# Rumor Detection on Social Media with Crowd Intelligence and ChatGPT-Assisted Networks

**Chang Yang[1], Peng Zhang[2]*, Wenbo Qiao[3], Hui Gao[2], Jiaming Zhao[3]**

[1] College of Information Science and Engineering, Shenyang University of Technology, China
[2] College of Intelligence and Computing, Tianjin University, Tianjin, China
[3] School of New Media and Communication, Tianjin University, Tianjin, China

yangchang@smail.sut.edu.cn
{pzhang,qiaowb,hui_gao}@tju.edu.cn
zn_zjm@163.com

## Abstract

In the era of widespread dissemination through social media, the task of rumor detection plays a pivotal role in establishing a trustworthy and reliable information environment. Nonetheless, existing research on rumor detection confronts several challenges: the limited expressive power of text encoding sequences, difficulties in domain knowledge coverage and effective information extraction with knowledge graph-based methods, and insufficient mining of semantic structural information. To address these issues, we propose a Crowd Intelligence and ChatGPT-Assisted Network(CICAN) for rumor classification. Specifically, we present a crowd intelligence-based semantic feature learning module to capture textual content's sequential and hierarchical features. Then, we design a knowledge-based semantic structural mining module that leverages ChatGPT for knowledge enhancement. Finally, we construct an entity-sentence heterogeneous graph and design Entity-Aware Heterogeneous Attention to integrate diverse structural information meta-paths effectively. Experimental results demonstrate that CICAN achieves performance improvement in rumor detection tasks, validating the effectiveness and rationality of using large language models as auxiliary tools.

## 1 Introduction

In recent years, the number of rumors posted on the Internet has increased significantly, and the harm of rumors is that they may cause panic and unnecessary actions, undermine social stability and trust, and negatively affect individuals, organizations, and society (Lin et al., 2021). Unlike traditional media channels, information on social networks has the characteristics of fast dissemination and wide coverage, leading to a large amount of information that cannot be verified in real-time (Khan et al., 2021; Vosoughi et al., 2018). In response to this problem, rumor detection methods based on crowd intelligence have attracted much attention. Crowd intelligence means social media users spread and interpret information by sharing personal opinions positions, questioning, and providing evidence (Phan et al., 2023). We realize that rumors are the result of the interaction between social media and swarm intelligence, a phenomenon of information dissemination among groups. That is, the characteristics of swarm intelligence are the fundamental reason why rumor detection becomes more complicated. From the perspective of crowd intelligence, we redefine rumors based on existing viewpoints (Phan et al., 2023; Zubiaga et al., 2018; Doan et al., 2011): the Rumor is an influential but unverified statement widely disseminated on social networks and other channels through people's collective participation and information interaction. Social media users usually share their views on the veracity of rumors through collective, subjective understanding (Tolmie et al., 2018; Ma et al., 2018), which may help reveal the truth behind the rumors. Therefore, the crowd's cognitive mechanism for rumors has important guiding significance for rumor detection, especially the detection method based on crowd intelligence. However, there are still some problems in effectively utilizing crowd intelligence in rumor detection:

1) Due to the complexity and irregularity of information on social networks, current research methods have certain limitations in extracting deep semantic features. Convolutional Neural Networks (CNNs) are commonly used for this purpose. Still, their max-pooling operations can lead to loss of detail and positional information, limiting the expressive power of text-encoded sequences. Therefore, improving the model's ability to detect deep semantic features while preserving the relationship between low and high-level features is crucial for optimal performance.

* Corresponding author
This work is done when Chang Yang visited Tianjin University for exchange.



*Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5705–5717
December 6-10, 2023 

2) The method (Wei et al., 2019; Tchakounté et al., 2020) based on Crowd Intelligence has been proven effective in identifying information on social networks. Since information on social networks is noisy, it is crucial to introduce additional knowledge to supplement semantic information. However, current primary supplement semantic information methods, such as knowledge graphs, face challenges in extracting information effectively. Therefore, it is essential to introduce external knowledge effectively.

3) Fully mining the semantic structure information in the text is crucial for rumor detection, and there are currently two main problems to be solved. First, the study (Zhu et al., 2022) found that insufficient learning of entity information will cause entity bias and affect the model's generalization to future data. Second, important information is usually distributed in various text positions rather than concentrated in a specific position, which increases the difficulty of extracting complex long-distance semantically dependent features. Efficiently constructing heterogeneous graphs and selecting appropriate meta-paths to integrate semantic structure information are the keys to addressing these issues.

Based on the above challenges and shortcomings, this paper proposes Crowd Intelligence and ChatGPT-Assisted Network(CICAN) that uses crowd intelligence and introduces a large language model as an auxiliary tool, combining semantic content and structure information for rumor detection. The model includes the Crowd Intelligence-based Semantic Feature Learning Module (CIS) and the Knowledge-based Semantic Structure Mining Module (KSS). First, we propose a context-aware semantic encoder and a multi-granularity feature extractor for extracting semantic content features, leveraging TweetBERT to enhance the model's ability to handle semantic and contextual relevance, and combining BiLSTM and capsule networks for sequential and hierarchical feature learning. Then, by leveraging ChatGPT to extract refined and structured knowledge from the raw data, we construct the Entity-Sentence heterogeneous graph containing two types of nodes and design a new hierarchical attention mechanism to enhance the model's ability to capture complex relationships. Our main contributions are as follows:

- Our proposed rumor detection model (CICAN) extracts semantic content and semantic structure features by introducing crowd intelligence and adequate external knowledge. Furthermore, we redefine rumors based on crowd intelligence and suggest meaningful future research directions in the field of rumor detection.

- We apply large language models(LLM) to rumor detection for the first time, using ChatGPT as an auxiliary means for text enhancement, entity linking, and text conceptualization, and demonstrate the effectiveness and accuracy of the approach.

- Our experiments on two real-world datasets show that our method is highly effective and outperforms the baseline approach. Our unique design has substantially enhanced the task of rumor detection.

## 2 Related Work

### 2.1 Semantic Content-based Method

At present, the entry point of rumor detection work mainly focuses on the content characteristics of social posts (Phan et al., 2023) because social posts carry crowd intelligence, which can be gathered in their creation, interaction, and dissemination. Traditional methods are mainly based on feature engineering (Castillo et al., 2011; Lim et al., 2017; Rayana and Akoglu, 2015; Al-Ghadir et al., 2021), using user opinions on social media platforms and feature information of social networks to extract features and use classifiers to detect rumors manually.

However, the complexity of social media data is mainly reflected in the diversity of content, topics, and styles. In contrast, deep learning-based methods can automatically learn representations of features. Existing studies (Ma et al., 2016; Asghar et al., 2021; Liu and Wu, 2018; Yu et al., 2017) mostly use CNN to extract key features from the text content of related posts. However, the max pooling operation of CNN comes at the cost of losing the precise encoding of object locations (Mazzia et al., 2021). To address this issue, Sabour et al. (2017) introduced capsule networks, which use vector outputs and the "routing-by-agreement" algorithm to retain position information and mine features from various perspectives. Capsule networks have proven effective in text classification tasks (Zhao et al., 2018; Yang et al., 2019). By incorporating capsule networks into NLP, we can capture different types of semantic information, representing word relationships, syntactic structures, and similar word characteristics.

### 2.2 External Knowledge-based Method

Text on social networking platforms is usually limited by the number of characters, which prevents an accurate understanding of content information based on sufficient context and details. Existing research in rumor detection has primarily focused on incorporating external knowledge, such as knowledge graphs. CompareNet (Hu et al., 2021) captures content-knowledge consistency by comparing contextual entity representations with knowledge-based ones. The multimodal knowledge-aware event memory network (MKEMN) (Zhang et al., 2019) integrates knowledge graphs and memory networks, considering both text and multimedia data to detect rumors. DDGCN (Sun et al., 2022) dynamically models message spreading and background knowledge from knowledge graphs in a unified framework. However, the complexity and informality of social network information pose challenges for knowledge graph-based methods to extract knowledge information in specific research fields effectively.

Recently, large-scale language models, such as ChatGPT (OpenAI, 2022), have shown remarkable performance in natural language processing tasks, capable of extracting accurate and rich features beyond traditional methods (Feng et al., 2023). Therefore, utilizing ChatGPT as an auxiliary tool to improve the model's performance has become a promising trend.

### 2.3 Semantic Structure-based Method

Graph Neural Networks (GNNs) have gained popularity for capturing contextual relations, syntactic structures, and semantic dependencies in text. Yao et al. (2019) constructed a single text graph based on word co-occurrences and document-word relationships. Liu et al. (2018) proposed Concept Interaction Graph for text matching, employing Siamese Encoded Graph Convolutional Network to match results. Huang et al. (2020) proposed a meta-path-based heterogeneous graph attention network to fuse information from source tweets in rumor detection. Yuan et al. (2019) employed a global-local attention network to capture local semantic relations and global structural information propagated by source tweets. Therefore, the attention mechanism designed based on heterogeneous graphs can effectively integrate semantic structure information and improve the ability of rumor detection.

## 3 Problem Statement

In this paper, we formulate the rumor detection problem as a classification task. We aim to learn a function $p(Y|V, Z; \theta)$, where V represents the semantic content information, and Z represents the semantic structure information. The input consists of text information containing source tweets and comments, as well as the entity-sentence heterogeneous graph structure information constructed from external knowledge. All the learnable parameters of the model are denoted as $\theta$. The class label $Y = \{T, F, N, U\}$ represents the four categories: T (true rumor), F (false rumor), N (non-rumor), and U (unverified rumor).

The semantic content information $V$ comprises a collection of event sentences, including source tweets and comments. For each event $X = \{\mathrm{x}, \mathrm{c}_1, \mathrm{c}_2, ..., \mathrm{c}_p\}$ , where x represents the source tweet, $c_i$ represents the comment corresponding to the source tweet, and $p$ is the maximum number of comments. To capture the semantic structure information, we construct the entity-sentence heterogeneous graph denoted as $G_i^{S\_E} = (V_i^{S\_E}, E_i^{S\_E})$, where $V_i^{S\_E}$ and $E_i^{S\_E}$ represent the nodes and edges in the graph, respectively. $V_i^{S\_E}$ is composed of a series of sentences from the event $X$(source tweet, comments) along with reinforced entity nodes $E_t'$. The graph integrates the entity and the event sentences, enabling the modeling of relationships between entities and sentences.

## 4 Methodology

We detail the overall framework of the proposed Crowd Intelligence and ChatGPT-Assisted Network(CICAN), as shown in Figure 1. We elaborate on the main constituent modules of CICAN and its rumor detection process.

### 4.1 Crowd Intelligence-based Semantic Feature Learning Module

The diverse nature and the challenge of discerning truth from falsehood in the discourse on social networks make it difficult to extract complex semantic content features for rumor detection from the massive amount of information. To address this issue, we design a contextual semantic encoder and multi-granularity feature extractor. Recent experiment

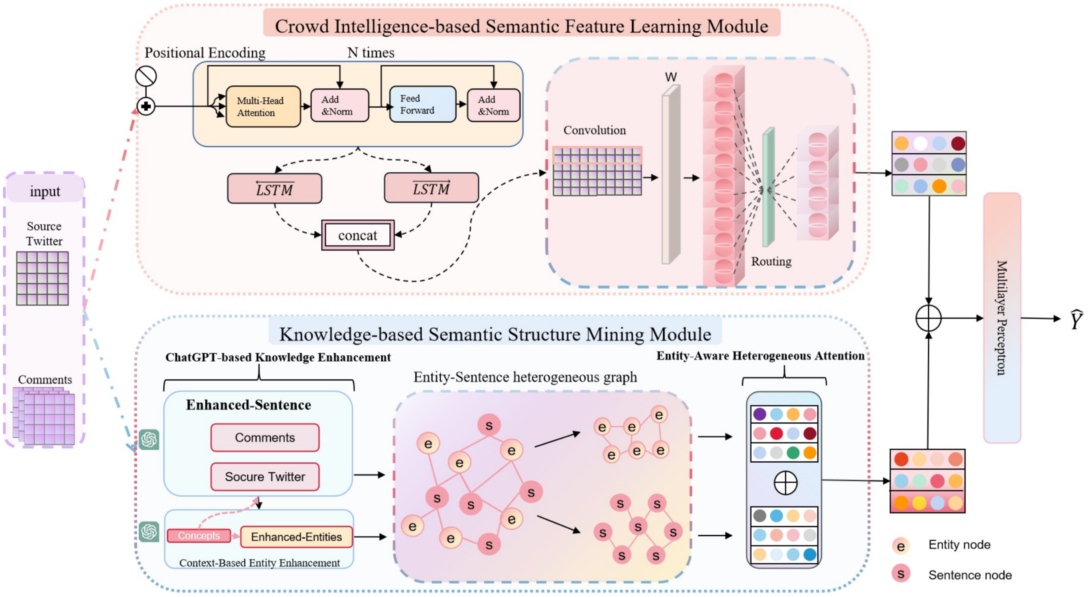



Figure 1: The architecture of Crowd Intelligence and ChatGPT-Assisted Network(CICAN)

(Farinneya et al., 2021) shows that TweetBERT performs well in identifying rumors on social media. TweetBERT (Qudar and Mago, 2020) is a pre-trained language model explicitly tailored for tweets, which is trained on a large corpus of tweet data, enabling it to capture nuances and essential features in tweets effectively. Inspired by this, we employ TweetBERT as the contextual semantic encoder to obtain the word embedding of each text sequence as $H^{\mathrm{TB}} = [H_0^{TB}, H_1^{TB}, \cdots, H_p^{TB}]$ for enhancing the model's ability to handle semantic and contextual relevance.

In addition, we also combine BiLSTM and capsule network as the multi-granularity feature extractor for text sequential and hierarchical feature learning. Specifically, BiLSTM is a recurrent neural network structure that can efficiently process sequential information. We obtain the hidden state representation $H_i^{BiLSTM} \in \mathbb{R}^{s\times 2h}$ by concatenating the output of BiLSTM,

$$H_i^{\mathrm{BiLSTM}} = [LSTM(\overrightarrow{H}_i), LSTM(\overleftarrow{H}_i)] \quad (1)$$

where $2h$ is the hidden state dimension of BiLSTM, and $s$ is the sequence length. For hierarchical feature learning, we use a 1D convolutional layer with a convolution kernel size of $k$ and the number of convolution kernels of $k\_num$ to extract n-gram features at different positions in the sentence, $u_i = Conv(H_i^{BiLSTM}) \in \mathbb{R}^{(s-k+1)\times k\_num}$, which can be regarded as the primary capsules with a quantity of $(s-k+1)$ and a dimension of $k\_num$, and use "squashing " function:

$$u_i = Squash(u_i) = \frac{||u_i||}{1\ +\ ||u_i||^2}\frac{u_i}{||u_i||} \quad (2)$$

Then, the prediction vector of the primary capsule $u_i$ is obtained by projection, $\hat{u}_i = W_i^t u_i, \hat{u}_i \in \mathbb{R}^{c_{in}\times(s-k+1)\times c_{dim}}$, where $W_i^t$ is the transformation matrix, the number of higher-level capsules is $c_{in}$, and the dimension of the capsules is $c_{dim}$. Next, we use the dynamic routing mechanism to generate prediction vectors for higher-level capsules based on the primary capsules. The higher-level capsule $v_k$ is the sum of prediction vector in primary capsule $\hat{u}_{k|i}$ and coupling coefficient $c_{k|i}$.

$$v_k = \sum_k c_{k|i} \cdot \hat{u}_{k|i} \quad (3)$$

$$c_{k|i} = softmax(b_{k|i}) \quad (4)$$

where $b_{k|i}$ is initialized to 0 and updates in each iteration $b_{k|i} = b_{k|i} + \hat{u}_{k|i} * v_k$, $c_{k|i}$ is determined by the iterative dynamic routing process. Finally, through $r$ rounds of dynamic routing iterations, we get the final output of the capsule network $V = [v_0, v_1, ..., v_p]$ , $v_k \in \mathbb{R}^{c_{in}\times c_{\mathrm{dim}}}$ , which represents the semantic content information of the event.

### 4.2 Knowledge-based Semantic Structure Mining Module

We design the Knowledge-based Semantic Structure Mining Module to capture long-distance structure information. As shown in Figure 2.

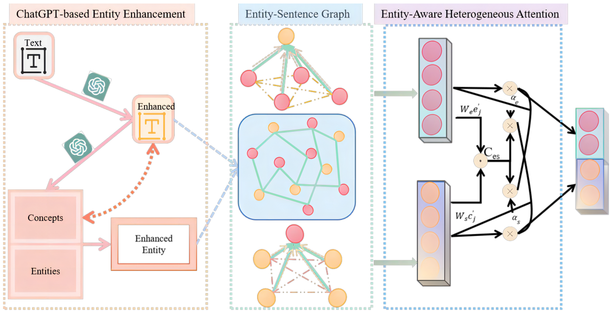



Figure 2: Knowledge-based Semantic Structure Mining Module

**ChatGPT-based Knowledge Extraction:** To deal with informal texts on social networks, including abbreviations, misspellings, and spoken language, etc., we use ChatGPT to introduce external knowledge effectively.

First, we process the raw texts using ChatGPT to build a corrected set of texts $X_i' = \{c_0', c_1', c_2', ..., c_p'\}$. We design task-specific prompts that aim to guide the understanding process of ChatGPT better. As shown in Prompt 1 of Appendix C. Then, we utilize ChatGPT to extract entities and concept information from the enhanced text via the "isA" relation, and the prompt used is shown in Prompt 2 of Appendix C. For a source tweet $c_0 \in \mathbb{R}^{n \times d}$ of length $n$, entities can be represented as $E_t \in \mathbb{R}^{m \times d}$, concepts can be represented as $C_t \in \mathbb{R}^{m \times k \times d}$, $m$ represents the maximum number of entities, and $k$ represents the maximum number of concepts corresponding to each entity. Finally, we design the Context-based Entity Enhancement method. Current research work (Zhang et al., 2019) mainly focuses on modeling the attention mechanism between concepts and entities but does not fully consider the interrelationship between concepts and sentences. In fact, each concept can be regarded as a semantic interpretation of the entity, which may have different importance in different contexts. Therefore, we calculate the similarity between the concept $C_t^j$ corresponding to the entity and the sentence representation $c_0$ to determine the importance of the concept for the context. By adding the calculation results to the original entity representation $E_t$, we obtain the enhanced entity representation.

$$E_t' = E_t + \sum_{j=1}^{k} \text{softmax}(c_0 \cdot C_t^j) \cdot C_t^j \tag{5}$$

**Entity-Aware Hierarchical Attention:** To better capture the long-distance semantic dependencies and complex structural information of sentences, we construct the entity-sentence heterogeneous graph with enhanced sentence nodes $X' = \{c_0', c_1', c_2', \ldots, c_n'\}$ and entity nodes $E_t' = \{e_0', e_1', e_2', \ldots, e_m'\}$. The graph contains two meta-paths: Entity-Sentence-Entity (ESE) considers the relationship between different entities in a sentence, and Sentence-Entity-Sentence (SES) reflects the relationship between other sentences containing the same entity. Inspired by Heterogeneous Graph Attention Network (Wang et al., 2019), the node-level attention mechanism is introduced to capture the information interaction between nodes and obtain richer feature representations.

To align the feature spaces of sentence nodes $c_i'$ and entity nodes $e_i'$, transformation matrices $W_c$ and $W_e$ are applied to project the features of these distinct node types into a unified feature space. $c_i' = W_c'^{\mathrm{T}} \cdot c_i', e_i' = W_e'^{\mathrm{T}} \cdot e_i'$. Then, the sentence attention and entity attention are respectively calculated for entity nodes with ESE meta-path structure and sentence nodes with SES meta-path structure, which can be expressed as:

$$\beta_{ji}' = \frac{\exp(\alpha(a_e^T \cdot [e_j'; e_i']))}{\sum_{k \in N_{e_j}} \exp(\alpha(a_e^T \cdot [e_j'; e_k']))} \tag{6}$$

$$\alpha_{ji}' = \frac{\exp(\alpha(a_s^T \cdot [c_j'; c_i']))}{\sum_{k \in N_{c_j}} \exp(\alpha(a_s^T \cdot [c_j'; c_k']))} \tag{7}$$

where $c_i$ and $c_j$ represent a node with a SES structure, $e_i$ and $e_j$ represents a node with a ESE structure. $a_e, a_s$ are learned parameters, $\alpha$ is the nonlinear activation function LeakyReLU (Xu et al., 2015). To capture multiple representations from different relations, a multi-head paradigm similar to multi-head attention is adopted, and the following output feature representation is obtained:

$$c_j'' = \overset{K}{\underset{k=1}{\Big\|}} \sigma \left( \sum_{i \in N(c_j')} \alpha_{ji}^k c_i' \right) \tag{8}$$

$$e_j'' = \overset{K}{\underset{k=1}{\Big\|}} \sigma \left( \sum_{i \in N(e_j')} \beta_{ji}^k e_i' \right) \tag{9}$$

where $c_j''$ represents the feature representation that $c_j'$ learned from the SES meta-path, $e_j''$ represents the feature representation that $e_j'$ learned from the

Table 1: Rumor detection results on two datasets.

| Method | Module | | | Tweet15 | | | | | Tweet16 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Content-based | Struction-based | Knowledge-based | ACC | F1 N | F1 F | F1 T | F1 U | ACC | F1 N | F1 F | F1 T | F1 U |
| PPC | ✓ | - | - | 0.668 | 0.579 | 0.604 | 0.591 | 0.610 | 0.626 | 0.630 | 0.617 | 0.636 | 0.628 |
| dEFEND | ✓ | - | - | 0.731 | 0.631 | 0.646 | 0.617 | 0.668 | 0.721 | 0.649 | 0.603 | 0.611 | 0.637 |
| TD-RvNN | ✓ | ✓ | - | 0.713 | 0.702 | 0.691 | 0.746 | 0.654 | 0.737 | 0.662 | 0.743 | 0.801 | 0.768 |
| BiGCN | ✓ | ✓ | - | 0.798 | 0.716 | 0.758 | 0.843 | 0.876 | 0.803 | 0.792 | 0.788 | 0.796 | 0.814 |
| ClaHi-GAT | ✓ | ✓ | - | 0.823 | 0.817 | 0.802 | 0.831 | 0.813 | 0.802 | 0.817 | **0.833** | 0.808 | 0.824 |
| KMGCN-NoVisual | ✓ | - | ✓ | 0.766 | 0.701 | 0.724 | 0.799 | 0.717 | 0.739 | 0.727 | 0.764 | **0.811** | 0.790 |
| DDGCN | ✓ | - | ✓ | 0.828 | 0.795 | **0.818** | 0.792 | 0.780 | 0.812 | 0.723 | 0.712 | 0.779 | 0.799 |
| **CICA-Net** | ✓ | ✓ | ✓ | **0.855** | **0.816** | 0.811 | **0.870** | **0.901** | **0.840** | **0.838** | 0.826 | 0.802 | **0.862** |

ESE meta-path, $k$ represents the number of attention heads.

Different from the Heterogeneous Graph Attention Network, We use co-attention (Lu et al., 2016) to automatically learn the independence of meta-paths and the synergy between different meta-paths, enabling more accurate heterogeneous fusion. First calculate the affinity matrix of $e''_j$ and $c''_j$: $C_{es} = \tanh(e''^T_j W_{es} c''_j)$. where $W_{es}$ is a learnable parameter. Then, $C_{es}$ is sent to the network as a feature, and the attention maps of entity features and sentence features are learned as follows:

$$H_s = \tanh(W_s c''_j + (W_e e''_j) C^T_{es}) \quad (10)$$

$$\alpha_s = soft\max(W^T_S H_s) \quad (11)$$

$$H_e = \tanh(W_e e''_j + (W_s c''_j) C_{es}) \quad (12)$$

$$\alpha_e = soft\max(W^T_E H_e) \quad (13)$$

where $W_s, W_e, W_S, W_E$ represent the learnable parameters, $\alpha_s$ and $\alpha_e$ represent the attention weights of sentence features $c''_j$ and entity features $e''_j$, as the importance of each meta-path. With the learned weights as coefficients, we can fuse meta-paths $z_j = \alpha_s c''_j + \alpha_e e''_j$. Finally, we use cascading to integrate semantic content features $V = \{v_0, v_1, \ldots, v_p\}$ and semantic structure features $Z = \{z_0, z_1, \ldots, z_p\}$ into a unified representation, and output accuracy prediction through a multilayer perceptron:

$$\hat{y} = \mathrm{softmax}(W[V; Z] + b) \quad (14)$$

where $W$ is a learnable parameter and $b$ represents a bias.

## 5 Experiments

### 5.1 Datasets and Experimental Settings

We evaluate the proposed model using two publicly datasets recognized in the field of rumor detection, including Twitter15 (Ma et al., 2017) and Twitter16 (Ma et al., 2017). Detailed dataset statistics are listed in Appendix A.

To maintain consistency, we set the batch size to 16 and the number of epochs to 30. The initial learning rate is set to 0.001, gradually decreasing during training, and the weight decay coefficient is 0.0001. We employ L2 regularization with a weight of 0.001 to control the model's complexity. Furthermore, the dataset is divided into a training set (70%), a validation set (20%), and a test set (10%). To ensure the robustness of the results, we perform 5-fold cross-validation to obtain reliable experimental outcomes. The detailed configuration of our model parameters is provided in Appendix B.

### 5.2 Baselines

We have conducted a comparative analysis of state-of-the-art (SOTA) models in the field of rumor detection, including models based on semantic content information, semantic structural information, and prior knowledge. Specifically, the following models were selected: 1) **PPC** (Liu and Wu, 2018): This approach combines Convolutional Neural Networks and Recurrent Neural Networks to construct a time series classifier for rumor detection. 2) **dEFEND** (Shu et al., 2019): This model explores the correlation between the source tweet and its corresponding comments using an attention mechanism to investigate semantic relationships. 3) **RvNN** (Ma et al., 2018): This model captures the traversal direction of the information propagation tree to obtain clues about the spread and the semantic content. 4) **BiGCN** (Bian et al., 2020): This model is a rumor detection approach based on Graph Convolutional Networks to capture the structural features

Table 2: Ablation experiment results of our proposed CICAN model. (CIS represents Crowd Intelligence-based Semantic Feature Learning Module, KSS represents Knowledge-based Semantic Structure Mining Module.)

| MODEL | CIS | | | KSS | | Dataset | |
|---|---|---|---|---|---|---|---|
| | TweetBert | Cap | BiLSTM | EE | ES graph | Twitter15 ACC | Twitter16 ACC |
| w/o CIS | - | - | - | ✓ | ✓ | 0.817 | 0.813 |
| w/o CIS-TweetBERT | - | ✓ | ✓ | ✓ | ✓ | 0.828 | 0.821 |
| w/o CIS-cap | ✓ | - | ✓ | ✓ | ✓ | 0.84 | 0.829 |
| w/o CIS-BiLSTM | ✓ | ✓ | - | ✓ | ✓ | 0.843 | 0.832 |
| w/o KSS | ✓ | ✓ | ✓ | - | - | 0.823 | 0.817 |
| w/o KSS-ES graph | ✓ | ✓ | ✓ | ✓ | - | 0.836 | 0.829 |
| w/o KSS-EE | ✓ | ✓ | ✓ | - | ✓ | 0.827 | 0.82 |
| CICAN | ✓ | ✓ | ✓ | ✓ | ✓ | 0.855 | 0.84 |

of rumor propagation. 5) **ClaHi-GAT** (Lin et al., 2021): This model builds an undirected interaction graph and uses a hierarchical graph attention network to capture the semantic features of posts for rumor detection. 6) **KMGCN-NoVisual** (Wang et al., 2020): KMGCN models semantic representations by combining text, knowledge, and visual information. We use the KMGCN-NoVisual model (which excludes visual information) for a fair comparison. 7) **DDGCN** (Sun et al., 2022): This model uses Graph Convolutional Networks to model two types of structural information, including the content information in the propagation and the conceptual information of the knowledge graph.

### 5.3 Performance Comparison

As shown in Table 1, CICAN outperforms other methods with accuracy rates of 85.5% and 84.0% on Twitter15 and Twitter16 datasets, respectively. The semantic content-based models PPC and dEFEND neglect deep semantic features and structural information. At the same time, we propose a method combining BiLSTM and capsule network, which can extract higher-level semantic information and capture part-whole relationships. Regarding models utilizing semantic structure information, the tree structure model RvNN exhibits slightly inferior performance due to its limited ability to capture long-distance dependencies in sequences. Conversely, graph-based models demonstrate better capability in capturing semantic structure features. Distinguished from BiGCN, our method constructs entity-sentence heterogeneous graphs, facilitating enhanced capture of structure features. Compared to ClaHi-GAT, we introduce Entity-Aware Heterogeneous Attention to discern the significance of different meta-paths automatically. Moreover, by leveraging chatGPT to extract knowledge and augment entities based on context, CICAN outperforms knowledge-based models, KMGCN and DDGCN, especially in dealing with complex semantic structures. The results confirm that CICAN can extract complex content features and capture semantic structure information, effectively enhancing text understanding and key feature extraction capabilities.

### 5.4 Ablation Study

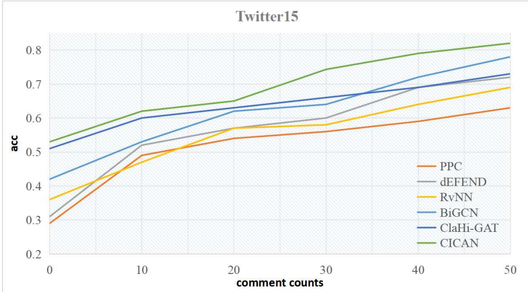
We conduct a series of ablation studies to evaluate the effectiveness of the individual modules of the CICAN model. The specific ablation experiments include the following aspects:

**w/o CIS**: Remove the crowd intelligence-based semantic feature learning module. **w/o CIS -TweetBERT**: Replace TweetBERT in CIS for BERT. **w/o CIS-cap**: Only remove the capsule network in CIS. **w/o CIS-BiLSTM**: Only remove the BiLSTM in CIS. **w/o KSS**: Remove the knowledge-based semantic structure mining module. **w/o KSS-ES graph**: Remove the entity-sentence heterogeneous graph in KSS. **w/o KSS-EE**: Remove the Context-Based Entity Enhancement in KSS.

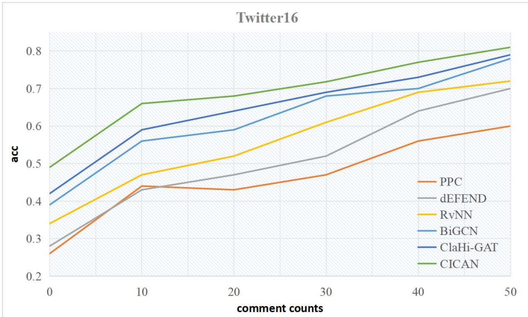
The experimental results are summarized in Table 2. w/o CIS has a significant decrease in the accuracy of the Twitter15 and Twitter16 datasets (decreased by 3.8% and 2.7%), which shows that this module plays a vital role in enhancing semantic representation and mining semantic information. Similarly, w/o CIS-TweetBERT also significantly impacts model performance, indicating that TweetBERT has superior context-dependent processing capabilities when processing text on social networks compared to BERT. Compared with w/o BiLSTM, the accuracy rate of w/o cap is lower because the capsule network can aggregate inputs from multiple directions when processing shorter texts, which is more robust. This result further validates the effectiveness of the method in this paper.

w/o KSS leads to 3.2% and 2.3% decrease in the accuracy of Twitter15 and Twitter16 datasets, respectively, emphasizing the significance of extracting structured knowledge and establishing global semantic connections to improve the model's understanding and judgment of text. The performance drops significantly when directly replacing entity-sentence heterogeneous graphs with directly concatenated entity results (w/o KSS-ES graph) because heterogeneous graphs can establish global semantic connections and contextual associations.

The comparison between w/o CIS and w/o KSS shows that w/o CIS significantly impacts the model's performance, indicating the effectiveness of the pre-trained model based on social network and multi-granularity feature extraction in fully mining features. However, introducing external knowledge (mainly involving modeling entity reinforcement and semantic structure) provides a complementary effect on the model's performance and enhances the model's generalization ability to unseen data.




(a) Twitter15




(b) Twitter16

Figure 3: Early Rumor Detection

## 5.5 Early Rumor Detection

One of the critical objectives in rumor detection is to identify and intervene in rumors as early as possible. In our experiments, we compare the performance of different methods by incrementally increasing the test data. As shown in Figure 3, even in the initial stage with limited comment quantities, our proposed CICAN model exhibited high accuracy. For instance, with only 30 comments, the CICAN model achieved accuracies of 74.3% and 71.8% on Twitter15 and Twitter16, respectively, demonstrating its ability to capture critical semantic features. It can be seen from the figure that the performance of all models fluctuates during training due to the increase of semantic information and noise in social networks. However, the CICAN model comprehensively considers semantic content and semantic structure information, so it has generalization and robustness and can consistently perform well. The CICAN model outperformed other models at each stage, confirming the effectiveness of early rumor detection and significantly improving detection performance.

# 6 Conclusion and Future Work

This study introduces a novel rumor detection model, the Crowd Intelligence and ChatGPT-Assisted Network (CICAN), which combines semantic content and structure information for rumor detection. In the crowd intelligence-based semantic feature learning module, we employ contextual semantic encoders and multi-granularity feature extractors to capture the semantic content information. In the knowledge-based semantic structure mining module, to capture complex semantic structural information, we extract refined knowledge utilizing ChatGPT to construct an entity-sentence heterogeneous graph and design entity-aware heterogeneous attention to integrate representations from different meta-paths effectively. Experimental results validate the effectiveness of the CICAN model in leveraging crowd intelligence and external knowledge, resulting in significant performance improvements in rumor detection tasks.

Future research directions can further expand the field of rumor detection in several ways. First, take inspiration from truth-finding systems that leverage human intelligence to extract reliable information from multiple sources. Second, to explore quantum cognition, using quantum theory to explain human behavior related to rumor detection. Finally, the application of large models in rumor detection can be further mined to improve the performance and interpretability of the models.

## Limitations

In this section, we discuss the limitations of our work and propose corresponding solutions. First, we exploit the semantic information of crowd intelligence for rumor detection. However, the user's profile, historical activity, or social network structure should also be considered. Future research will focus on capturing user features related to rumor spreading to improve detection accuracy. Second, ChatGPT shows promising results in data augmentation and concept extraction in this study. However, its effectiveness largely depends on the quality and design of the prompts used. Therefore, further research to explore more effective prompt engineering techniques, especially for customizing rumor detection tasks, is expected to improve model performance.

## Ethics Statement

This article deals with social media data, specifically social media data on Twitter. The benchmark datasets we experiment with for classification are publicly available in previous work, follow all Twitter regulations, and extract data through the official API. To mitigate the potential impact of using such data, all tweets related to the dataset were anonymized, and URLs were removed to ensure individual users' privacy and data security. This research aims to analyze and explore the methods and techniques of rumor detection to improve the credibility and reliability of information on social media. We will report the experimental results objectively, transparently, and comprehensively following academic norms and ethical standards.


## Acknowledgements

This work is supported in part by the Natural Science Foundation of China (grant No. 62276188), Tianjin Research Innovation Project for Postgraduate Students (grant No. 2021YJSB167).


## References


Abdulrahman I Al-Ghadir, Aqil M Azmi, and Amir Hussain. 2021. A novel approach to stance detection in social media tweets by fusing ranked lists and sentiments. *Information Fusion*, 67:29–40.

Muhammad Zubair Asghar, Ammara Habib, Anam Habib, Adil Khan, Rehman Ali, and Asad Khattak. 2021. Exploring deep neural networks for rumor detection. *Journal of Ambient Intelligence and Humanized Computing*, 12:4315–4333.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 549–556.

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.

Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. 2011. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96.

Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Sardar Hamidian, and Mona Diab. 2021. Active learning for rumor identification on social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4556–4565.

Yutao Feng, Jipeng Qiang, Yun Li, Yunhao Yuan, and Yi Zhu. 2023. Sentence simplification via large language models. *arXiv preprint arXiv:2302.11957*.

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 754–763.

Qi Huang, Junshuai Yu, Jia Wu, and Bin Wang. 2020. Heterogeneous graph attention networks for early detection of rumors on twitter. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Tanveer Khan, Antonis Michalas, and Adnan Akhunzada. 2021. Fake news outbreak 2021: Can we stop the viral spread? *Journal of Network and Computer Applications*, 190:103112.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wee Yong Lim, Mong Li Lee, and Wynne Hsu. 2017. ifact: an interactive framework to assess claims from tweets. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 787–796.

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*.

Bang Liu, Di Niu, Haojie Wei, Jinghong Lin, Yancheng He, Kunfeng Lai, and Yu Xu. 2018. Matching article pairs with graphical decomposition and convolutions. *arXiv preprint arXiv:1802.07459*.

Yang Liu and Yi-Fang Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. *Advances in neural information processing systems*, 29.

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.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

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

Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. 2021. Efficient-capsnet: Capsule network with self-attention routing. *Scientific reports*, 11(1):14634.

TB OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. *OpenAI*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Huyen Trang Phan, Ngoc Thanh Nguyen, and Dosam Hwang. 2023. Fake news detection: A survey of graph neural network methods. *Applied Soft Computing*, page 110235.

Mohiuddin Md Abdul Qudar and Vijay Mago. 2020. Tweetbert: a pretrained language representation model for twitter text analysis. *arXiv preprint arXiv:2010.11091*.

Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadataa novel approach to stance detection in social media tweets by fusing ranked lists and sentiments. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pages 985–994.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *Advances in neural information processing systems*, 30.

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*, pages 395–405.

Mengzhu Sun, Xi Zhang, Jiaqi Zheng, and Guixiang Ma. 2022. Ddgcn: Dual dynamic graph convolutional networks for rumor detection on social media. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4611–4619.

Franklin Tchakounté, Ahmadou Faissal, Marcellin Atemkeng, and Achille Ntyam. 2020. A reliable weighting scheme for the aggregation of crowd intelligence to detect fake news. *Information*, 11(6):319.

Peter Tolmie, Rob Procter, Mark Rouncefield, Maria Liakata, and Arkaitz Zubiaga. 2018. Microblog analysis as a program of work. *ACM Transactions on Social Computing*, 1(1):1–40.

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

Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032.

Youze Wang, Shengsheng Qian, Jun Hu, Quan Fang, and Changsheng Xu. 2020. Fake news detection via knowledge-driven multimodal graph convolutional networks. In *Proceedings of the 2020 international conference on multimedia retrieval*, pages 540–547.

Xuan Wei, Zhu Zhang, Mingyue Zhang, Weiyun Chen, and Daniel Dajun Zeng. 2019. Combining crowd and machine intelligence to detect false news on social media. *Mis Quarterly*.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Min Yang, Wei Zhao, Lei Chen, Qiang Qu, Zhou Zhao, and Ying Shen. 2019. Investigating the transferring capability of capsule networks for text classification. *Neural Networks*, 118:247–261.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, Tieniu Tan, et al. 2017. A convolutional approach for misinformation identification. In *IJCAI*, pages 3901–3907.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2019. Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In *2019 IEEE international conference on data mining (ICDM)*, pages 796–805. IEEE.

Huaiwen Zhang, Quan Fang, Shengsheng Qian, and Changsheng Xu. 2019. Multi-modal knowledge-aware event memory network for social media rumor detection. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1942–1951.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*.

Yongchun Zhu, Qiang Sheng, Juan Cao, Shuokai Li, Danding Wang, and Fuzhen Zhuang. 2022. Generalizing to the future: Mitigating entity bias in fake news detection. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2120–2125.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36.

## A Dataset Details

To evaluate our proposed approach, we conducted experiments on two publicly available benchmark datasets, Twitter15 and Twitter16. The statistical details of these datasets are summarized in Table 3.

Table 3: Statistics of the datasets

| | Veracity | Twitter15 | Twitter16 |
|---|---|---|---|
| **Training** | N | 262 | 144 |
| | F | 259 | 144 |
| | T | 262 | 142 |
| | U | 260 | 144 |
| | Total | 1043 | 573 |
| **Validing** | N | 75 | 41 |
| | F | 74 | 41 |
| | T | 75 | 41 |
| | U | 74 | 41 |
| | Total | 298 | 164 |
| **Testing** | N | 37 | 21 |
| | F | 37 | 21 |
| | T | 37 | 20 |
| | U | 37 | 21 |
| | Total | 149 | 82 |
| **Total** | N | 374 | 205 |
| | F | 370 | 205 |
| | T | 374 | 203 |
| | U | 372 | 205 |
| | User | 276663 | 173487 |
| | Com | 331612 | 204820 |

## B Implementation Details

We provide further details regarding the implementation of the models. The models were implemented using the PyTorch framework and executed on an NVIDIA Tesla V100 GPU. Specific parameter values were assigned to each module of the models:

In the crowd intelligence-based semantic feature learning module, the number of hidden units in the TweetBert layer is 768. Pre-trained model parameters are utilized, and their weights are frozen to prevent overfitting. The BiLSTM layer consists of 256 hidden units. The size of the convolutional kernels ($k$) is fine-tuned among the options {2, 3, 4, 5, 6} and eventually set to {2, 3, 4}. The number of convolutional kernels ($k\_num$) is fixed at 256. The capsule network layer contains a variable number of capsules ($c_{in}$) selected from {2, 4, 6, 8, 16}, with the final choice being 4. The output dimension of each capsule ($c_{dim}$) is set to 50. The number of iterations for capsule dynamic routing ($r$) is determined from {1, 3, 5, 7}, with the chosen value being 3. In the knowledge-based semantic structure mining module, we utilized GloVe 300d (Pennington et al., 2014) word embedding to encode individual entities and sentences. To enhance the effectiveness of Entity-Aware Hierarchical Attention, we experimented with different numbers of attention heads ($K$) and evaluated their impact on performance. Specifically, we explored the options {1, 2, 4, 6, 8} for $K$ and determined that setting $K$ to 4 yielded optimal results.

The cross-entropy loss function is employed during training to assess the disparity between the predicted results and the true labels. Through backpropagation, parameter updates are carried out using the Adam optimizer (Kingma and Ba, 2014). An early stopping (Yao et al., 2007) strategy is adopted to prevent overfitting. These parameter settings ensure reproducibility, fairness, and accurate experimental results.

## C Prompt Settings

Specifying clear instructions and tasks in the prompt is essential to enhance ChatGPT's performance in specific tasks. Therefore, we adopt an iterative experimental approach to optimize the final prompt, As shown in Figure 5. Prompt 1 is used for data augmentation operations in ChatGPT-based Knowledge Extraction, and Prompt 2 is used for entity linking and conceptualization in ChatGPT-

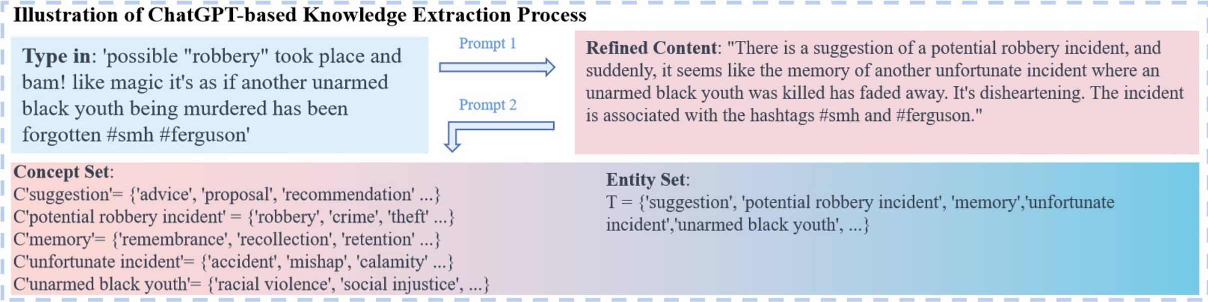



Figure 4: Illustration of ChatGPT-based Knowledge Extraction Process

**Prompt 1:**
You are a data processor, your task is to process the given rumor detection text data, and please ensure that the processed text:
1.The processed text should be abstract without changing any original meaning of the document.
2. The processed text should retain all important objects, concepts and the relationships between them.
3. The processed text should only contain information from the text.
4. The processed text should be clear and understandable without any abbreviations, special marks (such as hashtags) and spelling mistakes.
You need to generate the following result:
Type in: 'the text you need to process'
Refined content: 'the processed text'
Here is the text you need to process :

**Prompt 2:**
You are building a knowledge graph. Your task is to link and conceptualize entities from the given text. Please extract it step by step:
(1) Extract the entities in the given text. Entities can be nouns or noun phrases that refer to real-world objects or abstract concepts. You can use named entity recognition (NER) tools or part-of-speech (POS) taggers to identify entities.
(2) Link the extracted entities to the existing knowledge base or concept system. By comparing and matching the entity's context, entity attributes, etc., it is associated with the corresponding entity in the knowledge base, and the link between the entity and the knowledge base is established.
(3). For each identified entity, we obtain its concept information from the existing knowledge graph through conceptualization, adopting the "isA" relationship to obtain the concept. Conceptualization: Map the extracted entities to related concepts or topics. This step associates entities with a set of concepts or topics in order to better understand the semantic meaning and domain of entities.
(4) Most importantly, if you can't find any knowledge, output: "None"
You need to generate the following result:
Entity Set T = {'Entity 1', 'Entity 2', 'Entity 3', ..., 'Entity N'}
The set of concepts corresponding to each entity:
C'Entity 1' = {'Concept 1', 'Concept 2', 'Concept 3', ..., 'Concept N'}
C'Entity 2' = {'Concept 1', 'Concept 2', 'Concept 3', ..., 'Concept N'}
...
C'Entity N' = {'Concept 1', 'Concept 2', 'Concept 3', ..., 'Concept N'}
Here is the text you need to process:

**Prompt 3:**
You are a rumor detector, and your task is to judge whether an event is true, false, non-rumour, or undeterminable based on the source tweets and comment information given the event. You will see the source tweets and related comments of an event, please judge step by step according to the following requirements.
Notice:
1. When performing rumor detection, please consider the context and relevant information of the event, and judge its authenticity based on the text content. For source tweets and comments, watch for possible rumor characteristics such as expressions of uncertainty, lack of credible sources, conflicting information, etc.
2. Regarding the classification criteria, 'true' means that a true rumor is a situation that initially lacks evidence support but is confirmed to be true after fact verification, which may be caused by factors such as incomplete information, rapid spread, or inaccurate preliminary investigations, 'non -rumor' stands for non-rumor. The biggest difference between a true rumor and a non-rumor is whether there is no sufficient evidence to support it when it is first spread. 'false' stands for False Rumors are false information intentionally fabricated or spread to mislead audiences or to pursue specific interests. .When you encounter insufficient information or cannot determine the nature of the rumor, please select the tag 'unverified'.
3. You should only directly output the predicted labels for classification tasks. Your answer should be one of ['true', 'false', 'non-rumor', 'unverified'] , do not output a sentence, or other words.
4 When answering questions, try to find reasonable reasons or basis, and avoid subjective assumptions.
Example:
The judgment result you need to output: one of ['true', 'false', 'non-rumor', 'unverified']
Here's the source tweet and comment info entered:

* Due to space limitations, we give the prompt of 0-shot here, and the example of few-shot is omitted

Figure 5: Prompt instances used in the ChatGPT-based Knowledge Extraction part of the model

based Knowledge Extraction. The process of using ChatGPT for knowledge extraction is shown in Figure 4. Prompt 3 is designed to use ChatGPT alone for text classification tasks to enhance its understanding of text content and facilitate more accurate classification.

## D Quantitative Analysis of CICAN

This section provides a comprehensive analysis and effectiveness evaluation of the CICAN model.

**ChatGPT vs ChatGPT-Assisted**

We conducted experiments to compare the performance of using ChatGPT directly for downstream classification tasks versus using it as an auxiliary task. We developed prompts for ChatGPT (as shown in Prompt 3 of Appendix C) and conducted experiments with 0-shot, 2-shot, and 5-shot samples for each class in the target domain. The results of the experiments are shown in Figure 6.

It can be seen from the figure that the ChatGPT-Assisted method can improve the accuracy of text classification, but using ChatGPT alone for text classification may be limited because it is not designed explicitly for text classification tasks or pre-trained with data from social networks. On the contrary, using ChatGPT for knowledge extraction can combine its powerful generative ability with the contextual information of the original text, which helps the model capture the latent features of the text, thereby improving the accuracy of text classification. At the same time, the training of ChatGPT is very dependent on the quality of the prompt. Few-shot results are better than zero-shot because when dealing with zero-shot classification, ChatGPT generates labels or data that do not correspond to the given label set, resulting in lower accu-

Table 4: The effectiveness analysis of the methods

| Dataset | Entity Enhancement | | Hierarchical Attention | | CICAN |
|---|---|---|---|---|---|
| | CICAN-Non | CICAN-ConceptAtt | CICAN-Concat | CICAN-SemanticAtt | |
| Tweet15 | 0.827 | 0.838 | 0.840 | 0.846 | 0.855 |
| Tweet16 | 0.820 | 0.833 | 0.837 | 0.835 | 0.840 |

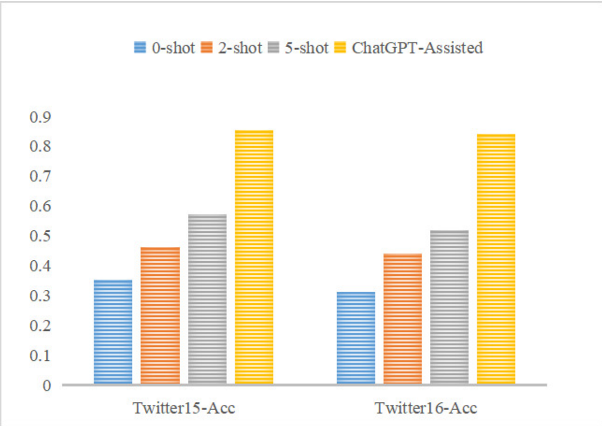



Figure 6: Performance comparison of ChatGPT and ChatGPT-Assisted

racy. To alleviate this problem, ChatGPT needs to be provided with examples that help it understand the content and facilitate accurate classification.

**Method Effectiveness Analysis**

We performed a comparative analysis with commonly used methods to demonstrate the efficacy of the optimization methods proposed in the model (Context-based Entity Enhancement and Entity-Aware Hierarchical Attention). Table 4 presents the experimental results of CICAN and the following models based on different modeling methods. According to different modeling methods, it mainly includes the following models.

1) CICAN-Non: No entity enhancement.

2) CICAN-ConceptAtt: Use Concept Attention to replace the Context-based Entity Enhancement method. Concept Attention is a method first proposed in MKEMN (Zhang et al., 2019) to measure the semantic similarity between concepts and entity representations.

3) CICAN-concat: Concat the representation of the meta-path to replace the Entity-Aware Hierarchical Attention method.

4) CICAN-SemanticAtt: Use Semantic Attention to replace the Entity-Aware Hierarchical Attention method. Semantic Attention is a method proposed in HAN (Wang et al., 2019) to calculate the importance of different meta-paths through nonlinear transformation.

**Effectiveness of Context-based Entity Enhancement Methods**

Compared to CICAN-Non, CICAN achieved an accuracy improvement of 2.8% and 2% on Twitter15 and Twitter16, respectively. Furthermore, compared to CICAN-ConceptAtt, CICAN demonstrated an accuracy improvement of 1.7% and 0.7% on Twitter15 and Twitter16, respectively. The context-based entity augmentation method shows the best performance in the rumor detection task. The method combines attention computation to capture the relationship between entities and the whole sentence and then applies attention weights to enhance the context-based entity enhancement representation. This finding underscores the significance of context-aware entity enhancement in the context of rumor detection tasks.

**Effectiveness of Entity-Aware Hierarchical Attention Method**

By comparing the experimental results, it can be observed that the performance of CICAN-concat and CICAN-SemanticAtt are comparable, while Entity-Aware Hierarchical Attention demonstrates relatively superior performance. Our Entity-Aware Hierarchical Attention performs relatively superior, increasing accuracy by 1.5% and 0.3% on the two datasets, respectively. Due to the complex and diverse associations among different meta-paths in the heterogeneous graph, the co-attention mechanism employed in Entity-Aware Hierarchical Attention can more effectively capture meta-paths' independence and collaborative interactions.

Overall, these results indicate the superiority of CICAN over its comparative models and highlight the importance of incorporating attention mechanisms and context-aware entity augmentation during the modeling process in the context of rumor detection tasks to achieve more accurate and robust rumor detection.