International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
SJIF (2025): 8.036

# A Short Text Classification Method Based on Cross-Source Heterogeneous Graphs and Adaptive Enhancement

**Senye Chen**

North China Electric Power University, School of Control and Computer Engineering, Changping District, Beijing, China

Email: csy13069313149[at]163.com

**Abstract:** ***Short text classification often suffers from semantic sparsity and limited contextual information. To address this challenge, this study proposes a short text classification model based on cross source heterogeneous graphs and adaptive augmentation. The method integrates short text data with external knowledge to construct a heterogeneous graph composed of words, entities, and part of speech information. A graph convolutional network is used to learn semantic representations of different node types. Text node association graphs are then enhanced through adaptive augmentation to generate complementary views. These views are fused to produce robust text representations, and contrastive learning is employed to improve semantic alignment for classification. Experimental results on MR, Twitter, and Snippets datasets demonstrate that the proposed model improves classification accuracy compared with several baseline methods. The results indicate that the proposed approach effectively enhances semantic representation and improves short text classification performance.***

**Keywords:** Short text classification, Graph neural networks, Contrastive learning, Adaptive augmentation

## 1. Introduction

In the era of big data, the ever-increasing volume of online text has led to an exponential growth in data size, prompting the emergence of numerous text processing methods to manage and organize this data. In daily life, the proliferation of short texts such as tweets, web search snippets, paper abstracts, and news feeds has made short text classification a fundamental task within text processing.

In short text classification tasks, deep learning approaches are widely adopted to tackle the problem. For instance, Lei et al. proposed a model integrating neural topic models with residual convolutional neural networks (CNNs) and bidirectional GRU networks, aiming to mitigate the issue of limited contextual information and enhance classification performance. However, this method still faces limitations in feature expansion, particularly in diversifying information sources and deepening data mining. Chen et al.[1] employed seed topic models to enrich information, addressing feature sparsity. Yet relying solely on external data for feature enrichment fails to maximize information utilization. Recent studies have introduced graph neural networks to tackle these challenges with promising results. For instance, Hu et al.[2] proposed HGAT, constructing a corpus-level graph to model latent topics, entities, and documents. However, HGAT fails to capture similarities among short documents—a critical property for propagating sparse labels across the graph. Building upon this, Wang et al.[3] proposed SHINE, which resolves this issue by constructing hierarchical heterogeneous graphs and dynamically learning short documents. However, the construction of its heterogeneous graph heavily relies on the completeness and timeliness of external knowledge bases. If the knowledge base fails to cover domain specific entities or update emerging concepts promptly, it may lead to missing entity node features or biased semantic associations, thereby weakening the model's representation capability and generalization performance in vertical domains or emerging scenarios.

Moreover, graph contrastive learning enables models to extract self-supervised signals from sufficient unlabeled data to enhance representation learning, driving its increasing adoption in short text classification. For instance, Sun et al.[4] proposed NC-HGAT, which combines neighborhood contrastive learning with a heterogeneous graph attention network. By computing neighborhood contrastive losses, this approach enables models to learn graph structural information with limited labeled data. However, contrastive views generated by random perturbations introduce noise or disrupt semantic relationships in text, leading to unstable capture of key features. To address this limitation, Liu et al.[5] proposed GIFT, which employs singular value decomposition (SVD) for low-rank reconstruction of the text-document matrix. This generates augmented views preserving global semantic associations while avoiding noise from local perturbations. However, its augmentation strategy remains dependent on static low-rank reconstruction, making it difficult to adapt to varying noise distributions across datasets.

To address these issues, this paper proposes CHG2A, a short text classification model based on cross-source heterogeneous graphs and adaptive augmentation. This model constructs a cross-source heterogeneous graph integrating external knowledge with internal data, employs GCN for semantic encoding of the graph structure, and combines an adaptive augmentation strategy. Through a hybrid optimization mechanism balancing static and dynamic elements, it achieves the dual objectives of preserving key information and suppressing noise interference, ultimately enabling efficient short text classification. The main contributions of this paper are summarized as follows: a) Designing a cross-source heterogeneous graph composed of a Wikipedia word graph, an entity graph, and a part-of-speech graph. This graph integrates the multidimensional semantic features of internal short texts with the authoritative syntactic information from external Wikipedia, effectively addressing knowledge gaps and coverage deficiencies. b) An adaptive augmentation method based on two-stage optimization is proposed. By performing edge denoising and criticality-guided edge optimization, complementary views are constructed that balance global semantics and local critical information without disrupting the graph structure. c) Experiments on three benchmark datasets demonstrate significant improvements over other algorithms.

## 2. Method

As shown in Figure 1, CHG2A first constructs a cross-source heterogeneous graph based on the short text dataset and Wikipedia data, which includes a Wikipedia word graph, an entity graph, and a part-of-speech graph. Each graph then undergoes a two-layer Graph Convolutional Network (GCN)[6] operation to obtain an updated feature matrix. Simultaneously, three corresponding text-node association graphs are constructed. Adaptive augmentation is applied to these graphs to generate two distinct augmented views. Information aggregation then yields two enhanced text representations, $Z_{aug}^{1}$ and $Z_{aug}^{2}$. Finally, joint optimization is achieved through weighted contrastive learning loss and cross-entropy loss.

### 2.1 Cross-Source Heterogeneous Graph Construction

To fully leverage textual and external information, the proposed cross-source heterogeneous graph consists of three components: a Wiki Word Graph, an Entity Graph, and a Part-of-Speech Graph.

**Wiki Word Graph.** This paper constructs a global pool and generates a word graph $\mathcal{G}_w = \{\mathcal{V}_w, A_w\}$ based on the WikiText[7] corpus. This corpus contains millions of tokens extracted from Wikipedia articles, utilizing only the summary portion of the data. Sentences are then annotated, and stop words and infrequent words appearing fewer than 10 times in the global pool are removed. This approach preserves more general information within the global pool.

Vw denotes the vocabulary set comprising 24,867 words from the global pool. $A_w \in \mathbb{R}^{|\mathcal{V}_w|\times|\mathcal{V}_w|}$ is an adjacency matrix determined by the similarity between each word pair, defined as $[A_w]_{ij} = \max(\mathrm{PMI}(v_w^i, v_w^j), 0)$, where PMI denotes the point mutual information. between words $v_w^i$ and $v_w^j$ Thus, Aw carries rich semantic information discovered from the Wikipedia global pool. Finally, the word graph node features $x_w^i \in \mathbb{R}^{|\mathcal{V}_w|}$ are initialized as one-hot vectors.

**Entity Graph.** The entity graph $\mathcal{G}_e = \{\mathcal{V}_e, A_e\}$ consists of entities from the knowledge graph, whose data provides supplementary information for text. Nodes $\mathcal{V}_e$ represent the entity set obtained via NELL knowledge base recognition. $A_e \in \mathbb{R}^{|\mathcal{V}_e|\times|\mathcal{V}_e|}$ is determined by the cosine similarity between each entity pair, i.e., $[A_e]_{ij} = \max(\cos(x_e^j, x_e^j), 0)$. Node features $x_e^i \in \mathbb{R}^{d_e}$ are composed of entity embeddings initialized via TransE[8] from entities residing in the knowledge graph.

**Part-of-Speech Graph.** The part-of-speech graph $\mathcal{G}_p = \{\mathcal{V}_p, A_p\}$ consists of part-of-speech tags (e.g., noun, verb) for words, indicating their syntactic roles and aiding disambiguation. Node $\mathcal{V}_p$ is the part-of-speech tag set obtained from NLTK's default POS tagger. Subsequently, this paper calculates the corresponding adjacency matrix $A_p \in \mathbb{R}^{|\mathcal{V}_p|\times|\mathcal{V}_p|}$ using PMI, where $[[A_P]_{ij} = \max(\mathrm{PMI}(v_p^i, v_p^j), 0)$. Finally, the node feature $x_p^i \in \mathbb{R}^{|\mathcal{V}_p|}$ is initialized as a one-hot vector.

### 2.2 Fusion of Graphical Representation

To effectively capture structured semantic features in short texts, the three constructed graphs each contain local dependencies between nodes that reflect different levels of linguistic structure. Therefore, by aggregating the semantic and structural features of nodes within the graph, deeper representations embedded in multi-source relationships can be further extracted.

This paper uniformly represents word-level component graphs of type $\tau$ as $G_\tau = \{V_\tau, A_\tau\}$, $\tau \in \{w, e, p\}$. Node features are uniformly expressed as $X_\tau \in \mathbb{R}^{|V_\tau|\times d_\tau}$, where the i-th row corresponds to a node feature $x_\tau^i$. These $G_\tau$ are used to capture pairwise relationships between nodes of the same type, unaffected by other types. For each graph class $G_\tau$, we apply a two-layer graph convolutional network (GCN)[6] to encode node embeddings H $\tau$. The definition is as follows:

$$H^{(\ell+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(\ell)}W^{(\ell)}\right) \tag{1}$$

Here, $H^{(\ell)}$ denotes the node embedding for layer $\ell$, $H^{(0)} = X$ represents the initialized features, $\hat{A}$ is the adjacency matrix with self-loops, $[\hat{D}]_{ij} = \sum_j[\hat{A}]_{ij}$ is the degree function, $W^{(\ell)}$ and $\sigma(\cdot)$ are trainable parameters and activation functions, respectively. The resulting node embedding is $H_\tau \in \mathbb{R}^{|V_\tau|\times d_\tau}$.

To map node representations to the text representation space, this paper constructs a text-node association graph for each node type to establish connections between text and nodes. For words or part-of-speech tags, this paper sets $M_\tau \in \mathbb{R}^{N\times|V_\tau|}$, $\tau \in \{w, p\}$ as the TF-IDF value between each text and the words or part-of-speech tags in the corpus, where N is the number of texts. For entities, this paper defines $M_e \in \mathbb{R}^{N\times|V_e|}$, where $M_{e,ij} = 1$if the i-th text contains the j-th entity, and 0 otherwise. Subsequently, this paper employs the following information aggregation operation:

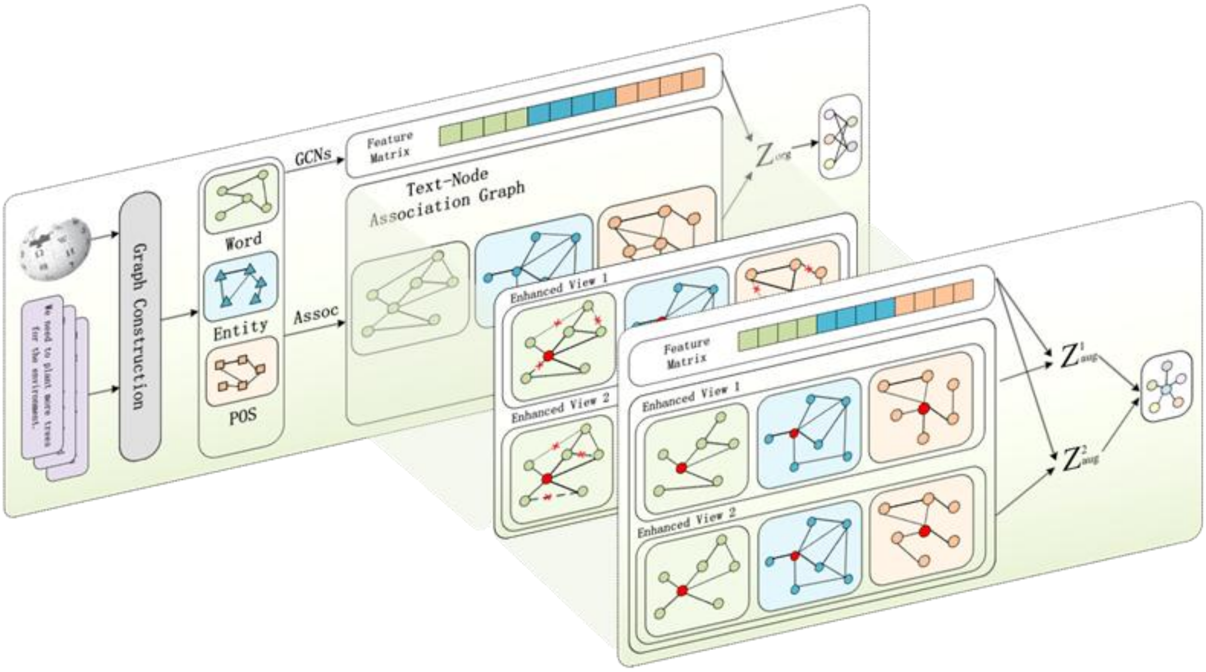


**Figure 1:** Overall network structure of CHG2A

$$Z_\tau = M_\tau H_\tau, \tau \in \{w, e, p\}$$
$$Z_{org} = Z_w || Z_e || Z_p \quad (2)$$

Among these, $M_\tau$ encodes the global semantic association strength of text nodes, while $H_\tau$ captures the local semantic features of nodes within the graph structure. These are aggregated to yield the overall semantic representation $Z_\tau$. Here, $Z_\tau \in \mathbb{R}^{N \times d_\tau}$ denotes text-specific features pertaining to nodes of type $\tau$. $Z_{org}$ is obtained by concatenating the text representations across the three graph structures.

For the original labeled text, this paper introduces a projection head $\Upsilon(\cdot)$ that maps the learned representations to a latent space. The cross-entropy loss function is then applied in this latent space, expressed as:

$$R = \sigma\left(W_{ce}\Upsilon(Z_{org})\right)$$
$$\mathcal{L}_{ce} = -\sum_{i \in \mathcal{D}_{lab}} \sum_{j}^{c} \mathcal{Y}_{ij} \log R_{ij} \quad (3)$$

Here, R denotes the model's predicted output, $\mathcal{Y}$ represents the ground truth label, and $W_{ce}$ signifies the trainable parameters.

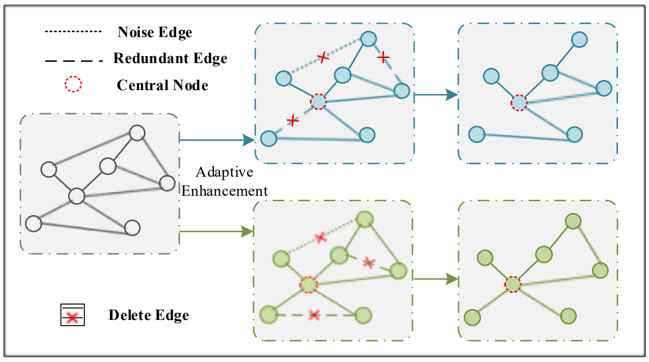


**Figure 2:** Adaptive augmentation

### 2.3 Adaptive Graph Data Augmentation

Relevant studies indicate that in contrastive learning, views derived from the same data through different transformations exhibit more stable and similar feature representations. Moreover, the model's learning objective is to extract stable features unaffected by structural variations[9]. Given this, this paper proposes a feature collaborative optimization enhancement method for graph-based text classification tasks. As shown in Figure 2, this method aims to maintain the critical structure of the text graph while applying moderate perturbations to non-critical parts. This approach enhances the quality of the text graph, thereby supporting model performance optimization.

#### 2.3.1 Adaptive Graph Data Augmentation

Rule-generated text graphs contain substantial noise that interferes with deep learning models. To mitigate noise from rule-generated text graphs, this paper sorts TF-IDF (Term Frequency-Inverse Document Frequency) values corresponding to words and part-of-speech tags, then removes noise edges based on the removal rate $r_t$ to initialize the text graph. This is defined as:

$$M_\tau = \mathrm{Matrix}\left(L_\tau \setminus \mathrm{MinIdx}(L_\tau, |L_\tau| \cdot r_t)\right), \tau \in \{w, p\} \quad (4)$$

Among these, $L_\tau$ denotes the set of edge weights in the text node association graph $M_\tau$, $r_t$ represents the deletion noise ratio, and $L \setminus \mathrm{MinIdx}(L, k)$ indicates the removal of these low-weight values from L.

#### 2.3.2 Key-Oriented Edge Optimization

In network science and analysis, the structural properties of nodes are crucial for identifying key elements within a graph[10]. Building upon the preliminarily optimized text graph described above, we further evaluate the importance of nodes and connections. By randomly removing non-critical connections, we preserve core structural information. This paper employs degree centrality[11] as a centrality metric to select important nodes within the graph. Subsequently, the original graph is perturbed with removal probabilities, where unimportant edges receive higher probabilities and important edges receive lower probabilities.

First, for an edge e connecting nodes u and v, let the centrality measures of its endpoints be $\phi(u)$ and $\phi(v)$, respectively. This paper employs the edge centrality $\Phi_{uv}$ from graph structural properties to measure edge importance, formally defined as:

$$\Phi_{uv} = \begin{cases} \frac{1}{2}\left(\phi(u) + \phi(v)\right) & \text{if G is not a directed graph} \\ \phi(u) & \text{if G is a directed graph} \end{cases} \quad (5)$$

Next, given the significant differences in node importance values, $\lambda_{uv} = \log\Phi_{uv}$ is set to normalize the data and mitigate the impact of nodes with high-density connections:

$$\lambda = \frac{\max(\lambda_{uv}) - \lambda_{vv}}{\max(\lambda_{wv}) - \operatorname{mean}(\lambda_{wv})} \quad (6)$$

The max() and mean() functions are used to calculate the maximum and average values of $\lambda_{uw}$ respectively. Edges with $\lambda$ values below 1.5 (i.e., edges deemed unimportant) are considered useful for information and are not removed. The final removal probability $p_{uw}$ is defined as follows:

$$p_{uv}^{k} = \begin{cases} \lambda \times p_k & \text{if } \lambda \geq 1.5 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Here, $k \in \{1，2\}$, and $p_k$ is the hyperparameter governing the overall probability of edge deletion.

Iterate through each element $M_{ij}$ in M. Based on the calculated edge deletion probability $p_{ij}^{k}$, decide whether to delete the edge, thereby determining the value of the corresponding element in M.

$$M_{ij}^{k} = \begin{cases} 0 & \mathit{probability}\ p_{ij}^{k} \text{ equal to } 0 \\ M_{ij} & \text{With probability } 1 - p_{ij}^{k} \end{cases} \quad (8)$$

Finally, edge denoising and key-directed edge optimization are performed on M to obtain the enhanced view. This view is then multiplied by the node feature matrix H to generate the semantic representation of the text under this enhanced view:

### 2.4 Comparative Learning

To fully leverage the complementary information embedded in the two augmented views, this paper introduces a contrastive learning mechanism. After obtaining the two text views $Z_{aug}^{1}$ and $Z_{aug}^{2}$, a nonlinear projection head $f(\cdot)$ is applied to map them to a latent space. Within this space, a contrastive loss is computed and subsequently normalized to yield $P_i = \operatorname{norm}(f(Z_i))$.

During the contrastive learning phase, each text sample generates two representations $P_i$ and $P_j$ through augmentation paths, forming a positive pair where i and j denote indices of the same text representation. Other distinct text representations constitute negative pairs. The paper employs a standard contrastive learning loss function (InfoNCE) to align positive pairs and separate negative pairs. The mathematical expression for the contrastive loss is:

$$\mathscr{L}_{cl} = -\sum_{(i,j)\in\square} \log \frac{\exp\left(\operatorname{sim}(P_i, P_j)/\tau\right)}{\sum_{k\in\square(i)} \exp(\operatorname{sim}(P_i, P_k)/\tau)} \quad (10)$$

Here, $\mathcal{P}$ denotes the set of positive sample pairs, $\operatorname{sim}(P_i, P_j)$ represents the normalized cosine similarity, and $\mathcal{N}(i)$ indicates the set of negative samples paired with $P_i$. $\tau$ is the temperature parameter, which regulates the contrast intensity between positive and negative samples.

By maximizing the similarity between $P_i$ and its corresponding positive sample $P_j$ while minimizing similarity with all other samples $P_k$, the model learns more robust and discriminative semantic representations.

Ultimately, the model is optimized through a combination of two loss functions, which can be expressed as:

$$\mathscr{L} = \eta\mathscr{L}_{cl} + \mathscr{L}_{ce} \quad (11)$$

Where $\eta$ is the control parameter.

During training, the text embeddings from the test set are input into the classifier $\Upsilon(\cdot)$ to obtain the corresponding evaluation metrics.

## 3. Experiments

### 3.1 Experimental Dataset

Three representative open-source short text classification datasets were selected for the constructed experimental dataset:

1) MR is a movie review binary classification dataset, where each review contains a sentence labeled as positive or negative.

2) Twitter is a binary classification dataset comprising a large collection of tweets expressing two emotions, gathered by NLTK.

3) Snippets consists of web search snippets returned by Google's search engine.

Detailed dataset information is shown in Table 1. For each category across all datasets, 40 annotated samples were randomly selected. Half were used for training, half for validation, and the remaining data for testing, simulating scenarios with limited annotated samples.

**Table 1:** Dataset statistics

| Dataset | Category Count | Training Set Count (Ratio) | Document Count | Average Length |
|---|---|---|---|---|
| MR | 2 | 40 (0.38%) | 10662 | 7.6 |
| Twitter | 2 | 40 (0.40%) | 10000 | 3.5 |
| Snippets | 8 | 160(1.30%) | 12340 | 14.5 |

### 3.2 Experimental Evaluation Criteria and Reference Methods

This paper employs the most commonly used precision evaluation metrics for classification tasks: accuracy (ACC) and F1-score (F1).

The model is primarily compared with the following types to demonstrate its superiority. (1) Traditional models: TF-IDF+SVM and LDA+SVM utilize TF-IDF features and LDA features respectively to represent text, then train SVM for classification. PTE learns word embeddings on a heterogeneous text graph and averages these embeddings as document embeddings. (2) Deep learning models: CNN and LSTM initialize text with pre-trained GloVe word embeddings before feeding them into corresponding deep networks. BERT is pre-trained on large corpora to generate context-aware embeddings for specific tasks. This paper uses BERT-base fine-tuned with classifiers on short texts. (3) Graph-based models: TLGNN, HyperGAT, TextING, and TextGCN model text as graph-structured data, leveraging graph neural networks (GNNs) to learn feature representations from node and edge topologies. (4) Deep short text classification methods include STCKA, STGCN, HGAT, and SHINE.

### 3.3 Experimental Environment and Model Hyperparameters

All experiments in this paper were conducted under identical software and hardware conditions. The model was constructed using the open-source deep learning framework PyTorch, with hardware configuration comprising an NVIDIA GeForce RTX 3090. The proposed model employs a two-layer GCN to encode each heterogeneous graph, with each hidden dimension set to 128. The model was optimized using the Adam method with a learning rate of 0.001 and a dropout rate of 0.5. The control parameter $\eta$ was set to 0.9, and the temperature parameter $\tau$ was set to 0.6.

### 3.4 Experimental Results and Analysis

This paper conducted extensive experiments on three benchmark datasets to compare the proposed model with other baselines. All experiments were repeated ten times to obtain averaged metrics.

**Table 2:** Test accuracy (ACC) and Macro-F1 (F1) of different models on three benchmark datasets. The best results are highlighted in bold

| Models | MR | | Twitter | | Snippets | |
|---|---|---|---|---|---|---|
| | ACC | F1 | ACC | F1 | ACC | F1 |
| TF-IDF | 54.29 | 48.13 | 53.62 | 52.46 | 64.70 | 59.17 |
| LDA | 54.40 | 48.39 | 54.34 | 53.97 | 62.54 | 56.40 |
| PTE | 55.02 | 52.62 | 54.24 | 53.17 | 63.10 | 59.11 |
| CNN | 59.06 | 59.01 | 57.29 | 56.02 | 77.09 | 69.28 |
| LSTM | 60.89 | 60.70 | 60.28 | 60.22 | 75.89 | 67.72 |
| BERT | 51.69 | 50.65 | 54.92 | 51.16 | 79.31 | 78.47 |
| TLGNN | 59.22 | 59.36 | 59.02 | 54.56 | 70.25 | 63.29 |
| HyperGAT | 58.65 | 58.62 | 59.15 | 55.19 | 70.89 | 63.42 |
| TextING | 58.89 | 58.76 | 59.62 | 59.22 | 71.10 | 70.65 |
| TextGCN | 59.12 | 58.98 | 60.15 | 59.82 | 77.82 | 71.95 |
| STCKA | 53.25 | 51.19 | 57.56 | 57.02 | 68.96 | 61.27 |
| STGCN | 58.25 | 58.22 | 64.33 | 64.29 | 70.01 | 69.93 |
| HGAT | 62.75 | 62.36 | 63.21 | 57.02 | 82.36 | 74.44 |
| SHINE | 64.58 | 63.89 | 72.54 | 72.19 | 82.39 | 81.62 |
| NC-HGAT | 62.16 | 62.14 | 63.76 | 62.94 | 82.52 | 74.62 |
| GITT | 65.21 | 65.16 | 73.16 | 73.16 | 83.73 | 82.35 |
| CHG2A | **66.54** | **66.50** | **74.72** | **74.71** | **85.16** | **84.84** |

Table 2 presents the accuracy and F1 scores of different models on the MR, Twitter, and Snippets test sets. Overall, the proposed CHG2A model achieves state-of-the-art performance across all three datasets, demonstrating its suitability for short text classification tasks. Specifically, traditional models like TF-IDF+SVM and LDA+SVM exhibit the lowest accuracy and F1 scores across all datasets. Analysis suggests these models overly rely on manually designed feature extraction methods, struggling to uncover complex semantic relationships within short texts. Among deep learning models, while CNNs and LSTMs can capture local information, they lack global semantic modeling capabilities due to limitations imposed by short text length and semantic sparsity. Furthermore, fine-tuned BERT pre-trained models underperform on key tasks due to insufficient labeled samples. Notably, graph-based models like TLGNN, HyperGAT, TextING, and TextGCN demonstrate strong competitiveness by modeling syntactic structure and benefiting from label propagation, though they fall short in information extraction and utilization. Among short text classification models,



Paper ID: SE26303203536 DOI: https://dx.doi.org/10.70729/SE26303203536

HGAT and SHINE demonstrate robust performance across multiple datasets. This stems from their adoption of heterogeneous information network architectures and attention mechanisms, enabling adaptive learning of weights for adjacent nodes. Furthermore, NC-HGAT employs random perturbations for view augmentation, yet its performance lags behind CHG2A, indicating that such augmentation methods fail to preserve most critical information in views. Notably, across three datasets, the CHG2A model outperforms other state-of-the-art models with accuracy improvements of 1.33%, 1.56%, and 1.43%, respectively. This advantage stems from two key aspects: (1) leveraging diverse critical information to construct heterogeneous graphs while integrating external knowledge bases to enrich semantic content; and (2) applying adaptive enhancement processing on the constructed text-node association graph, where the generated augmented views not only preserve crucial graph information but also eliminate noise.

## 4. Conclusion

This paper addresses the issue of insufficient key information extraction in short text classification caused by semantic sparsity and lack of external knowledge. It proposes a cross-source heterogeneous graph and adaptive augmentation-based short text classification model, CHG2A. By constructing a heterogeneous graph incorporating information from external corpora and integrating data from the Wikipedia word graph, entity graph, and part-of-speech graph, it effectively compensates for the deficiencies in semantic and syntactic information within short texts. Simultaneously, it introduces contrastive learning with an adaptive augmentation strategy, employing edge denoising and importance-guided edge optimization to generate dual views that complement global semantics with local core structures. Finally, contrastive learning aligns the dual-view representations fused with semantic embeddings, enabling deep semantic mining and robust representation of short texts. Experiments on three benchmark datasets demonstrate superior performance compared to other models. Future research will focus on dynamically updating external knowledge bases, optimizing augmentation strategies, and developing lightweight graph learning frameworks to further enhance the model's domain adaptability and robustness.

## References

[1] Chen Q, Yao L, Yang J. Short text classification based on LDA topic models[C]//2016 International Conference on Audio, Language and Image Processing (ICALIP). IEEE, 2016: 749-753.

[2] Yang T, Hu L, Shi C, et al. HGAT: Heterogeneous graph attention networks for semi-supervised short text classification[J]. ACM Transactions on Information Systems (TOIS), 2021, 39(3): 1-29.

[3] Wang Y, Wang S, Yao Q, et al. Hierarchical heterogeneous graph representation learning for short text classification[J]. arXiv preprint arXiv:2111.00180, 2021.

[4] Sun Z, Harit A, Cristea A I, et al. Contrastive learning with heterogeneous graph attention networks on short text classification[C]//2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022: 1-6.

[5] Liu Y, Huang L, Giunchiglia F, et al. Improved graph contrastive learning for short text classification[C]//Proceedings of the AAAI conference on artificial intelligence. 2024, 38(17): 18716-18724.

[6] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[7] Merity S, Xiong C, Bradbury J, et al. Pointer sentinel mixture models[J]. arXiv preprint arXiv:1609.07843, 2016.

[8] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[J]. Advances in neural information processing systems, 2013, 26.

[9] Xiao T, Wang X, Efros A A, et al. What should not be contrastive in contrastive learning[J]. arXiv preprint arXiv:2008.05659, 2020.

[10] Zhu Y, Xu Y, Yu F, et al. Graph contrastive learning with adaptive augmentation[C]//Proceedings of the web conference 2021. 2021: 2069-2080.

[11] Cai B, Zeng L, Wang Y, et al. Community detection method based on node density, degree centrality, and K-means clustering in complex network[J]. Entropy, 2019, 21(12): 1145.



Paper ID: SE26303203536 DOI: https://dx.doi.org/10.70729/SE26303203536