the hell can you say there is no direct link, what proof do you have to prove this.

An intended shooting spree for policeman. yeah BBC worded it decent enough

Figure 1: An illustration of rumor cascades and typical roles of messages helpful for rumor verification.

Recently, researchers are dedicated to investigating the intrinsic mechanism of rumor propagation rather than linguistic or rhetorical features. Vosoughi et al. (2018) reveal that it is the *information novelty* that stimulates discussion desire which makes rumors spread faster and deeper. Choi et al. (2020) manage to locate *echo chamber members* who tend to amplify rumor threads and find them responsible for viral propagation. N. Zehmakan and Galam (2020) divide rumor participants into 3 groups (*seeds* who adamantly convince the truth, *agnostics* who firmly reject and *others*) and analyze their roles during diffusion. Inspired by these sociological findings, we propose to explore and model different roles of messages as rumor evolves.

Figure 1 illustrates three types of messages with

[*] Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 2748–2758
October 12–17, 2022.

triggering effects via a cascade instance drawn from PHEME dataset (Zubiaga et al., 2016; Kochkina et al., 2018). The source tweet seemingly reports objectively and *amplifies* discussion topics regard to a shooting event. After several rounds of retweet, someone presents an attitude of *denying* and asks for evidence. Finally, a user comes out to *clarify* the real condition and confirm the falseness of the source. Suchlike online discussion is widely existed, however, only a few messages present these critical roles and the majority are insignificant reposts and comments. Accordingly, our goal is to identify ***triggers***, i.e., messages that have prominent effects on rumor proliferation and dominate the judgment of cascade credibility. We also claim that identifying the *role-aware* propagation mode will contribute to rational and sound rumor verification.

To practice the idea, we formulate the task of trigger identification and annotate all the messages in PHEME to form a jointly labeled dataset. Based on the well-adopted graph learning methods, we further put forward the UGRN (**U**nsymmetric **G**raph **R**ecurrent **N**etworks) framework by additionally considering the direction and interaction of information flow to simulate trigger effect. Moreover, we devise the role-aware integration and warm-up strategy to facilitate rumor verification.

Our contributions are of three-folds:

- Following the propagation mechanism of rumors, we formulate trigger identification as a prepositive task of rumor verification and supplement message-level annotations to the PHEME dataset.
- What's more, we design a graph-based framework to jointly identify triggers and verify rumors by progressively modeling tree-structured rumor cascades.
- Taking advantage of our annotated dataset, extensive experiments are conducted and demonstrate the effectiveness of our model.

2 Related Work

As our research focuses on technically resolving rumors in social media, we relate it with existing computational rumor resolution methods and our motivation for applying graph neural networks.

2.1 Rumor Resolution

Since rumors online are enriched with multi-form data, early work concentrates on extracting prominent features from perspectives of content, user reliability and communication impact (Castillo et al., 2011; Kwon et al., 2013; Liu et al., 2015). With deeper understanding of rumor propagation, researchers endeavor to model the whole cascade considering it as time-series sequences (Ma et al., 2015, 2016; Yu et al., 2017), tree-structured diffusion networks (Ma et al., 2018b; Kumar and Carley, 2019; Wei et al., 2019; Ma and Gao, 2020; Li et al., 2020) or the combination of both (Sun et al., 2022).

A remarkable progress lately is to exploit stance information to enhance rumor verification. Kochkina et al. (2018) treat rumor detection, stance classification and rumor verification as a consistent pipeline and confirm the effectiveness of multi-task learning. Following studies explore various mechanism of parameter sharing (Ma et al., 2018a; Wu et al., 2019) and improve efficiency of data usage (Yu et al., 2020). However, annotating stances in all rumor cascades is labor-intensive thus existing corpus cannot reach a perfect match between stance and verification data, which requires superior multitask training skills and still fails to explain how stance information instructs rumor verification.

Another trend is to excavate the explainability of rumor resolution networks. Ma et al. (2019) utilize hierarchical attention networks to locate evidential sentences. Lu and Li (2020) employ a graph-based co-attention model to capture the relevance between the source text and spreader behavior. Wu et al. (2020a) select suspicious retweets and apply co-attention mechanisms to explore their relationship with the source at token level. Although these approaches can identify unreliable words, sentences and users, they only practice in range-fixed and randomly-split corpus making them short of stability when faced with unknown events.

Recently, researchers attempt to model rumor cascades based on the widely-existed propagation mechanism. Wu and Rao (2020) employ gated mechanisms and devise adaptive interaction fusion networks to model the emotional associations and semantic conflicts which rationalize rumor verification. Chen et al. (2020a) utilize discrete variational autoencoders to model interaction between messages and capture their temporal evolution. Lin et al. (2021) design hierarchical graph attention networks to implement claim-guided rumor detection. Different from their work, our goal is to explicitly identify critical messages with triggering effects so that we construct a jointly labeled dataset.

What's more, we also investigate how triggers progressively facilitate rumor resolution.

2.2 Graph Neural Networks

With increasing complexity of data structure and ingenious construction of intrinsic relation, Graph Neural Networks (GNNs) have gained incremental popularity in modeling topological or tree-structured data (Wu et al., 2020b). Among all the variants, the basic skeleton Graph Convolutional Networks (GCNs) exploit structure information to aggregate and share features of neighbors which provides a rapid and effective solution for node classification, link prediction and community detection (Kipf and Welling, 2017). Following works mainly focus on refined aggregation of adjacent nodes, such as incorporating attention mechanism (Velickovic et al., 2018) and sampling neighbors to avoid over-smoothing and improve computation efficiency (Hamilton et al., 2017).

Nowadays, GNNs are extensively applied in the area of natural language processing, ranging from syntax-based machine translation (Bastings et al., 2017), knowledge-based question answering (Saxena et al., 2020) and aspect-level sentiment classification (Chen et al., 2020b). Despite the tree structure owned by rumor cascades, it is difficult to model cascades directly via GNNs. On one hand, propagation graphs present a high level of heterogeneity in which neighboring nodes usually possess different roles, while most GNNs are only effective for homogenous node classification by sharing mutual features. On the other hand, traditional ways of entire graph learning apply mean or attention pooling which is too coarse to capture evolutionary characteristics of rumor cascades. In this paper, we come up with an innovative way of message passing by inheriting the pioneering idea of Gated Graph Neural Networks (Li et al., 2016) that update node representation via gated recurrent unit, but also considering the direction and integration of information flow.

3 Task and Dataset

3.1 Task Formulation

The task of **rumor verification** is formulated as a supervised classification problem on the cascade level. Given a source tweet r_0, the tree-structured cascade can be constructed with its responsive tweets $\{r_1, r_2, ..., r_T\}$ following the retweet relationship while their textual, temporal and user-related features are available. The goal is to assess the veracity of the cascade by classifying $\mathcal{Y}^v$ into *true*, *false* or *unverified.*

In this paper, we propose a progressive framework to implement **trigger identification** during verification which aims at recognizing the role of each tweet $\mathcal{Y}_i^t$ as *amplify*, *deny*, *clarify* or *null*. *Amplify* indicates tweets that initiate new concerns or enlarge the discussion scale related to the social event. *Deny* means presenting doubt or rejection towards previous messages. *Clarify* introduces factual or substantial information. Other messages are left as *null* which means they are insignificant for rumor propagation or verification.

3.2 Dataset Construction

Our corpus is built on PHEME dataset released by Zubiaga et al. (2016) including TWITTER threads from 5 hot-debated social events. Although subsequently Zubiaga et al. (2018) expand the total event amount to 9, the additional cascades are small-scaled and extremely unbalanced, thus we only consider the original 5 events. They also supplement stance labels on the message level, but only 13% rumor cascades have been annotated limited by visualization technique and labor resources.

To implement role-aware rumor resolution, we annotate triggers for all the messages in rumor cascades. The main difference between two types of message-level labels is that stance just presents sentiment polarity towards the source tweet, while triggers imply their global roles for rumor evolution and are more context-sensitive.

The annotation process consists of three steps. First, We devise a tree-based annotation system containing textual information and propagation path[1]. We remove cascades that only contain source tweet and drop messages missing parents. Second, each cascade is sent to 3 undergraduates who need to read all the tweets in the cascade to understand how the circulating story develops, and then assign trigger labels to messages with critical roles. We only adopt the label if more than 2 people reach an agreement. Other messages are labeled as *null* to ensure the significance of triggering effects. Finally, we evaluate annotation quality with Fleiss's kappa coefficient (Fleiss, 1971) and achieve a moderate agreement of 0.515. Annotating triggers is challenging because social me-

[1] http://fudan-disc.com/project/annotation/propagation/demo.html

dia statements are full of abbreviations and slang words. The incompleteness of cascades caused by privacy restrictions also impedes global comprehension. Statistics of the extended dataset is shown in Table 1.

event	# of cas.	# of mes.	verify dist. (F:T:U)	trigger dist. (N:A:C:D)
CH	449	6110	114:187:148	4705:915:271:219
OS	467	6036	72:327:68	4793:868:254:121
SS	508	7832	76:378:54	5868:1050:471:443
FG	268	4516	8:9:251	3679:527:181:129
GW	237	2377	111:94:32	1762:388:147:80
All	1929	26871	381:995:553	20807:3748:1324:992

Table 1: Statistics of extended PHEME dataset. The abbreviation of different events is in short of *Charlie Hebdo*, *Ottawa Shooting*, *Sydney Siege*, *Ferguson Unrest*, *Germanwings Crash* respectively. The next two columns represents the amount of cascades and messages involved in different events. As for distribution of verification and trigger labels, capital letters stand for possible categories (F: *false*, T: *true*, U: *unverified*, N: *null*, A: *amplify*, C: *clarify*, D: *deny*).

3.3 Data Analysis

For purpose of exploring how triggers interact with neighbors and affect rumor proliferation, we analyze their contextual content continuity and capture their temporal characteristics as rumor develops.

Content Continuity. Since triggers are assumed to be more context-sensitive, we attempt to measure the similarity between successive messages using the ratio of overlapped word count to the longer sentence length. As propagation is irreversible, we differentiate the information flow either from parent nodes or child nodes. Specifically, after calculating the similarity of all the message pairs, we take the average of child posts as the similarity with children. Figure 2 shows the content continuity for different types of triggers in different events.

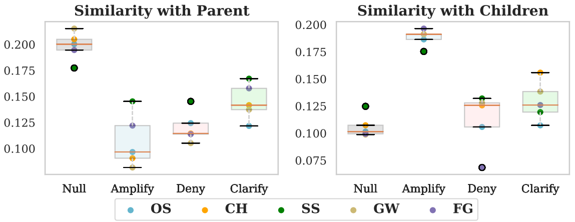


Figure 2: Context similarity for different types of triggers in different events. Scatter points represent the averaged context similarity for a certain kind of trigger in a specific event. Shapes of boxes depict the degree of trigger assimilation for different events.

It shows that triggers with different background tend to possess similar continuity property as the length of all the boxes is comparatively short. The type of *amplify* holds higher probability to bring information novelty compared with its parent and launch discussion associated with it, while the *null* type is totally on the contrary. Words in *deny* posts are less repeated in parent and child message. Posts of *clarify* present a moderate similarity with both parent and children partly because factual information is usually targeted towards the preceding content but also provides hints for further debate.

In addition, we can observe that the property of triggers is naturally endowed no matter what social event they are related to, and the same for their prior categorical distribution (shown in Table 1). Hence, we consider triggers hold higher transfer ability focusing on the universally existed propagation patterns instead of concrete topics or stories, which is helpful to debunk rumors nonexistent in history.

Temporal Variation. In order to investigate what role triggers play for verifying rumors, we calculate the amount of triggers in different diffusion stages. To ensure the amount declination is not from cascades rather small, we select 1,297 cascades whose conversation last for more than 30 minutes and count the number of different triggers in every 3 minutes. Then we count how many triggers of a certain type emerge in each evolution stage for every cascade. As shown in Figure 3, the y-axis represents the averaged number of specific triggers for each cascade in the time interval.

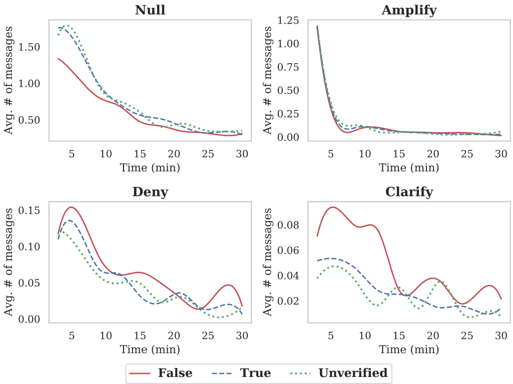


Figure 3: Temporal variation of trigger distribution. Each subgraph represents a certain kind of trigger. Different line styles stand for the category of rumors.

On the whole, the majority of discussion burst in early stages. Except the *amplify*, other types of triggers are distinguishable in different rumors.

Amount of *null* is relatively small in misinformation especially in early stages which means there exist more triggering posts arguing about cascade veracity. *Deny* and *clarify* appear more frequently in false rumors, while *clarify* takes longer time to fade away. With all the findings, we assume trigger identification as an effective way to promote verification and generalization for rumor resolution.

4 Proposed Model

Based on observations in previous section, we propose the **U**nsymmetric **G**raph **R**ecurrent **N**etworks (UGRN) to identify triggers and progressively verify rumors. Figure 4 illustrates the overall architecture which is composed of two components, the sharing GRN layers of two tasks and the trigger-aware prediction module.

4.1 Unsymmetric Graph Recurrent Networks

We use pretrained model to encode textual information for each tweet, and then decompose the propagation tree as two unsymmetric adjacency matrix to employ different GRN layer for interaction direction control.

Graph Initialization. Following the online conversational records, each rumor cascade can be constructed as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ represents the set of nodes (messages in this circumstance) and $\mathcal{E}$ represents the set of edges (retweet relationship). Node representation is initialized via the pretrained BERTweet (Nguyen et al., 2020) and fine-tuned afterwards. We directly take the final representation of `[CLS]` token $\boldsymbol{s}$ as the semantically meaningful features of sentences.

Structure Decomposition. After representing messages with pretrained model, node attributes $\mathbf{X} \in \mathbb{R}^{d\times|\mathcal{V}|}$ can be obtained by concatenating $\{\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_{|\mathcal{V}|}\}$, where d is the dimension of sentence embedding and $|\mathcal{V}|$ represents the total amount of tweets in the cascade.

Generally, edges are represented with a symmetric adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 1$ if there exists an edge between node i and j. However, implementing graph convolution in this way ignores the direction of information flow which is assumed prominent for classifying triggers (denoted in section 3.3). Therefore, we differentiate direction of information flow by decomposing the original adjacency matrix into two unsymmetric matrices $\mathbf{A}^p$ and $\mathbf{A}^c$, where $\mathbf{A}^p_{ij} = 1$ if child node i is connected with parent node j and $\mathbf{A}^c_{ij} = 1$ if parent i is connected with child j.

Since the decomposed adjacency matrix is sparse especially for tree-structured data, we add self-loops to the root (the source tweet) in $\mathbf{A}^p$ and all the leave nodes (the last tweet of propagation path) in $\mathbf{A}^c$ to ensure the sum for each row is larger than 0, thus can be divided for normalization. Figure 4 shows a concrete case to construct $\mathbf{A}^p$ and $\mathbf{A}^c$ for a specific cascade. Then we employ unsymmetric normalized transformation $\hat{\mathbf{A}}^p = (\mathbf{D}^p)^{-1}\mathbf{A}^p$ and $\hat{\mathbf{A}}^c = (\mathbf{D}^c)^{-1}\mathbf{A}^c$ to avoid value scale changing after graph convolution, where $\mathbf{D}^p, \mathbf{D}^c \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ are the diagonal degree matrices where $\mathbf{D}_{ii}$ equals to the row sum of the adjacency matrix.

Graph Reccurent Networks. Our GRN layer is based on the idea of Gated Graph Neural Networks (Li et al., 2016) but we also employ efficient way of graph convolution (Kipf and Welling, 2017). The GRNs can be extended to L layers, the l^{th} GRN layer ($l \in [1, L]$) can be represented as follows.

First, we utilize the normalized unsymmetric adjacency matrix ($\hat{\mathbf{A}}^p$ for example) to aggregate neighbor information from the top-down direction to acquire the intermediate state for node i,

$$\boldsymbol{h}_{i,p}^{(l-1),p} = \sum_{j\in\{j|\hat{\mathbf{A}}^p_{ij}\neq 0\}} \hat{\mathbf{A}}^p_{ij}\boldsymbol{h}_j^{(l-1),p} \quad (1)$$

where $\boldsymbol{h}$ with a subscript of p means the intermediate information aggregated from parent and the right $\boldsymbol{h}$ without p as subscript is the final output of the $(l-1)^{\text{th}}$ GRN layer.

Then we employ the Long Short-Term Memory unit (Hochreiter and Schmidhuber, 1997) to recurrently implement graph convolution and obtain the output $\boldsymbol{h}_i^{l,p}$ of l^{th} GRN layer,

$$\boldsymbol{f}_i^l = \sigma_g\left(\boldsymbol{W}_f\boldsymbol{h}_i^{(l-1),p} + U_f\boldsymbol{h}_{i,p}^{(l-1),p} + \boldsymbol{b}_f\right) \quad (2)$$

$$\boldsymbol{i}_i^l = \sigma_g\left(\boldsymbol{W}_i\boldsymbol{h}_i^{(l-1),p} + \boldsymbol{U}_i\boldsymbol{h}_{i,p}^{(l-1),p} + \boldsymbol{b}_i\right) \quad (3)$$

$$\boldsymbol{o}_i^l = \sigma_g\left(\boldsymbol{W}_o\boldsymbol{h}_i^{(l-1),p} + \boldsymbol{U}_o\boldsymbol{h}_{i,p}^{(l-1),p} + \boldsymbol{b}_o\right) \quad (4)$$

$$\tilde{\boldsymbol{c}}_i^l = \sigma_c\left(\boldsymbol{W}_c\boldsymbol{h}_i^{(l-1),p} + \boldsymbol{U}_c\boldsymbol{h}_{i,p}^{(l-1),p} + \boldsymbol{b}_c\right) \quad (5)$$

$$\boldsymbol{c}_i^l = \boldsymbol{f}_i^l \circ \boldsymbol{c}_i^{(l-1)} + \boldsymbol{i}_i^l \circ \tilde{\boldsymbol{c}}_i^l \quad (6)$$

$$\boldsymbol{h}_i^{l,p} = \boldsymbol{o}_i^l \circ \sigma_c\left(\boldsymbol{c}_i^l\right) \quad (7)$$

where $\boldsymbol{W} \in \mathbb{R}^{m\times h}$, $\boldsymbol{U} \in \mathbb{R}^{m\times h}$ and $\boldsymbol{b} \in \mathbb{R}^h$ (m is the input size and h is the hidden size) are weight

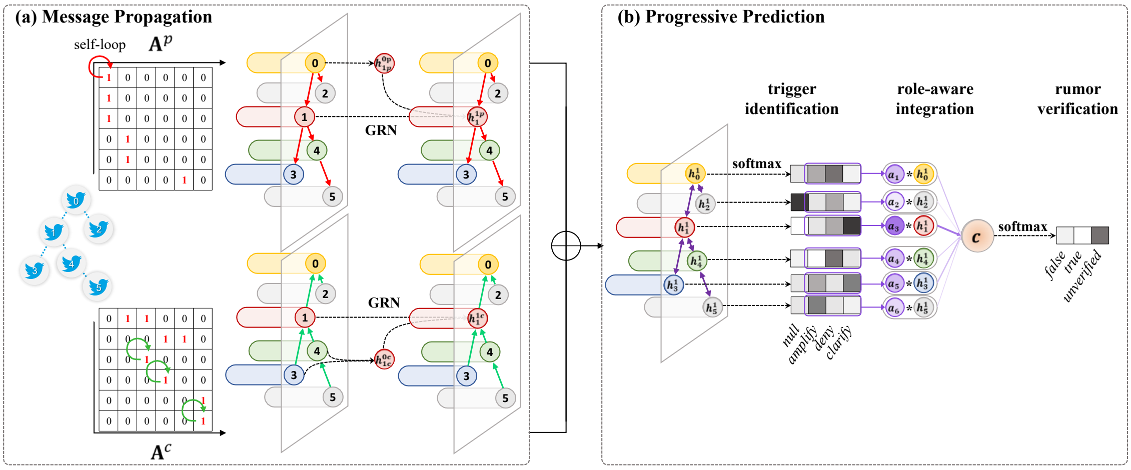


Figure 4: Overall architecture of our proposed model. The two squares on the left represent the decomposed adjacency matrices $\mathbf{A}^p$ and $\mathbf{A}^c$ that control the direction of information flow. Both of the two tasks share the unsymmetric GRN Layers. The updated node representation is used to predict trigger labels. Role-aware integration mechanism is then applied to acquire cascade representation and produce verification prediction.

matrices and bias vector, σ_g and σ_c represent sigmoid and hyperbolic tangent activation functions.

Computing $\boldsymbol{h}_i^{l,c}$ which integrates information from child nodes is identical. Afterwards, we concatenate the representation from parent and children to obtain updated node states $\boldsymbol{h}_i^1$, while 1 denotes that we only adopt one layer of GRN.

4.2 Progressive Prediction

After obtaining the node representation associated with message interaction, we implement node classification to identify triggers and then exploit trigger prediction to integrate nodes and make role-aware verification on cascade level.

Trigger Identification. We simply apply a Feed Forward Network (FFN) and softmax operator to classify each node.

$$\mathcal{Y}_i^t = \text{softmax}(\text{FFN}(\boldsymbol{h}_i^1)) \in \mathbb{R}^4 \tag{8}$$

The loss function of trigger identification is computed by cross-entropy criterion,

$$\mathcal{L}_t = -\frac{1}{|\mathcal{V}|}\sum_i^{|\mathcal{V}|}\sum_j^{L_t}\mathcal{Y}_i^{t,j}\log\hat{\mathcal{Y}}_i^{t,j} \tag{9}$$

where L_t is the number of trigger classes, $\hat{\mathcal{Y}}_i^{t,j}$ represents the ground-truth label of trigger.

Role-Aware Verification. Since we assume that triggers play an important role in rumor verification, we briefly design a trigger-informed and role-aware pooling mechanism that attends more to triggering posts when integrating the whole cascade. Intuitively, we calculate the weight of each post by dot product to weaken the impact of *null* messages.

$$a_i = \begin{bmatrix}0 & 1 & 1 & 1\end{bmatrix}\mathcal{Y}_i^t \tag{10}$$

Then we apply softmax to normalize attention weights in the cascade and sum up representations of all the nodes considering their role-aware weights to obtain the cascade representation $\boldsymbol{c}$.

$$\boldsymbol{c} = \sum_i^{|\mathcal{V}|} a_i \boldsymbol{h}_i^1 \tag{11}$$

Similarly, we make verify prediction and compute the loss function for verification,

$$\mathcal{Y}^v = \text{softmax}(\text{FFN}(\boldsymbol{c})) \in \mathbb{R}^3 \tag{12}$$

$$\mathcal{L}_v = -\sum_j^{L_v}\mathcal{Y}^{v,j}\log\hat{\mathcal{Y}}^{v,j} \tag{13}$$

where L_v is the number of verification classes, $\hat{\mathcal{Y}}^{v,j}$ represents the actual label.

Jointly Learning. We add two loss terms to obtain a joint loss function $\mathcal{L}$ for optimization.

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_v \tag{14}$$

Moreover, we adopt a warm-up strategy that only reserves $\mathcal{L}_t$ in the first few rounds of training and then employs the overall loss $\mathcal{L}$ to test the validity of progressive learning.

5 Experiments

5.1 Experimental Setup

Data Split. Based on our dataset, we adopt 2 types of cross validation to compare performance and generalizaiton ability of different models. (1) *Random*: to split the dataset into train, validation and test set with a proportion of 8:1:1 randomly. (2) *LOEO*: to implement leave-one-event-out cross validation (Kochkina et al., 2018) which means to treat data equally drawn from a target event as test and validation set and leave others as train set. Although model performance is usually unsatisfactory to implement LOEO validation since semantics differs a lot between events, it is more representative of real world when unprecedented event emerges.

Model Comparison. We compare various models by replacing node updating module with following methods:

CNN: A CNN-based model (Yu et al., 2017) to extract informative local comment.

RNN: A RNN-based model (Ma et al., 2016) that treats rumor cascade as time series to capture dynamic signals.

TreeLSTM: A treeLSTM-based network (Kumar and Carley, 2019) to encode propatation tree.

TreeTrans: A model (Ma and Gao, 2020) using transformer to recursively update tree nodes.

GCN: A GCN-based model (Wei et al., 2019) first treating propagation trees as graphs.

GraphSage: A graph-based model (Li et al., 2020) that randomly samples neighbors to aggregate contextual information.

UGRN: Our proposed model.

Implementation Details. The network is trained with AdamW optimizer (Loshchilov and Hutter, 2017). Hyperparameters performing best in validation set are recorded for testing. The batch size (number of cascades) is set as 5. The hidden unit size for GRN is set as 300. We adopt initial learning rate of 8e-5, 2e-5 respectively for trigger classifier layers and others. The maximum number of training epochs is 100. We have made our extended dataset[2] and code[3] publicly available.

5.2 Overall Performance

We implement the task of trigger identification and rumor verification to evaluate the performance of our proposed model, as shown in Table 2. Since these two tasks are both evil-balanced, we choose macro F1-score to compare model performance.

Method	**Trigger**		**Verify**	
	Random	LOEO	Random	LOEO
CNN	0.524	0.501	0.741	0.308
RNN	0.562	0.560	0.785	0.314
TreeLSTM	0.538	0.514	0.710	0.317
TreeTrans	0.541	0.511	0.714	0.314
GCN	0.548	0.542	0.772	0.322
GraphSage	0.549	0.561	0.781	0.304
UGRN	**0.574**	**0.570**	**0.819**	**0.346**

Table 2: Results of trigger identification and rumor verification. All the numerical values represent macro F1-score when adopting random or LOEO cross validation. The result of LOEO validation is the average of 5 folds. **Bold**: the best performance in each column.

It can be seen that our model can identify triggers more accurately and achieves the highest macro F1-score for verification. Considering the task of **trigger identification**, **CNN** model provides the baseline for trigger identification using pretrained sentence representation for classification (since features are not updated on message level and pooling for nodes is not applied). Models with reccurent unit (**UGRN**, **RNN** and **TreeLSTM**) is more competitive for trigger identification. Compared with the other two graph-based models (**GCN** and **GraphSage**), our unsymmetric and recurrent framework can model conversational structures and learn high-quality representation of triggering posts while preserving the effective operation of graph convolution. As for the task of **rumor verification**, models all present a drastic decline faced with *LOEO* test. The direct reason probably lies in the extreme imbalance of verify labels between different events and the absence of semantic sharing.

Besides, by comparing the performance between different settings of cross validation, we find that trigger identification is more robust when coming cross underrepresented semantics as the decrease in LOEO setting is not as significant as the task of verification. The direct reason probably lies in the extreme imbalance of verify labels between different events, but we do expect trigger identification to have higher transfer ability which can facilitate handling unprecedented rumor cascades.

[2] http://fudan-disc.com/data/PHEME_trigger.zip

[3] https://github.com/lchen96/trigger_identification

5.3 Ablation Study

To examine the effectiveness of key components of our **UGRN** framework, we perform ablation study by degrading the birectional graph-based node representation, as shown in Table 3. From bottom to top, we first substitute the concatenated node representation by only using parent aggregation (**UGRN-p**) or child aggregation (**UGRN-c**). The performance drops a lot when only considering information flow in one direction. Then we leave out the process of structure decomposition and directly use the symmetric adjacency matrix to apply recurrent graph convolution. The simplified model (**GRN**) can hardly distinguish triggers indicating it is valid to model information flow from different directions.

Component	Trigger		Verify	
	Random	LOEO	Random	LOEO
GRN	0.531	0.514	0.754	0.324
UGRN-c	0.541	0.522	0.768	0.321
UGRN-p	0.552	0.541	0.778	0.334
UGRN	**0.574**	**0.570**	**0.819**	**0.346**

Table 3: Ablation study on key components of UGRN. Presentation of result is the same with Table 2.

5.4 Trigger Role for Verification

In this paper, we propose three types of mechanisms to exploit trigger information for enhanced rumor verification, including parameter sharing, trigger-aware cascade pooling and warm-up of trigger identification. We examine the effect of these mechanisms to investigate the role of triggers.

Multi-Task Learning. We run our model on the two tasks separately to demonstrate the validity of multi-task learning. Table 4 shows the comparison between single-task and multi-task settings. As can be seen, the performance gain of multi-task learning is significant especially for the task of rumor verification which demonstrates the strong correlation between these two tasks and the rationality of capturing triggers.

Task	Trigger		Verify	
	Random	LOEO	Random	LOEO
Trigger	0.568	0.558	-	-
Verify	-	-	0.795	0.286
Multi-Task	**0.574**	**0.570**	**0.819**	**0.346**

Table 4: The effect of multi-task learning framework. Presentation of result is the same with Table 2.

Role-Aware Integration. Our model is designed to pay more attention to messages with triggering effect when implementing graph pooling. We replace the role-aware integration with general attention pooling to explore whether triggers can help verify rumors. Figure 5 shows the difference when converting the pooling strategy in different cascade modeling methods. Although the performance contrast is not as obvious as multi-task learning, among these 12 groups of experiments, 9 instances demonstrate role-aware integration is better than plain attention pooling which also covers that locating triggers and take full advantage of their semantics is favorable for rumor verification.

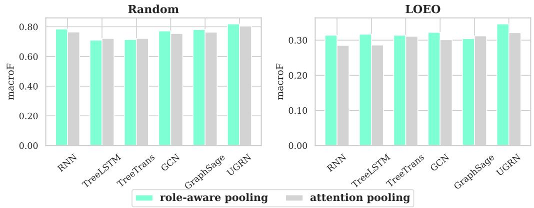


Figure 5: The effect of role-aware integration.

Trigger Warmup. During training, we adopt a warm-up strategy that only trains the network for trigger identification in the first few epochs. Consequently, we set various number of warm-up epochs to see the effectiveness of progressive learning. Figure 6 shows the impact of trigger warm-up in different validation settings. For random validation, 1 rounds of warm-up can slightly improve the verification performance but then the prediction precision drops a lot as warm-up epochs increase. However, the averaged performance of LOEO validation is steadily increasing with increment of warm-up epochs. Except when treating *Charlie Hebdo* as test event, other folds perform better with larger warm-up epochs. This is partly because the network tends to learn refined node representation under the supervision signal of triggers and provides better initialization for verification.

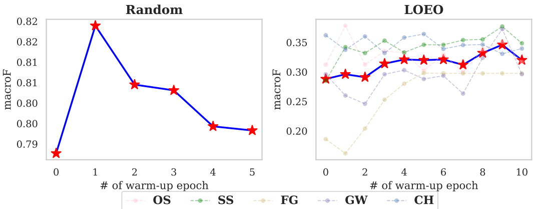


Figure 6: The effect of trigger warm-up strategy. Blue solid lines represent the averaged result and the dashed lines stand for results with different test event.

6 Conclusion and Future Work

In this paper, we propose the task of trigger identification to progressively resolve rumors. We extend PHEME dataset with annotations on message level. We design the framework of Unsymmetric Graph Reccurent Networks which significantly improves performance of two tasks. In the future, we would like to further model the relationship between triggers and rumor cascades.

Acknowledgement

This work is partially supported by Natural Science Foundation of China (No.6217020551, 71991471), National Social Science Foudation of China (No. 20ZDA060), Science and Technology Commission of Shanghai Municipality Grant (No.20dz1200600, 21QA1400600, 21511101000) and Zhejiang Lab (No. 2019KD0AD01).

References

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark.

J Scott Brennen, Felix Simon, Philip N Howard, and Rasmus Kleis Nielsen. 2020. Types, sources, and claims of covid-19 misinformation. *Reuters Institute*, 7(3):1.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM.

Lei Chen, Zhongyu Wei, Jing Li, Baohua Zhou, Qi Zhang, and Xuanjing Huang. 2020a. Modeling evolution of message interaction for rumor resolution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6377–6387, Barcelona, Spain (Online).

Xiao Chen, Changlong Sun, Jingjing Wang, Shoushan Li, Luo Si, Min Zhang, and Guodong Zhou. 2020b. Aspect sentiment classification with document-level sentiment preference modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3677, Online.

Daejin Choi, Selin Chun, Hyunchul Oh, Jinyoung Han, et al. 2020. Rumor propagation is amplified by echo chambers in social media. *Scientific reports*, 10(1):1–10.

Nicholas DiFonzo and Prashant Bordia. 2007. *Rumor psychology: Social and organizational approaches.* American Psychological Association.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 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*, pages 1024–1034.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Thomas N. Kipf and Max Welling. 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*.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*.

Sumeet Kumar and Kathleen M Carley. 2019. Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5047–5058.

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th international conference on data mining*, pages 1103–1108. IEEE.

Jiawen Li, Yudianto Sujana, and Hung-Yu Kao. 2020. Exploiting microblog conversation structures to detect rumors. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5420–5429, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. 2016. Gated graph sequence neural networks. In *Proceedings of ICLR'16*.

Hongzhan Lin, Jing Ma, Mingfei Cheng, Zhiwei Yang, Liangliang Chen, and Guang Chen. 2021. Rumor detection on twitter with claim-guided hierarchical graph attention networks. *arXiv preprint arXiv:2110.04522*.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1867–1870. ACM.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yi-Ju Lu and Cheng-Te Li. 2020. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. Association for Computational Linguistics.

Jing Ma and Wei Gao. 2020. Debunking rumors on Twitter with tree transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5455–5466, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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*, pages 2561–2571, Florence, Italy.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *International Joint Conferences on Artificial Intelligence*, pages 3818–3824.

Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1751–1754.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018a. Detect rumor and stance jointly by neural multi-task learning. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 585–593. International World Wide Web Conferences Steering Committee.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018b. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

Ahad N. Zehmakan and Serge Galam. 2020. Rumor spreading: A trigger for proliferation or fading away. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(7):073122.

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

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online.

Mengzhu Sun, Xi Zhang, Jiaqi Zheng, and Guixiang Ma. 2022. Ddgcn: Dual dynamic graph convolutional networks for rumor detection on social media.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Penghui Wei, Nan Xu, and Wenji Mao. 2019. Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4787–4798, Hong Kong, China.

Lianwei Wu and Yuan Rao. 2020. Adaptive interaction fusion networks for fake news detection. *Frontiers in Artificial Intelligence and Applications*, 325:2220–2227.

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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4644–4653, Hong Kong, China.

Lianwei Wu, Yuan Rao, Yongqiang Zhao, Hao Liang, and Ambreen Nazir. 2020a. Dtca: Decision tree-based co-attention networks for explainable claim verification. Association for Computational Linguistics.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020b. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. 2017. A convolutional approach for misinformation identification.

Jianfei Yu, Jing Jiang, Ling Min Serena Khoo, Hai Leong Chieu, and Rui Xia. 2020. Coupled hierarchical transformer for stance-aware rumor verification in social media conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1392–1401, Online.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. 2018. Discourse-aware rumour stance classification in social media using sequential classifiers. *Information Processing & Management*, 54(2):273–290.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2016. Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv preprint arXiv:1610.07363*.