LLM-Driven Evidence Retrieval and Graph Learning for Explainable Rumor Detection

Jiafu Yang, Junyan Chen, Wei Zhang(✉), and Yong Liu(✉)

Heilongjiang University, Harbin, China
{2231984, 2242039}@s.hlju.edu.cn, {zhangwei_jsj, liuyong123456}@hlju.edu.cn

Abstract. With the proliferation of false information on social networks, rumor detection has become a critical and urgent research area. Nonetheless, current rumor detection models mainly rely on textual features, which makes it difficult to fully understand the context and background information. The integration of large language models promises to enhance semantic understanding and provide more comprehensive background information. Unfortunately, existing large language models often fail to provide real background information, thereby limiting their practicality in rumor detection. Moreover, most rumor detection methods focus solely on providing detection labels, neglecting the need for explanations. To address these challenges, we propose a new evidence retrieval rumor detection model called RGED. This model first utilizes a large language model to identify relevant statement entities, then retrieves relevant evidences and summarizes the relationships between a given statement and its evidences. Subsequently, RGED constructs an evidence graph to represent the statements, evidence, and their interconnections. Finally, RGED employs Graph Neural Network, attention mechanisms, and Pre-trained Language Model to detect the veracity of rumors and generates interpretable evidence.

Keywords: Rumor Detection · Large Language Models · Graph Neural Network · Interpretable Evidence.

1 Introduction

Social media platforms are primary channels for rapid rumor dissemination, leading to significant social, political, and economic disruptions [7]. Therefore, accurate and transparent rumor detection is crucial. Traditional methods rely heavily on textual features, analyzing keywords, sentence structure, and grammar. However, these models often struggle with ambiguous language or context-dependent information, resulting in suboptimal performance. Some studies incorporate propagation structures or user information, yet obtaining such data

This work was supported by the National Natural Science Foundation of China (No. 6247074060), the Natural Science Foundation of Hei- longjiang Province in China (No. PL2024F029), the Fundamental Research Funds for Province-owned Higher Education Institutions in Heilongjiang Province (2021- KYYWF-0043, 2023-KYYWF-1463, 2024-KYYWF-0115) and Doctoral Postdoctoral Funding Project in Heilongjiang Province.

in real-world scenarios is challenging, prompting the adoption of deep learning techniques. Graph neural network (GNN)-based methods excel at capturing local features [11], but often fail to model global semantic relationships and complex structured information. With the emergence of Large Language Models (LLMs), which perform on par with or even surpass human capabilities in various Natural Language Processing (NLP) tasks [5], there is renewed potential for rumor detection. Although LLMs currently lack the ability to directly assess veracity, they can accurately identify relevant keywords and grasp true meanings. Furthermore, while most rumor detection tasks classify statements as rumors or non-rumors, we argue that providing evidence to refute rumors is more crucial.

To address these limitations, we propose a novel framework called RGED, which leverages GPT as a powerful tool for evidence retrieval rather than directly for rumor classification. The retrieved evidence is constructed into a learnable evidence graph, where nodes represent evidence and statements, and edges represent the relationships between them. A key innovation of our model is the use of learnable edges, allowing the model to dynamically adjust the importance of different edges, thus enhancing detection accuracy and interpretability. Additionally, our model provides more than just binary classification; it offers interpretable evidence by explaining how the evidence supports classification decisions.

In summary, our contributions are as follows:

We retrieve related evidences for each statement by using GPT, which can more thoroughly understand the semantics of statement when data is scarce, and minimize the introduction of noise in evidence retrieval.

We propose to build an evidence graph containing statements and their related evidences, and add weights to the edges in the graph to achieve joint learning of nodes and edges.

We improve the generalization ability of our model through data enhancement methods such as synonym replacement, random deletion and random exchange.

We introduce a multi-head self-attention mechanism to generate explanatory vectors, enhance the interpretability of the model, and enable each prediction to provide corresponding evidence.

2 Related Work

In this section, we review the current state of research in rumor detection.

2.1 Rumor Detection Model

Early rumor detection relied on manually designed semantic features and syntactic rules, which are complex and lack scalability.

As deep learning technology has advanced, neural network-based methods have become research hotspots. CNNs, RNNs, and LSTMs [6,7] have been widely used to capture long-term sequential semantic information.

More recently, dynamic graph convolutional networks have shown significant advantages in tracking message propagation and the evolution of background content, while a dual attention mechanism combined with graph convolutional networks has improved detection accuracy and robustness.

The emergence of pre-trained language models, such as BERT [5], has further propelled rumor detection technology. However, prevalent methods rely on large amounts of annotated data and insufficiently model semantic details, with limited interpretability and transparency, leading to suboptimal performance.

2.2 Large Language Model

At present, large-scale pre-trained language models demonstrate excellent transfer learning capabilities. Early models like word2vec and GloVe captured word semantics. BERT leveraged a bidirectional encoder and masked language model for rich contextual information, while RoBERTa optimized BERT's pre-training scheme [5]. Scaling up model sizes, the GPT series—especially GPT-3 with hundreds of billions of parameters—markedly improved text generation and performance across tasks [2].

Despite these advances, migrating pre-trained models to downstream tasks with limited samples remains challenging. Our research leverages LLM strengths and integrates key entity evidence to enhance understanding of complex, ambiguous rumor texts. Through multi-task collaborative training, our method improves evidence retrieval and interpretation, increasing the model's sensitivity to semantic details and optimizing rumor detection performance.

3 Methodology

In this section, we describe our proposed RGED framework. The overview of RGED is shown in Fig. 1.

3.1 Evidence Collection

At the initial stage, given a statement c, we use the GPT API with the following prompt: "Based on Named Entity Recognition results, help me identify the entities in this text: { c }." to extract relevant entities. These identified entities are then utilized to search for related evidence via the Wikipedia and GPT APIs. The sets of related evidence are represented as $S_W = \{s_{W_1}, s_{W_2}, \ldots, s_{W_m}\}$ for Wikipedia and $S_G = \{s_{G_1}, s_{G_2}, \ldots, s_{G_m}\}$ for GPT, respectively.

Next, we compute the similarity between the statement c and each evidence s_i in the set S using the cosine similarity metric, defined by the following formula:

$$\text{sim}(c, s_i) = \frac{c \cdot s_i}{\|c\|\|s_i\|} \tag{1}$$

`https://github.com/Yang1aa/RGED`

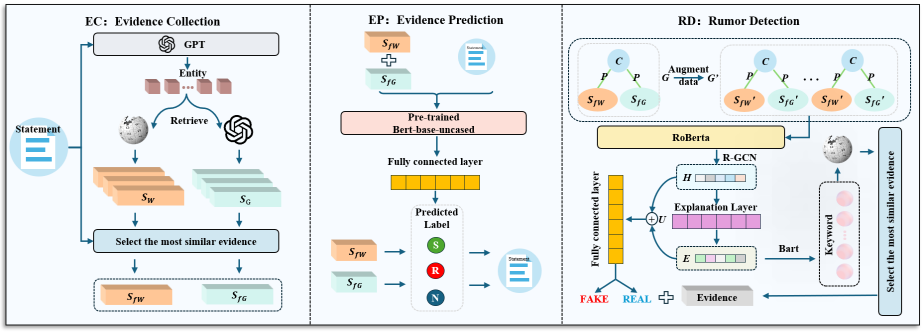

Fig. 1. The structure of RGED.

where $\text{sim}(c, s_i)$ denotes the cosine similarity between the statement c and the evidence s_i, with $\|c\|$ and $\|s_i\|$ representing the Euclidean norms of c and s_i, respectively.

After computing the similarities, all paragraphs are ranked in descending order based on their cosine similarity with the statement c. This process results in two sorted sets, one denoted as $S_{W_{\text{top}}} = \{s_{Wi1}, s_{Wi2}, \ldots, s_{Win}\}$ and $S_{G_{\text{top}}} = \{s_{Gi1}, s_{Gi2}, \ldots, s_{Gin}\}$. We select only the evidence with the highest similarity, denoted as S_{fW} and S_{fG}, for evidence predict.

3.2 Evidence Prediction

We employ a BERT model pre-trained on the FeverDataSet. The model's input comprises the following contents: c, S_{fW}, and S_{fG}. After concatenation, they are fed into a fine-tuned BERT model. The model outputs z , representing the unnormalized scores for each category, calculated as:

$$z = \text{BERT}(c, S_{fW}, S_{fG}) \tag{2}$$

The z are converted into a probability distribution p using the $softmax$ function:

$$p_i = \frac{\exp(z_i)}{\sum_{j=0}^{R-1} \exp(z_j)}, \quad i \in \{0, 1, \ldots, R-1\} \tag{3}$$

where R denotes the number of categories (*SUPPORTED*, *REFUTED*, *NOTENOUGHINFO*).

The model predicts the evidence classification label P by selecting the category index with the highest probability:

$$P = \arg\max_i p_i \tag{4}$$

To train the classification model, we utilize the Cross-Entropy Loss function. We then update the model parameters using the AdamW optimizer, which incorporates weight decay regularization to enhance generalization performance.

3.3 Rumor Detection

We construct an evidence graph where the statement c serves as the root node, and the previously obtained S_{fW} and S_{fG} serve as leaf nodes. These nodes are connected via edges, where each edge is a bidirectional connection labeled with the evidence classification P. This forms the graph $G = (V, E)$.

In order to enhance the model's ability to resist noise, we implement three primary data augmentation techniques: synonym replacement, random deletion, and random swapping. **1) Synonym Replacement** substitutes selected words with their synonyms to generate semantically similar sentences, enhancing the model's ability to learn diverse lexical expressions. **2) Random Deletion** removes certain words to create a simplified version, simulating scenarios with incomplete semantic information and compelling the model to classify effectively with missing data. **3) Random Swapping** exchanges the positions of two words to produce a sentence with altered word order but identical meaning, increasing the model's tolerance to variations in word sequence and ensuring focus on underlying semantics.

By incorporating these strategies, we enhance the model's generalization on diverse, noisy real-world data, ultimately transforming the graph G into G'.

Next, we utilize RoBERTa to transform the nodes of graph G' into vector representations. The transformation is defined as:

$$H = \text{RoBERTa}(G') \tag{5}$$

where H represents the matrix of vector embeddings for all nodes in G'.

Subsequently, we employ R-GCN to perform feature extraction on both edges and nodes within graph G'. The R-GCN layer updates the node features as follows:

$$\mathbf{H}^{(l+1)} = \sigma\left(\sum_{r\in R}\sum_{k\in\mathcal{N}_r(v)}\frac{1}{|\mathcal{N}_r(v)|}\mathbf{W}_r\mathbf{H}_k^{(l)} + \mathbf{b}\right) \tag{6}$$

where $\mathbf{H}^{(l)}$ is the node feature matrix at the l-th layer, R is the set of relation types (e.g., different types of predicted evidence labels), $\mathcal{N}_r(v)$ denotes the neighbors of node v connected via relation r, $\mathbf{W}_r$ is the weight matrix corresponding to relation r, σ is the activation function, $\mathbf{b}$ is the bias term.

To enhance the interpretability of the model, we incorporate a multi-head attention mechanism alongside an explanation layer to generate explanatory vector E.

The multi-head attention output is used to generate the explanatory vector E. In the BART model, E is fed into the decoder to produce keywords verifying the statement c. These keywords query Wikipedia APIs, and we compute cosine similarity with c, selecting the evidence with the highest similarity.

Meanwhile, we concatenate the vector H generated by the R-GCN with the vector E obtained through the attention mechanism to form the combined vector U. This concatenated vector U is then passed through a fully connected layer

for authenticity classification. The loss function for authenticity classification is defined as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}) \tag{7}$$

where $\mathcal{L}$ represents the overall loss, N denotes the number of samples, C refers to the number of classes, y_{ij} is the ground truth label of sample i for class j, and $\hat{y}_{ij}$ is the predicted probability of sample i belonging to class j.

4 Experiments

4.1 Datasets

We evaluate on three public datasets: Twitter15, Twitter16, and PHEME. In the Twitter15 and Twitter16 datasets, each statement is tagged with one of four labels—non-rumor (N), false rumor (F), true rumor (T), or unverified rumor (U). In contrast, PHEME [13] comprises five breaking news events, each labeled as either True or False. For a more detailed understanding of each dataset's characteristics and statistical information, please refer to Table 1.

Table 1. Statistics of the datasets.

Statistic	Twitter15	Twitter16	PHEME
total statements	1490	818	5802
true rumors	374	205	–
false rumors	370	205	1564
non-rumors	372	205	1600
unverified rumors	374	203	–
Avg. Words per statement	15.5	15.2	15.9
Max. Words per statement	29	28	31
Min. Words per statement	1	3	3

4.2 Experiment Settings

Hyperparameters were tuned via grid search with AdamW (max sequence length: 256, learning rates: 1×10^{-5} to 5×10^{-5}, L2 weight decay: 1×10^{-5}, warm-up: 0.1), using early stopping (patience: 12 epochs), a batch size of 16, for 40 epochs.

4.3 Baseline Method

Methods Based on Textual Features: Textual feature-based methods (SVM-TS [9], CNN [12], GRU [8], TRNN [9]) enhance text modeling via time series features, convolutional layers, recurrent units, and tree structures.

`https://fever.ai/dataset/fever.html`

Methods Based on Propagation Structures: Propagation structure-based methods (DTC [1], RDEA [3]) leverage decision trees and event structures to capture rumor dissemination.

Methods Based on Graph Neural Networks: Graph neural network-based methods (TextGCN [11], GACL [10]) leverage graph convolution or adversarial contrastive learning.

Methods Based on Large Language Models: Large language model-based methods (BERT [5], ARG [4]) employ Transformer-based representations or generative capabilities.

4.4 Performance Comparison

The experimental results are shown in Table 2 and Table 3. Our model achieves a significant improvement over all baseline models. Compared with the best-performing baseline methods, RGED consistently achieves accuracy improvements of 2.7%, 2.4%, and 5.1%, respectively. We attribute this superior performance to the following factors: **1) Diverse Evidence Retrieval:** By leveraging both Wikipedia and GPT APIs, RGED reduces reliance on a single data source and gathers a broader spectrum of evidence. **2) Enhanced Structural and**

Table 2. Performance comparison of RGED against the baselines on Twitter15 and Twitter16.

Methods	Twitter15					Twitter16				
	Acc	N	F	T	U	Acc	N	F	T	U
		F1	F1	F1	F1		F1	F1	F1	F1
SVM-TS	0.534	0.789	0.476	0.398	0.481	0.574	0.569	0.565	0.571	0.566
CNN	0.583	0.569	0.574	0.610	0.576	0.613	0.598	0.586	0.622	0.594
GRU	0.634	0.764	0.578	0.442	0.479	0.762	0.718	0.735	0.694	0.721
TRNN	0.715	0.685	0.620	0.813	0.646	0.742	0.729	0.733	0.744	0.737
DTC	0.443	0.424	0.362	0.721	0.325	0.471	0.466	0.483	0.469	0.470
RDEA	0.855	0.831	0.857	0.901	0.816	0.880	0.823	0.878	0.937	0.875
TextGCN	0.703	0.679	0.718	0.746	0.657	0.715	0.687	0.751	0.802	0.654
GACL	0.882	0.853	0.890	0.902	0.848	0.902	**0.912**	0.847	0.902	0.876
BERT	0.735	0.731	0.722	0.730	0.804	0.804	0.777	0.525	0.824	0.787
ARG	0.798	0.750	0.805	0.880	0.756	0.766	0.652	0.771	0.905	0.727
RGED	**0.909**	**0.883**	**0.922**	**0.922**	**0.912**	**0.926**	0.876	**0.925**	**0.975**	**0.935**

Contextual Representation: The integration of R-GCN enables the effective aggregation of information from both nodes and edges, capturing intricate structural and contextual relationships. **3) Improved Feature Focus:** A multi-head attention mechanism is employed to better emphasize critical features, enhancing the model's capacity to detect subtle interactions.

Table 3. Performance comparison of RGED against the baselines on PHEME.

Methods	Acc	Pre		Rec		F1	
		N	F	N	F	N	F
SVM-TS	0.651	0.642	0.663	0.786	0.617	0.707	0.639
CNN	0.665	0.661	0.671	0.679	0.652	0.670	0.661
GRU	0.742	0.754	0.737	0.730	0.753	0.742	0.745
TRNN	0.792	0.804	0.755	0.786	0.806	0.795	0.780
DTC	0.581	0.579	0.582	0.788	0.473	0.668	0.522
RDEA	0.798	0.822	0.791	0.776	0.806	0.798	0.798
TextGCN	0.772	0.808	0.782	0.763	0.787	0.785	0.784
GACL	0.850	0.871	0.801	**0.901**	0.750	0.886	0.775
BERT	0.700	0.571	0.642	0.810	0.774	0.670	0.702
ARG	0.882	0.893	0.869	0.880	0.884	0.887	0.876
RGED	**0.933**	**0.962**	**0.910**	0.893	**0.968**	**0.926**	**0.938**

4.5 Ablation Study

To further investigate the effectiveness of our key modules, we conduct ablation studies to isolate each component's impact on overall performance, as shown in Table 4.

Table 4. Ablation study of RGED model.

Dataset	Method	Acc	F1			
			N	F	T	U
Twitter15	-w/o $S_{finalGPT}$	0.862	0.844	0.833	0.877	0.844
	-w/o $S_{finalWIKI}$	0.849	0.803	0.821	0.904	0.861
	-w/o $R-GCN$	0.842	0.806	0.813	0.895	0.853
	-w/o $DataAugmentation$	0.872	0.865	0.834	0.914	0.877
	-w/o $InterpretableEvidence$	0.845	0.745	0.760	0.831	0.740
	RGED	**0.909**	**0.883**	**0.922**	**0.922**	**0.912**
Twitter16	-w/o $S_{finalGPT}$	0.896	0.867	0.894	0.935	0.889
	-w/o $S_{finalWIKI}$	0.896	0.835	0.914	0.929	0.902
	-w/o $R-GCN$	0.884	0.854	0.850	0.975	0.860
	-w/o $DataAugmentation$	0.908	0.860	0.884	0.962	0.933
	-w/o $InterpretableEvidence$	0.891	0.753	0.800	0.876	0.779
	RGED	**0.926**	**0.876**	**0.925**	**0.975**	**0.935**
PHEME	-w/o $S_{finalGPT}$	0.888	0.890	0.885	-	-
	-w/o $S_{finalWIKI}$	0.916	0.912	0.919	-	-
	-w/o $R-GCN$	0.904	0.901	0.908	-	-
	-w/o $DataAugmentation$	0.921	0.920	0.922	-	-
	-w/o $InterpretableEvidence$	0.863	0.885	0.854	-	-
	RGED	**0.933**	**0.926**	**0.938**	-	-

Below, we provide a detailed description of these variants:

-w/o S_{fG}: In this variant, we exclusively utilize the Wiki-based evidence S_{fW} obtained after entity extraction, omitting the GPT-generated evidence.

-w/o S_{fW}: Here, we solely employ the GPT-based evidence S_{fG} derived from entity extraction, excluding the Wiki-based evidence.

-w/o $R{-}GCN$: We remove the R-GCN module entirely, directly inputting the obtained node representations H into the subsequent connection layer without graph-based feature aggregation.

-w/o $DataAugmentation$: This variant excludes the data augmentation component, thereby evaluating the performance of the framework without augmented data.

-w/o $InterpretableEvidence$: We removed the multi-head attention mechanism along with the subsequent keyword generation module based on BART.

Our ablation studies show that relying on a single evidence source reduces performance, while removing the R-GCN limits edge information aggregation. Omitting data augmentation affects only partially missing statements, and excluding interpretability prevents the model from using multi-head attention to extract higher-level representations. These results highlight that each component is crucial, and their seamless integration is key to optimal performance.

4.6 Case study of evidence

We exploited the evidence to debunk rumors and enhance the interpretability of the model. For illustration, we randomly selected one true example and one false example from the dataset. The evidence generation results are shown in Fig. 2. The evidence highlighted in red was the most relevant.

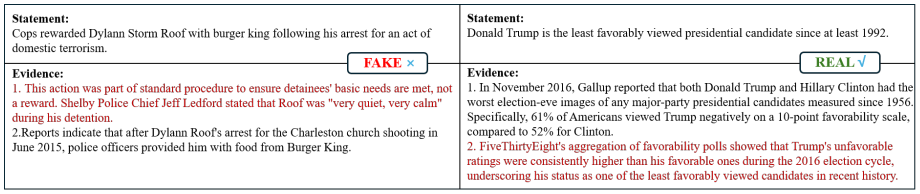

Fig. 2. The evidence generation results of real statements.

In the first case, the evidence indicates a standard procedure rather than a reward, which contradicts the rumor. In the second case, multiple pieces of evidence provide authentic and relevant information, indicating that the statement is real.

5 Conclusion

In this study, we introduce the RGED framework for interpretable rumor detection. Our approach combines LLMs, evidence retrieval, and GNNs to detect

rumor, leveraging the multi-head attention mechanism and the BART model to generate interpretable evidence. Experiments demonstrate the effectiveness of RGED in identifying rumors and providing clear and interpretable evidences, which takes an important step towards building a reliable rumor detection system.

References

1. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 675–684 (2011)
2. Floridi, L., Chiriatti, M.: Gpt-3: Its nature, scope, limits, and consequences. Minds and Machines **30**, 681–694 (2020)
3. He, Z., Li, C., Zhou, F., Yang, Y.: Rumor detection on social media with event augmentations. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 2020–2024 (2021)
4. Hu, B., Sheng, Q., Cao, J., Shi, Y., Li, Y., Wang, D., Qi, P.: Bad actor, good advisor: Exploring the role of large language models in fake news detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 22105–22113 (2024)
5. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of naacL-HLT. vol. 1, p. 2. Minneapolis, Minnesota (2019)
6. Ma, J., Dai, J., Liu, Y., Han, M., Ai, C.: Contrastive learning for rumor detection via fitting beta mixture model. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. pp. 4160–4164 (2023)
7. Ma, J., Liu, Y., Liu, M., Han, M.: Curriculum contrastive learning for fake news detection. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 4309–4313 (2022)
8. Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B.J., Wong, K.F., Cha, M.: Detecting rumors from microblogs with recurrent neural networks (2016)
9. Ma, J., Gao, W., Wei, Z., Lu, Y., Wong, K.F.: 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. pp. 1751–1754 (2015)
10. Sun, T., Qian, Z., Dong, S., Li, P., Zhu, Q.: Rumor detection on social media with graph adversarial contrastive learning. In: Proceedings of the ACM Web Conference 2022. pp. 2789–2797 (2022)
11. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 7370–7377 (2019)
12. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T., et al.: A convolutional approach for misinformation identification. In: IJCAI. pp. 3901–3907 (2017)
13. Zheng, P., Huang, Z., Dou, Y., Yan, Y.: Rumor detection via assessing the spreading propensity of users. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)