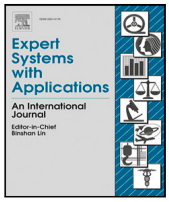
Expert Systems With Applications 251 (2024) 124069

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Edge-enhanced minimum-margin graph attention network for short text classification

Wei Ai [a], Yingying Wei [a], Hongen Shao [a], Yuntao Shou [a], Tao Meng [a,*], Keqin Li [b]

[a] College of Computer and Mathematics, Central South University of Forestry and Technology, Hunan 410004, China
[b] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Keywords:
Short text classification
Graph neural networks
Attention mechanism
Feature enhancement

ABSTRACT

With the rapid advancement of the internet, there has been a dramatic increase in short-text data. Due to the brevity of short texts, sparse features, and limited contextual information, short-text classification has become a challenging task in natural language processing. However, current methods primarily capture semantic information from locally-sequenced words in short text, which ignores the intricate feature relationships that pervade both the intra-text and inter-text. Therefore, this paper proposes a novel Edge-Enhanced Minimum-Margin Graph Attention Network (EMGAN) for short text classification to address this issue. Specifically, we construct a Heterogeneous Information Graph (HIG) to represent complex relationships among short text features. HIG mainly considers the relationship between document features and three attribute features, such as entities, topics, and keywords, and can represent short text features from multiple dimensions and levels. Then, to enhance the connectivity and expressiveness of the HIG for more effective propagation of feature information within it, we present a novel X-shaped structure edge-enhancement method. It enriches their relationships by reconstructing the edge structures. Furthermore, we design a Minimum Margin Graph Attention Network (MMGAN) for short text classification. Specifically, this method aims to explore the minimum margin between high-order neighbors and central nodes at the minimum cost, efficiently extracting and aggregating feature information. Extensive experimental results demonstrate that our proposed EMGAN model outperforms existing methods on five datasets, validating its effectiveness in short-text classification. Our code is submitted at https://github.com/w123yy/EMGAN.

1. Introduction

During the era of information proliferation, natural language processing (NLP) subtasks have undergone extensive scrutiny and found practical utility across many real-world predicaments (Hirschberg & Manning, 2015). Among these tasks, the challenge of text classification emerges as both a timeless quandary and an arduous undertaking (Chakraborty & Singh, 2022). As individuals increasingly acquire and disseminate information through diverse applications and websites, the succinct format of short texts, such as news tags, application reviews, instant messages, and tweets, has become an inseparable part of our daily lives. Its pervasive influence extends to various domains, including news categorization, social media (Kateb & Kalita, 2015), sentiment analysis (Balomenos et al., 2005), e-commerce, and spam filtering. Consequently, the role of short text classification proves indispensable in information retrieval. In light of its exceptionally high practical value, scholars diligently devote their efforts to exploring diverse methodologies (Yu, Ho, Arunachalam, Somaiya, & Lin, 2012).

Recently, deep neural networks have been proposed by researchers and widely utilized in the task of short text classification, such as convolutional neural networks (CNN) (Zhou, Li, Chi, Tang, & Zheng, 2022) and recurrent neural networks (RNN) (Graves & Graves, 2012; Zhou, Xu, Xu, Yang, & Li, 2016). Compared with traditional classification models, these models have achieved significant progress in short text classification (Pham, Nguyen, Pedrycz, & Vo, 2023). However, these models mainly focus on modeling sequential structural features, which significantly limits their ability to handle heterogeneous relationships among features. Graph neural networks (GNN) can solve the limitations of sequence models by explicitly modeling and utilizing the inherent graph structure of the data, and show excellent performance in processing complex semantic and topological information. Therefore, transforming text into graph structures (Wang et al., 2022; Wu et al., 2020) has become an increasingly popular approach in text classification tasks. As shown in Fig. 1, in such studies, it is customary

* Corresponding author.
E-mail addresses: aiwei@hnu.edu.cn (W. Ai), yingying.wei@csuft.edu.cn (Y. Wei), hongen.shao@csuft.edu.cn (H. Shao), shouyuntao@stu.xjtu.edu.cn (Y. Shou), mengtao@hnu.edu.cn (T. Meng), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.eswa.2024.124069
Received 18 December 2023; Received in revised form 5 April 2024; Accepted 18 April 2024
Available online 23 April 2024

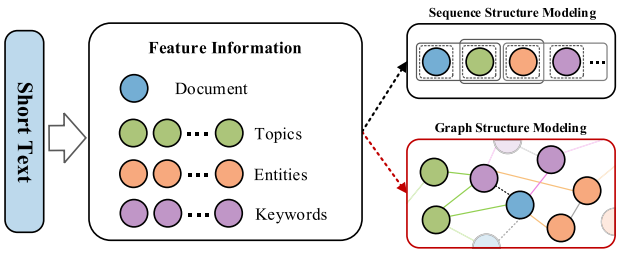


Fig. 1. A comparison of sequence and graph structures for modeling short text representations. In the sequence structure, the relationship between features is relatively simple, generally related to the context where the feature is located. In the graph structure, the relationship between features is more complex, and it is not limited to the context where the features are located and can represent the deep semantic relationship between features.

to construct a graph structure (Ragesh, Sellamanickam, Iyer, Bairi, & Lingam, 2021; Wang, Liu, Yang, Liu, & Wang, 2021) by treating text features (e.g., keywords, entities.) and their corresponding relationships as nodes and edges. This method can handle unstructured data, capture correlations among different features, and effectively address issues such as sparse features and data imbalance by leveraging the graph structure. By applying this approach, researchers like Joachims (2005) have achieved better classification results by exploring latent themes, documents, and word-level graph operations in a corpus. This graph structure can efficiently represent interactions and associations within textual data, leading to improved semantic information capture and enhanced classification performance.

However, due to the concise nature of short text sentences and their sparse semantic features, as well as weak contextual associations, the task of short text classification becomes increasingly challenging. Firstly, short texts require incorporating additional information and utilizing external knowledge bases to enhance feature representation. For example, Chen, Yao, and Yang (2016) used a seed topic model to expand the information to solve the problem of sparse feature information. However, enriching the features solely through topic representation does not maximize the utilization of information, posing a key concern regarding how to effectively augment feature information and semantic associations. Secondly, existing short text classification methods based on graph convolutional neural networks (Pham et al., 2023) often focus on aggregating first-order neighbor information within each layer while overlooking the capture of long-distance higher-order semantics. Dealing with distant information propagation often requires multiple stacked layers, For example, Zhang, He, and Zhang (2022) used multi-layer GCN to learn the features of the graph, leading to convergence issues and the potential loss of feature information. Hence, obtaining distant information is a worthy research challenge. Thirdly, short texts need more training data in practical scenarios, and manual annotation consumes time. In order to improve the classification effect, many scholars adopt a semi-supervised method based on graph neural network (GNN) to classify short texts with limited labeled data (Ai, Wang, Shao, Meng, & Li, 2023; Linmei, Yang, Shi, Ji, & Li, 2019). Among them, Wang, Wang, Yao, and Dou (2021) proposed a semi-supervised method for classifying brief texts using a heterogeneous graph neural network. This approach effectively utilizes limited labeled data and numerous unlabeled instances, propagating information through auto-generated graphs. However, it lacks interconnections between nodes of the same type, limiting its ability to capture document similarity and propagate labels. Thus, utilizing the limited labeled data remains a significant challenge.

To address the problems above, we propose a novel Edge-Enhanced Minimum-Margin Graph Attention Network (EMGAN) for short text classification. This method cleverly combines the edge enhancement technology and the minimum margin graph attention mechanism, which can optimize the overall topology and accurately capture high-order feature information, and is applied to heterogeneous information graphs for short text classification. Specifically, we construct a novel Heterogeneous Information Graph (HIG), which can well represent short text features and their complex relationships. HIG simultaneously considers entities, topics, and keywords as expanded features, addressing the inadequacy of short text features from multiple dimensions and perspectives. Then, we incorporate an edge-enhancement technique based on an X-shaped structure that reconstructs the edge structure between nodes, enriching relationships and forming a high-order HIG with dense and rich features. Furthermore, we also design the Minimum Margin Graph Attention Network (MMGAN) to address the feature aggregation issue in short-text classification. It utilizes edge-based higher-order attention, particularly focusing on exploring the minimal margin between high-order neighbors and center nodes at the lowest cost, facilitating feature extraction and aggregation, updating node features, reducing noise interference, and addressing the issue of sparse features in short texts. In short, EMGAN can effectively solve the sparse problem of short text features and significantly improve the model performance and classification accuracy.

The main contributions of this article can be summarized as follows:

- We introduce a novel Heterogeneous Information Graph (HIG), which takes document features as central nodes and considers three related attribute features: entities, topics, and keywords, which expands features from multiple dimensions, effectively addressing the limitations of short text features.
- Then, we incorporate an edge-enhancement technique based on an X-shaped structure that forms an X-shaped high-order heterogeneous graph by reconstructing the edge connections between different central nodes. It enhances the connectivity of HIG, thereby improving the propagation and interaction of feature information between nodes.
- We design the Minimum Margin Graph Attention Network (MMGAN) for short text classification, which centers around the central node and comprehensively explores the structure of the HIG at the lowest cost. It effectively aggregates the content of distant neighbor nodes to supplement the central node with rich feature information, thus resolving the issue of feature sparsity.
- We perform comprehensive experiments on real-world datasets encompassing news articles, concise comments, and search snippets to assess the efficacy of our model in comparison to eleven baseline approaches. The experimental findings unequivocally establish that our model surpasses the current state-of-the-art baseline methods on the benchmark datasets.

Due to the pervasive nature of short text across various domains but the challenge of sparse feature information, we propose EMGAN, which introduces a novel approach integrating Heterogeneous Information Graph (HIG), edge-enhancement technology, and Minimum Margin Graph Attention Network. The aim is to offer a richer understanding of short text content, thereby significantly enhancing the accuracy and effectiveness of text classification. In summary, EMGAN represents innovation in this field and underscores the pressing need for advancements in short text classification techniques.

The remainder of the paper is structured as follows: Section 2 reviews previous work. Section 3 describes our proposed method and model, including building HIG for short texts, edge-enhanced methods, and graph attention network models. In Section 4, we perform comprehensive experiments on the datasets and analyze the outcomes. Lastly, Section 5 concludes the paper, offering insights into the future research directions.

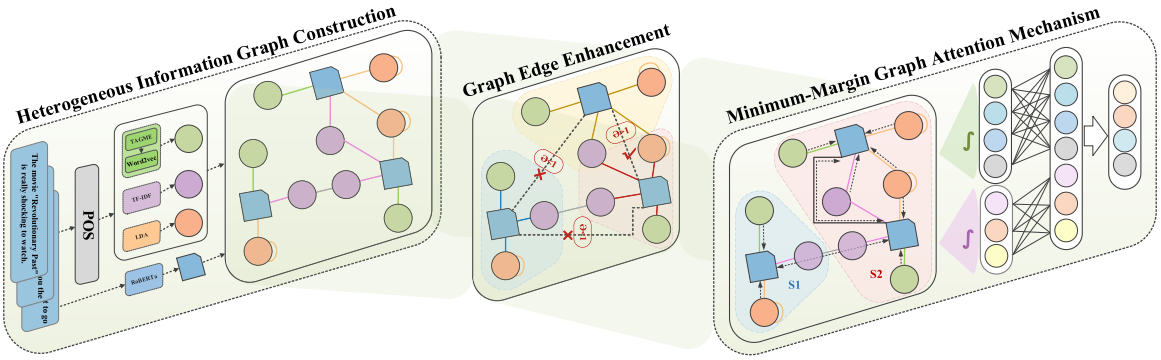


Fig. 2. The overall framework of the EMGAN model consists of heterogeneous information graph construction, graph edge augmentation, and minimum margin graph attention network.

2. Related work

This section presents an overview of pertinent literature concerning classifying concise textual content, encompassing both conventional approaches and deep neural network methodologies. Subsequently, we delve into contemporary research that explores the utilization of graph neural networks for short text classification, focusing on the current state of affairs.

2.1. Traditional short text classification

Text classification refers to extracting features from raw textual data and predicting categories for text data. Over the past several decades, researchers have introduced a multitude of models (Flisar & Podgorelec, 2020), including traditional machine learning algorithms such as NB (Lu, Chiang, Keh, & Huang, 2010; Xia, Wang, Chen, Duan, et al., 2018), Support Vector Machines (SVM) (Xia et al., 2020) and K-means (Joachims, 2005; Zhang, Yoshida, & Tang, 2008). However, traditional methods encounter a significant challenge of feature sparsity when dealing with short texts. Recent studies (Rousseau, Kiagias, & Vazirgiannis, 2015; Wang, Song, Li, Zhang, & Han, 2016) have employed graphical representations of text and extracted path-based features for text classification. Despite their initial success in formal texts, these approaches often fail to deliver satisfactory performance due to the inadequacy of short text features. In order to address this issue, many domestic and international researchers employ external corpora or leverage associated internal semantic information to enhance the features of short texts. For instance, Phan, Nguyen, and Horiguchi (2008) harnessed external corpora to extract latent themes from short texts. Wang, Chen, Jia, and Zhou (2013) introduced external entity information from the Wikipedia knowledge base to represent text. Yao, Bi, Huang, and Zhu (2015) enriched short text with semantic similarity information. However, these model architectures are relatively straightforward and have failed to fully unearth the latent characteristics of short texts, hence yielding limited classification efficacy.

2.2. Deep neural networks for short text classification

In recent years, with the continuous advancement of deep learning, text classification based on deep learning techniques has gradually emerged as the prevailing trend in natural language processing tasks (Wang, Wang, Zhang, & Yan, 2017). The most prominent advantage of deep learning methods over traditional text classification approaches lies in their efficient handling of text representation issues, enabling a more precise capture of textual features and achieving end-to-end problem resolution. The current explores various methodologies grounded in deep learning principles, including models based on long short-term memory networks (LSTM), recurrent neural networks (RNN), and convolutional neural networks (CNN). For instance, Wang et al. (2019) introduced a bidirectional RNN model enriched with an attention mechanism for short text classification. This model finds applications in health monitoring and the automated filtration of health-related tweets. RNN can effectively capture bidirectional information in sequence data, but it is prone to disappearance and explosion of gradients during the training process. LSTM can handle this problem better. Li et al. (2022) have devised a versatile distributed LSTM network that accommodates large-scale, high-velocity short text streams. However, it may struggle to capture complex patterns in lengthy sequences. In addition, because CNN can effectively capture local features and patterns, it also performs well in local feature extraction for tasks such as text classification. Zhou et al. (2022) have ingeniously devised a multichannel convolution framework based on CNN, thereby generating feature maps of diverse scales and facilitating the capture of semantic features spanning various dimensions. However, CNNs struggle to effectively capture long-range dependencies in lengthy text sequences due to their inherent local perception mechanism and fixed window size. The introduction of the Transformer architecture has effectively mitigated this issue. Bert, through its bidirectional encoding mechanism, comprehensively parses input text, capturing both local and global information, thus delving deeper into understanding the contextual relationships within the text. Cui, Wang, and Yu (2023) used a fusion model combining Bert and TextRNN. The Bert model uses the deep bidirectional Transformer component to build the entire model, thereby ultimately generating a deep bidirectional language representation that can integrate the context of both parties. However, the BERT model has certain restrictions on the length of the input text, which usually requires truncation or padding, which may result in the loss or redundancy of text information and affect the performance of the model. Moreover, it cannot establish relationships across texts. The multi-stage attention model can weighted average the importance of different positions, effectively handle variable-length sequences and capture long-distance dependencies. Meanwhile, Liu, Li, and Hu (2022) introduced a multi-stage attention model, amalgamating TCN and CNN, enhancing the model parallelism and overall efficiency. These innovative approaches have yielded commendable results in a multitude of NLP tasks. Nonetheless, there are still problems such as loss of useful information, relative complexity, and high computational consumption.

2.3. Graph neural networks for short text classification

Short text classification involves categorizing concise content using machine learning and data mining. Unlike extended text classification, it is more challenging due to length constraints. Short texts lack significant contextual details and strict syntactic structures, which are crucial for comprehensive text understanding (Wang et al., 2017). Therefore, methods customized for short text classification strive to integrate various auxiliary information to enrich short text representation. The

continuous development of graph neural networks (GNNs) has achieved the latest performance on short text classification. Here, we introduce the short text classification models in graph neural networks proposed in recent years. First, Defferrard, Bresson, and Vandergheynst (2016) proposed the usage of convolutional neural networks (CNNs) on graphs by treating text data as graph structures and applying local spectral filtering techniques. This approach significantly reduces computational complexity and has achieved notable results in text classification tasks. Subsequently, based on graph neural networks, Yao, Mao, and Luo (2019) designed a short text classification method that models words and texts as nodes in a graph, formulating text classification as a node classification problem. This approach can comprehensively integrate global information among texts, enhancing the understanding of text semantics and context. It adapts well to unstructured and irregular text data and is one of the earliest papers to propose this method. However, it only utilizes word semantic similarity information to enrich document representation, which is not enough for sparse short texts. Furthermore Ye, Jiang, Liu, Li, and Yuan (2020) found that the semantic information of word node representation and word order is very useful in short text classification. They developed a short text graph convolution network (STGCN) based on words, document relationships and text topic information, and combined the nodes into The representation is merged with word embeddings obtained by pre-training BERT. Yang et al. (2021) noticed the importance of attention mechanisms and proposed HGAT based on the double-layer attention mechanism, supplemented by additional relations and external knowledge bases to classify short texts. External knowledge bases are very helpful for short text classification because they can provide more features and initial knowledge for short texts, thereby elevating the precision of the model. However, it should be noted that using external knowledge bases can also cause noise interference. Recently, Jin, Sun, and Ma (2022) developed a concise method for short text classification using a dual-channel hypergraph convolutional network. This approach effectively learns two different representations of short text features. It enhances text embedding through an attention network, improving computational efficiency. Wu (2023) proposed a new heterogeneous graph attention network based on HGAT. The prior knowledge introduced in HIN enhances the semantic representation of short texts. Hua et al. (2024) integrated heterogeneous graph convolutional neural networks of text, entities and words, represented features through word graphs, enhanced word features through BiLstm, and predicted document categories. However, due to length constraints, GNNs, when dealing with short texts, typically do not consider adding additional information. Instead, they treat each short text as a single node in the graph, resulting in insufficient information features and poor performance. Despite most of these methods utilizing graphs to model texts, they neglect the influence of graph structure on short text attribute relationships. They also overlook the relevance of overall content features when dealing with feature attributes and relationships between texts, resulting in inadequate connections between nodes. To address the challenges of short text classification, we employ multiple features as nodes, enhance edges to enrich their relationships, and finally utilize efficient exploration and aggregation mechanisms.

Unlike the above existing studies, in this article, we address the issue of feature sparsity in short text classification by constructing a heterogeneous graph for short text corpora and using edge enhancement methods to rebuild relationships between nodes and enhance edge structures, thereby obtaining higher-order relationships. We propose a novel EMGAN model for classification that fully explores the structure of the heterogeneous graphs, further aggregates the adequate information of distant neighbor nodes into the attention mechanism, and dynamically extracts the critical characteristics of short texts instead of directly processing the entirety of the information.

3. Proposed method

3.1. The design of the EMGAN structure

In this section, we detail the design of the EMGAN structure. Fig. 2 visually shows the architecture of the EMGAN model proposed in this paper. Our model includes three key stages: (1) Heterogeneous information graph construction: In order to better represent the features of short texts, we utilize a heterogeneous information graph. Specifically, we first use a part-of-speech tagger to mark the part-of-speech (POS) of each word in the short text and then use different attributes (document, entity, topic, keyword) to model the short text as nodes and construct edge relationships to form heterogeneous information graphs. (2) Graph edge enhancement: We propose an edge enhancement method based on the X-shaped structure, which reconstructs the edge structure between nodes, enriches edge relationships, enhances the connectivity of the global topology of heterogeneous graphs, and forms multi-dimensional complex network relationships. (3) Minimum margin graph attention mechanism: We design a novel model (MMGAN) that uses a minimum margin graph attention mechanism to embed heterogeneous information graphs for short text classification. MMGAN can utilize information propagation along the graph to explore the structure of heterogeneous graphs at the lowest cost, fully utilize the characteristics of various types of nodes to integrate short text features, address the issue of sparse features in short texts, and attain superior outcomes in the classification of such texts.

3.2. Heterogeneous information graph for short texts

Due to the short text, there are problems such as short data and sparse features. In the case of such discontinuous vocabulary, modeling text as a graph structure is helpful for mutual learning of feature information between nodes. It transforms text classification tasks into graph classification tasks. Inspired by the model SHINE (Wang, Wang, et al., 2021), but different from it, we also introduce document and topic features to represent short texts. Specifically, our graph construction first uses part-of-speech taggers to reduce errors caused by ambiguity, then takes document features as central nodes, uses multiple attributes such as topics, entities, and keywords as nodes to compensate for missing features, and flexibly builds relationships between nodes. Specifically, entities serve as the subjects of events and can encapsulate rich information. The majority of entities possess information intrinsically tied to their respective domains. For example, the entity "Microsoft" often appears in the technology field. Topics represent the primary subjects or themes of discussion, providing insights into the central focus of the text. Keywords, important terms, or phrases highlight key information and aid in summarization and indexing. These elements enrich the representation of short text features, provide contextual understanding, and improve the differentiation between texts with similar features, thus compensating for feature deficiencies. They address the limitations of short text features by capturing different aspects of content and flexibly integrating rich relationships. The specific construction process of the heterogeneous information graph is shown in Fig. 3.

Here, we consider constructing a heterogeneous information graph $G = (V, B)$ consisting of documents, entities, keywords, and topic nodes, where $V = \{v_1, \ldots, v_n\}$ and $B = \{b_1, \ldots, b_m\}$ represent sets of nodes and edges respectively, n is the number of nodes, and m is the number of edges. In the node set of graph G, documents, entities, keywords, and topic nodes are represented by $D = \{d_1, \ldots, d_a\}$, $E = \{e_1, \ldots, e_y\}$, $W = \{w_1, \ldots, w_r\}$ and $T = \{t_1, \ldots, t_g\}$ respectively. In the short text, entities, keywords, and topic nodes are all connected to the document node (central node). The features of the central node are obtained by encoding the short text through the RoBERTa model. The construction of other nodes and edges is described in detail below.

First, in short texts, different parts of speech can create ambiguity. For instance, "uniform" can be categorized as "clothing" if used as a

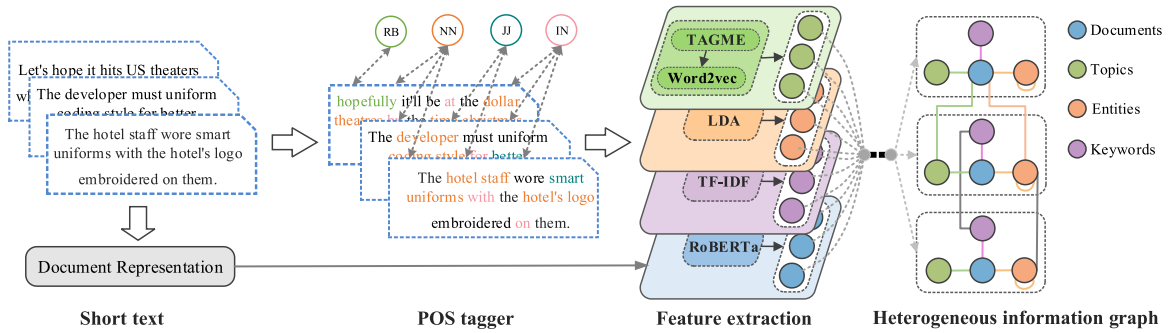


Fig. 3. Illustration of heterogeneous information graph for short text. (a) Use NLTK's to tag the words in the short text. (b) Use RoBERTa to represent the short text. (c) Extract and represent entities, topics, and keywords in short texts. (d) Construct the relationship between each feature to form a heterogeneous information graph.

noun. However, it does not belong to the "clothing" category if used as a verb. In order to eliminate ambiguity, we use a POS tagger to assign POS tags to the words in short texts, which are syntactic affixes such as nouns and verbs that mark each word in the short text. In particular, we utilize NLTK's default part-of-speech tagging to obtain the part-of-speech tags of each word in the document, resulting in a set of part-of-speech tag nodes $V' = \{v'_1, \ldots, v'_n\}$. We splice entities, keywords, and topics with corresponding parts of speech to eliminate semantic ambiguity.

Second, entities E in document D need to be identified to establish more prosperous edge relationships. Compared to the many keywords and topics in the document, the quantity of entities is considerably smaller, as most short documents encompass a single entity. We chose the entity-linking tool TAGME, which performs well in short texts. Using TAGME to link entities to Wikipedia, we obtain a set of entity nodes $E = \{e_1, \ldots, e_y\}$. If a document contains entities, we establish edges between the document and the entities. We then leverage the classic text embedding model word2vec to learn entity embeddings and measure the cosine similarity between each entity in all short texts. We predefine a threshold δ, and when the similarity is more remarkable than δ, we build edges between them and merge the two entity node information.

Third, we employed the LDA (Blei, Ng, & Jordan, 2003) topic model to extract latent topic T, as shown in Eq. (1). Topic modeling is a statistical model that clusters data based on the latent semantic meaning. This can help us enrich semantic relationships, especially by identifying latent words within documents or finding connections between similar documents without common words. The top $t_i = \{\epsilon_1, \ldots, \epsilon_z\}$ (where z represents the size of the lexicon) constitutes a conditional probability distribution across a collection of words. In order to avoid the interference of noise, We choose the foremost F words with the utmost probabilities as the topic words and allocate the document to these words of elevated likelihood. When assigning documents to topics, we establish edge relationships between documents and topics. For document d, the class label t_d can be predicted as the topic with the highest probability:

$$P(w|d) = \sum_{d} P(w|t) * P(t|d), \tag{1}$$

$$t_d = \arg\max_i P(w_i|d). \tag{2}$$

Fourth, we extract keywords from the tagged short texts to form a set of keyword nodes $W = \{w_1, \ldots, w_r\}$. We establish edges based on the inclusion relationship between documents and keywords. To extract keywords, We employ the term frequency-inverse document frequency (TF-IDF) computation technique, wherein the term frequency denotes the frequency of a word occurrence within a document. In contrast, inverse document frequency represents the logarithmically scaled reciprocal fraction. To establish edges between keywords that have co-occurrence relationships, We employ pointwise mutual information (PMI) to compute the weighting factor between two keywords. When PMI is positive, the keywords correlate more in the corpus, and edges are created between keywords with positive PMI values. Formally, the weight of the edge between node i and node j is defined as:

$$H_{ij} = \begin{cases} \mathcal{PMI}(i,j), & i,j \text{ are keywords;} \\ TF-IDF_{ij}, & i \text{ is document, } j \text{ is keyword;} \\ 1, & i = j; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The calculation method for the PMI value is as follows:

$$\mathcal{PMI}(i,j) = log\frac{\mathcal{P}(i,j)}{\mathcal{P}(i)\mathcal{P}(j)} \tag{4}$$

$$\mathcal{P}(i,j) = \frac{\Lambda(i,j)}{\Lambda} \tag{5}$$

$$\mathcal{P}(i) = \frac{\Lambda(i)}{\Lambda} \tag{6}$$

where $\Lambda(i,j)$ represents the number of sliding windows that contain both word i and word j, $\Lambda(i)$ signifies the count of sliding windows containing word i, and Λ denotes the total number of sliding windows contained in the entire corpus.

By using multiple attributes such as topic, entity, document, and keyword as nodes and specific relationships as edges to construct a heterogeneous graph, more abundant feature information can be obtained, thereby compensating for the semantic shortcomings of short texts and playing an important role in subsequent classification tasks (see Fig. 4).

3.3. *X*-Shaped structure graph edge enhancement method

In the realm of heterogeneous graphs, we amalgamate a plethora of attributes as nodes to enrich the informational fabric of the graph. Nevertheless, the brevity and sparsity of features in short textual data render such efforts insufficient. Furthermore, most heterogeneous graphs exclusively contemplate the characteristics of low-level neighboring nodes, thus failing to augment higher-level information. Therefore, we propose an edge enhancement method for heterogeneous graphs based on an X-shaped structure. The enhancement process is illustrated in Fig. 4. The core idea is as follows: Initially, within the constructed heterogeneous graph, we provide the following definition: An X-shaped structure is a substructure of a heterogeneous graph with a central node that connects to at least three different types (topics, entities, and keywords) of four nodes, forming a structure resembling the letter "X". The purpose of the X-shaped structure is to establish edge relationships between two different X-shaped structures when they share connections of the same node type. This maximizes the connectivity of nodes with feature relevance between the two central nodes. At this point, the information of the two central nodes can complement each other as features, and the original set of edges is merged with the new set to form a new total edge set. This structure facilitates the flow of feature information from nodes of other X-shaped structures toward

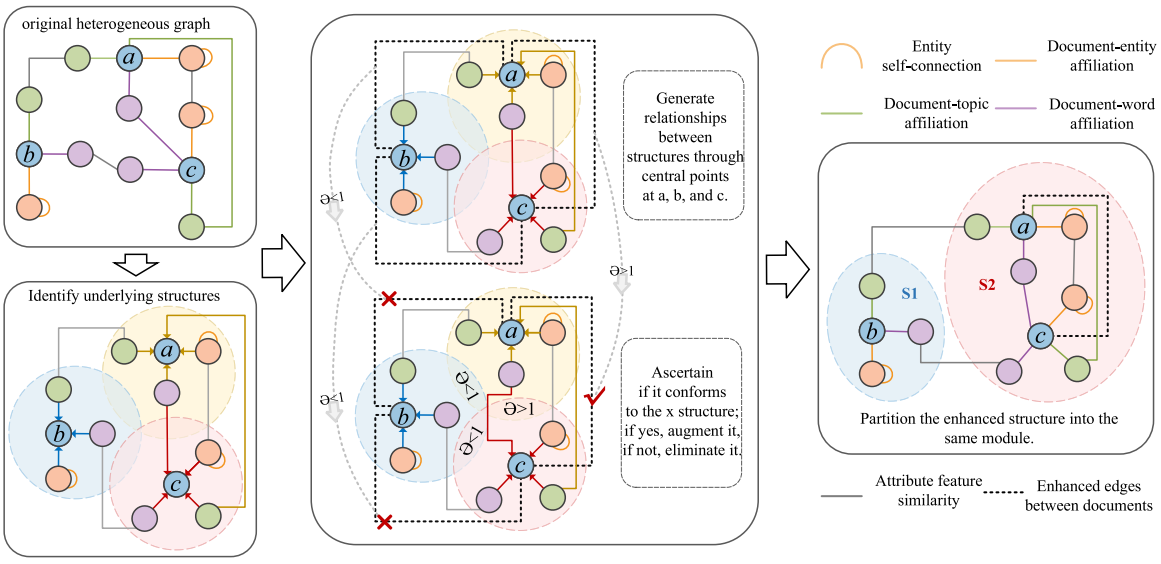


Fig. 4. Illustration of the edge enhancement method for heterogeneous information graph. (a) Identify the X-shaped structure in the heterogeneous information graph with the central node as the unit. (b) To measure the similarity between X-shaped structures, add an edge to the central node between similar structures.

its central node, aiding in capturing the multiple associations between short texts, topics, entities, and keywords. It enhances the connectivity of the heterogeneous graph's topological structure, enriching the feature information on the graph. This is vital for a better comprehension of the contextual and feature aspects of the text in classification tasks, ultimately leading to improved accuracy and effectiveness. This preparation sets the stage for capturing higher-order feature information for the model to be introduced in the following section. We will now explore how to employ edge enhancement techniques to enhance graph construction.

We first perform high-order encoding connections by constructing an X-shaped adjacency matrix A_X, where $(A_X)_{ij}$ is the number of X-shaped structure instances containing nodes i and j. The network diagram is represented as follows:

$$G^X = \left\{V', B^X\right\}, \tag{7}$$

where G^X represents a heterogeneous graph based on the X-shaped structure, V' represents the same set of nodes as the original heterogeneous graph, and B^X is the weighted edge set generated based on the X-shaped structure:

$$B^X = \left\{(k,l,\eta)_i | i \in \left\{1,\ldots,m_x\right\}\right\}, \tag{8}$$

where $k,l \in V'$ are the two endpoints of the i-th ($i \in \left\{1,\ldots,m_x\right\}$) edge and η represents the weight.

Next, we will first identify the connected structures based on the heterogeneous graph above, and any set of X-shaped connected structures are represented as follows:

$$\boldsymbol{\Phi} = \left\{\phi_i\right\}, \tag{9}$$

$$\phi_i = \left\{V'^{\phi}_i, B^X_i\right\}, \tag{10}$$

$$V'^{\phi}_i = \left\{t_i, e_i, w_i\right\}, \tag{11}$$

where $\boldsymbol{\Phi}$ represents the total node-set, ϕ_i ($i \in \left\{1,\ldots,m_{\boldsymbol{\Phi}}\right\}$) represents the set of nodes and edges for the ith structure, $V'^{\phi}_i \subseteq V'$ is the set of nodes in the ith connected structure, which includes entities e_i, topics t_i, and words w_i, and contains at least four nodes of three different types, and $B^X_i \subseteq B^X$ is the weighted edge set of the ith connected component.

The nodes that make up the X-shaped connected component represent feature attributes in the document and are connected by edges. This connected structure is a stable, X-shaped connected component that can better supplement feature information and possess higher-order structural capabilities.

Next, we shall refer to the set of nodes in the jth X-shaped structure as $V'^{\phi}_j = \left\{t_j, e_j, w_j\right\}$. If $\forall(t_i,e_i,w_i) \in V'^{\phi}_i$, we establish an edge relationship between the two structures, allowing the feature information of the ith and jth structures to complement each other. We denote this set of edges as follows:

$$B^{X'} = \left\{(\bar{k},\bar{l}) | \forall \bar{k},\bar{l} \in X_j, \forall j = 1,\ldots,m_{\boldsymbol{\Phi}}\right\}, \tag{12}$$

where $\bar{k}, \bar{l} \in V'$ represent the two endpoints of an edge between the ith and jth ($j \in \left\{1,\ldots,m_{\boldsymbol{\Phi}}\right\}$) central nodes in X-shaped structures.

We divide the X-shaped connected structure $\phi_Q \in \boldsymbol{\Phi}$ that has reconstructed edge relationships into the same modules. We choose Louvain (Blondel, Guillaume, Lambiotte, et al., 2008) as the module division method, and the input is each X-shaped connected structure ϕ_Q. We represent the modularization S (Newman & Girvan, 2004) as:

$$\begin{aligned} S &= \frac{1}{4\lambda}\sum_{ij}(A_{ij} - \frac{\gamma_i\gamma_j}{2\lambda})(\partial+1) \\ &= \frac{1}{4\lambda}\sum_{ij}\partial(A_{ij} - \frac{\gamma_i\gamma_j}{2\lambda}), \end{aligned} \tag{13}$$

where $\lambda = \frac{1}{2}\sum_i$ represents the total number of edges in the network, γ_i and γ_j denote the degrees of the ith and jth structural hub nodes, and $\frac{\gamma_i\gamma_j}{2\lambda}$ signifies the expected number of connections between these two structures. $\partial = \frac{\forall[(e,t,w)\cap(e\cup t\cup w)]}{\sum_{V'>4}(e,t,w)}$ represents the probability that two X-shaped structures share a common attribute, and if $\partial \geq 1$, the two structures can be linked and belong to the same module; if $\partial < 1$, then they do not belong to the same module. A_{ij} is the element in the adjacency matrix between central nodes i and j (the number of connecting edges between central nodes i and j).

The output is a module S composed of several connected structures $\phi_Q \in \boldsymbol{\Phi}$, put all modules together to obtain a module set, which we denote as $\left\{S_1,\ldots,S_{\bar{s}}\right\}$, $\bar{s}$ is the number of all modules obtained by the fusion of X-shaped connected components.

Finally, we perform a reconstruction of the relationships between nodes by strengthening the connectivity structure of each module in the set $\left\{S_1,\ldots,S_{\bar{s}}\right\}$, thereby enhancing the edge relationships and supplementing high-order structures with low-order structures to reinforce the topological structure of the graph. We use an X-shaped structure with better transitivity for connectivity. For nodes in the same module

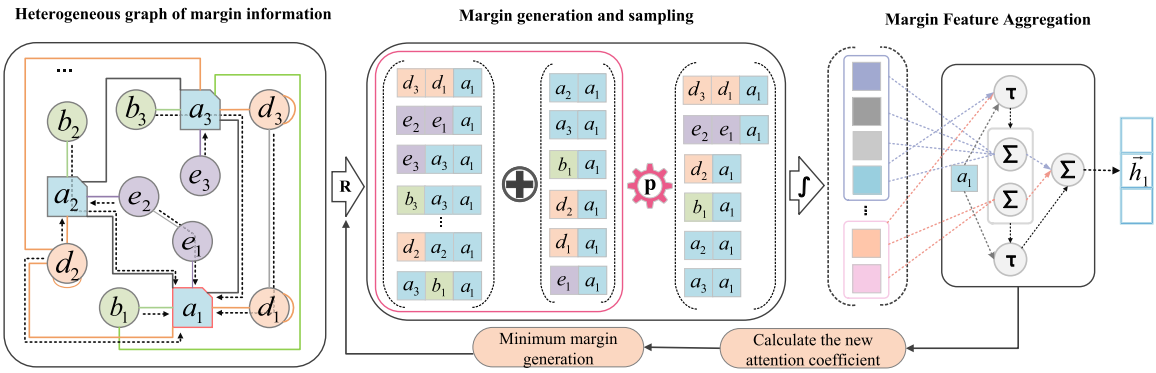


Fig. 5. Illustration of the minimum margin graph attention mechanism. (a) Computes the minimum margins from higher-order neighbor nodes (entity, keyword, and topic nodes) to other central nodes. (b) Calculate the attention coefficient from the high-order neighbor node to the central node according to the minimum margin.

$S_i \in \{S_1, \ldots, S_{\bar{s}}\}$ $(i \in \{1, \ldots, \bar{s}\})$, we allow feature information to complement each other, thus constructing a new set of edges, represented as:

$$B^*_{mod} = B^X \cup B^{X'}, \tag{14}$$

Based on the new edge set B^*_{mod}, the edge relationships of the original graph structure are strengthened to form a new network connectivity graph, and the documents with high feature correlation are maximally connected, expressed as:

$$G^*_{mod} = \{V', B^*_{mod}\}, \tag{15}$$

3.4. Minimum margin graph attention network

By performing edge enhancement on heterogeneous graphs, the connections between topological nodes in the graph are made complete, which provides a supplement for the problem of sparse feature representation in short texts. However, it is worth considering how to explore this high-order feature information. Most classification models only consider low-order feature information in the network, which cannot capture the high-order features in the graph. Although the graph has rich information, it cannot be captured at a high-order level, resulting in a significant performance loss. For example, in traditional models (Kipf & Welling, 2016), only the information of low-order nodes within a single layer is examined. The most common approach to capturing features of high-order neighbors is to stack multiple layers to expand the field of view. However, experimental results have shown that stacking multiple layers in the GAT model fails to expand the field of view and leads to performance degradation.

Therefore, this paper proposes a minimum margin graph attention network model that captures high-order topological features. The model can perform complete walks and exploration in heterogeneous graphs with high-order topological structures, finding the minimum distance between the central node and other attribute nodes, even for distant nodes, with the minimum cost. Then, the attention coefficients of the node distance and features are calculated for updating. By applying the minimum margin graph attention network, we can explore other nodes in the graph structure to obtain feature information and effectively aggregate this information into the central node, supplementing the short text content and improving the accuracy of short text classification tasks.

The overall process of the minimum margin attention mechanism is shown in Fig. 5. Firstly, we select the document node D in the heterogeneous graph as the center node. For each center node, we compute the minimum margin R of the high-order neighbors (keywords, topics, entities) of other center nodes to the center node with different lengths and extract their connection features as margin features. Then, we utilize the minimum margin attention mechanism to calculate the attention coefficients of these high-order neighbors to the center node. Finally, we iteratively update the features of each document using margin features and attention coefficients, aggregating information from other nodes to the center node.

In addition, the features of each node in our model are only related to the graph of topological structure. They are independent of the order of the node embedding features and neighboring nodes. During aggregation, the model relies on nodes and explores the minimum distance from the document node. Next, we will provide a detailed explanation of the model.

3.4.1. Minimum margin search and sampling

First, our input consists of the minimum margin R and node features h. In the initial stage, the minimum margin is R with uniform edge weights. This is done to minimize the loss of specialized tasks, such as cross-entropy loss in classification tasks. After training, the attention function generates edge weights based on learned attention coefficients.

Then, the minimum margin is calculated using Dijkstra's algorithm (Dijkstra, 2022), where the edge weights are first inverted and then transformed into positive values using the Suurballe method (Sidhu, Nair, & Abdallah, 1991). After computation, different attention coefficients have varying impacts on the edges. To ensure the stability of edge weights, we choose the attention coefficients of the network's last layer and take the average of all attention coefficients.

$$\eta_{ij} = \frac{1}{P} \sum_{p=1}^{P} \underline{f} \alpha_{ij}^{(p)} \tag{16}$$

where P represents the number of attention heads in a layer, $\underline{f}$ denotes the final layer, $\alpha_{ij}^{(p)}$ refers to the attention coefficient from node i to node j in the pth attention head, and η_{ij} represents the edge weight from node i to node j.

Let R_{ij}^c represent the minimum edge distance of length c between nodes i and j, where c is the length of an arbitrary edge, and let $\mathfrak{R}$ represent the set of such distances. The document nodes themselves are added to the set $\mathfrak{R}$. Within an edge distance of length c, we allow the document nodes to access nodes up to c hops away so that the maximum value of c can be used to control the size of the single-layer visual field.

For edges with the same minimum edge distance, those with higher costs in heuristics are less correlated with the document's features. In comparison, those with lower costs are more correlated. We sample the first p edges for a given central node and use the minimum cost, reducing computational pressure and highlighting the importance of more relevant edge distances. We represent the set of all sample edge distances as:

$$\mathfrak{I}_i^c = top_p(\mathfrak{R}^c), \tag{17}$$

$$p = \varphi_i * \mu, \tag{18}$$

where $\mathfrak{T}_i^c$ represents the set of all sample distances with a length c centered around node i, φ_i denotes the degree of node i, and p is determined by the degrees of the document nodes, ensuring the comparability of embedded features from distances of varying lengths. $\mathfrak{R}^c$ signifies a subset of R, encompassing all the shortest distances of length c. μ is a hyperparameter, representing the ratio between the number of sample distances and the degree of document nodes.

3.4.2. Aggregation of margin information

Margin aggregation is the cornerstone of our model. By meticulously exploring minimal margins, we select feature nodes that exhibit the lowest cost in proximity to document nodes while maintaining a high degree of feature relevance. Subsequently, we aggregate the feature information from these diverse nodes into the document nodes. This process enables capturing more intricate topological information by accommodating varying lengths of the shortest margins. Consequently, it augments the features of short texts. To this end, we have devised a dual-layer attention-based margin aggregation mechanism that addresses attention to identical and disparate edge distances. With attention to the same margin, for each document node i and the set of shortest margins $\mathfrak{T}_i^c$, we aggregate the features of each shortest margin of length c and represent the aggregated features as:

$$\zeta_i^c = \Theta_{p=1}^{P}\left\{\sum_{R_{ij}^c \in \mathfrak{T}_i^c} \alpha_{ij}^{(p)} \int\left(R_{ij}^{(p)}\right)\right\}, \tag{19}$$

ζ_i^c is the aggregated feature of node i concerning $\mathfrak{T}_i^c$, where $\mathfrak{T}_i^c$ is the shortest edge distance of length c centered at node i. The operator Θ represents the concatenation of all intermediate layer connections and the final layer averaging operation, which calculates the mean feature of all nodes in the edge distance. P is the number of attention heads for all edge distances of the same length c, and $\int$ maps edge distances of different lengths to a fixed length. $\alpha_{ij}^{(p)}$ is the attention coefficient between node i and edge distance R_{ij}^c, which can be expressed as:

$$\begin{aligned}\alpha_{ij}^{(p)} &= \tau\left(\vec{h}_i', \int\left(\left(R_{ij}^c\right)|\theta_\alpha\right)\right) \\ &= \frac{exp\left(\sigma\left[\theta_\alpha, \vec{h}_i' \,\|\, \int\left(R_{ij}^c\right)\right]\right)}{\sum_{\vec{h}_i' \in \mathfrak{T}_{\theta_\alpha}} exp\left(\sigma\left[\theta_\alpha, \vec{h}_i' \,\|\, \int\left(R_{ij}^c\right)\right]\right)},\end{aligned} \tag{20}$$

where τ represents the attention function, which outputs the attention between node feature h and the minimum edge margin R. $\vec{h}_i'$ refers to the linearly transformed features of sample node i, while θ_α denotes the parameters of the defined attention function τ. When we set $c = 2$, the generated attention coefficients are equivalent to the node attention that can be used to update edge weights. σ denotes any non-linear operation, and $\|$ represents concatenation. In the first level, $\mathfrak{T}_\theta$ represents the set $\mathfrak{T}_i^c$. The above is an aggregation of the same margins from the first layer.

The second layer focuses on variations in margin features of different lengths, utilizing an attention mechanism to capture embedded features of document nodes:

$$\vec{h}_i = \sigma\left\{\sum_{c=2}^{C} \beta_c \zeta_i^c\right\}, \tag{21}$$

where we set $c = 2$, the attention coefficient generated at this time is equal to the node attention that can be used to update the edge weight, C is the maximum allowed edge distance, and ζ_i^c is the aggregation feature of node i with edge distance c. β_c is the attention coefficient of ζ_i^c, which we express as:

$$\beta_c = \tau\left(\vec{h}_i, \zeta_i^c|\theta_\beta\right), \tag{22}$$

it can be derived from the identical attention mechanism at document node i by the attention function θ_β in this layer, where $\mathfrak{T}_\theta$ represents the collection of aggregate features $\mathfrak{T}_i^c$ for all nodes i regarding ζ_i^c.

At the initial stage, the entire network is updated iteratively based on h and R, with R generated using equal edge weights. As the network converges, R is regenerated based on the attention of the final layer, which is used for the next iteration.

After going through an f-layer EMGAN, We feed the obtained final embedding J of short text into a softmax layer for classification. Formally,

$$Z = softmax(J^{(f)}), \tag{23}$$

During the model training process, we minimize the model's loss using the cross-entropy loss function while employing L2 regularization to prevent model overfitting:

$$L = -\sum_{i \in D_{train}} \sum_{j=1}^{O} Y_{ij} \cdot log Z_{ij} + Y \|\Psi\|_2, \tag{24}$$

where O is the number of classes, D_{train} corresponds to the training dataset, Y_{ij} denotes the corresponding label matrix, Ψ stands for model parameters, and Y represents the regularization factor.

4. Experiments

To validate the availability and accuracy of our classification method, we conducted experiments on five real-world datasets and eleven baselines. This section describes the experimental setup, including the benchmark datasets, baseline algorithms, and parameter settings. Then, we compare methods and analyze the results.

4.1. Experimental setup

4.1.1. Datasets

In order to thoroughly assess the efficacy of our approach, we juxtapose it against cutting-edge methods in diverse scenarios. The evaluation encompasses five datasets: TagMyNews, Snippets, Ohsumed, MR, and Twitter. Table 1 provides a detailed depiction of these datasets.

- **TagMyNews**: This dataset contains 32,600 news articles collected from RSS feeds in English (Vitale, Ferragina, & Scaiella, 2012). The dataset has been filtered to exclude all titles. It includes articles from seven categories: sports, business, US, entertainment, world, health, and Sci.
- **Snippets**[1]: This dataset is published by Phan et al. (2008) search fragments returned by Web search engines, consisting of 12,340 short texts, divided into business, computer, health, sports, culture and arts, education and science, engineering, politics and society eight categories.
- **Ohsumed**[2]: This dataset is a medical dataset mixed with 7400 single-label samples and 6529 multi-label samples (Yao et al., 2019). We only used titles for short text classification, and documents with multiple labels were removed, including 23 cardiovascular disease categories.
- **MR**[3]: This dataset constitutes an English movie review corpus employed for binary sentiment classification (Pang & Lee, 2005). It encompasses two distinct categories: positive and negative sentiments, comprising 5,331 affirmative reviews and an equivalent number of pessimistic reviews, with an average sentence length of 20.
- **Twitter**[4]: It is a dataset of English tweets designed for binary sentiment classification. It consists of 5,000 positive and 5,000 negative tweets, allowing for the evaluation of our model's classification capability on social media.

[1] SnippetsandTagMyNewsaredownloadedfromhttp://acube.di.unipi.it:80/tmn-dataset/

[2] http://disi.unitn.it/moschitti/corpora.htm

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[4] http://www.nltk.org/howto/twitter.html#corpus_reader

Table 1
Summary statistics of datasets.

	#Docs	#Classes	#Avg.Length	#Words	#Docs with entities	#nodes	#edges	#X-structures
TagMyNews	32,549	7	5.1	38,629	86%	64,557	425,391	26,853
Snippets	12,340	8	14.5	29,040	94%	57,105	371,538	10,389
Ohsumed	7,400	23	6.8	11,764	96%	33,992	214,165	6,864
MR	10,662	2	7.6	18,764	76%	35,853	264,754	8,683
Twitter	10,000	2	3.5	21,065	65%	49,547	325,186	7,129

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

Model	Metrics	TagMyNews	Snippets	Ohsumed	MR	Twitter
CNN-rand	ACC	28.76	48.34	35.25	54.85	52.58
	F1	15.82	42.12	13.95	51.23	51.91
CNN-pretrain	ACC	57.12	77.09	32.92	58.32	56.34
	F1	45.37	69.28	12.06	57.99	55.86
LSTM-rand	ACC	25.89	30.74	23.30	53.13	54.81
	F1	17.01	25.04	5.20	52.98	53.85
LSTM-pretrain	ACC	53.96	75.07	29.05	59.73	58.20
	F1	42.14	67.31	5.09	59.19	58.16
TextGCN	ACC	54.28	77.82	41.56	59.12	60.15
	F1	46.01	71.95	27.43	58.98	59.82
HGAT	ACC	61.72	82.36	42.68	62.75	63.21
	F1	53.81	74.44	24.82	62.36	62.48
STGCN	ACC	34.74	70.01	33.91	58.18	64.33
	F1	34.01	69.93	27.22	58.11	64.29
SHINE	ACC	62.50	82.39	45.57	64.58	72.54
	F1	56.21	81.62	30.98	63.89	72.19
STHCN	ACC	63.44	83.45	46.07	64.81	73.24
	F1	56.28	78.17	31.28	64.44	73.01
ST-Text-GCN	ACC	65.43	85.78	46.42	68.44	75.23
	F1	58.72	80.63	32.14	66.54	74.36
Bert+ TextRNN	ACC	69.08	87.54	40.69	61.73	68.44
	F1	61.53	84.31	19.37	60.42	66.95
WC-HGCN	ACC	67.26	86.33	47.72	68.93	75.63
	F1	60.19	82.20	35.59	67.35	75.81
EMGAN(ours)	ACC	**70.13**	**88.06**	**50.85**	**71.04**	**77.82**
	F1	**65.68**	**84.52**	**37.79**	**70.38**	**76.64**

In our experiment, we preprocess all datasets, encompassing the filtration of special characters, segmentation, elimination of stop words, and removal of low-frequency words occurring less than five times. Table 1 presents comprehensive information about the dataset, encompassing document count, category quantity, average sentence length, word count, and the proportion of documents containing entities. In our dataset, most text (approximately 80%) incorporates entities. Regarding the MR dataset, we refrained from word deletion after performing data cleansing due to the brevity of sentences.

Regarding dataset allocation, we randomly sampled 40 labeled short-text documents per class. Half were used for training, and the other half for parameter tuning validation. In addition, we randomly sampled 1,000 unlabeled documents for training, in which HIG is generated in the training set. In addition, we selected 1,000 unlabeled documents for training. Most texts contained entity attributes and two pre-trained word embedding models, Word2vec and TF-IDF. We then performed part-of-speech tagging on the short text words, extracted entity, topic, and keyword attributes from the corpus, and established edge relationships based on rules to form a short text heterogeneous graph.

4.1.2. Baselines

To comprehensively evaluate the performance of our proposed short text classification method, we compared it with eleven baseline methods, as detailed below:

- **CNN**: Kim (2014) proposed the renowned convolutional neural network (CNN) in deep learning. Our experiments utilized two CNN variations: CNN-rand with random word embeddings and CNN-pre with pre-trained embeddings.
- **LSTM** (Liu, Qiu, & Huang, 2016): The model excels at handling sequential data and utilizes the last hidden state to represent the entire text, making it widely applicable for tasks involving textual data processing.
- **TextGCN**[5]: TextGCN (Yao et al., 2019) applies graph convolutional networks to represent a text corpus as a graph, capturing informative features by treating words as nodes. This method transforms text classification into node classification.
- **HGAT**[6] (Yang et al., 2021): The Heterogeneous Graph Attention Network is used to model entities, topics, and document corpora by embedding HIN. It is employed for short text classification based on a dual attention mechanism.
- **STGCN**[7] (Ye et al., 2020): The model represents words, topics, and documents in a corpus as a graph, combines the node representations obtained through Bi-LSTM and Bert word embeddings, and is directly fed into a softmax layer for classification.
- **SHINE**[8] (Wang, Wang, et al., 2021): SHINE models the corpus as a layered heterogeneous graph composed of word-level components, incorporates rich feature information, dynamically learns graph representations of short documents, and facilitates effective propagation of similar short text labels.
- **STHCN** (Jin et al., 2022): STHCN devised a short text classification method utilizing a dual-channel hypergraph convolutional network. This approach learns two distinct representations of short text features. It combines them using an attention network to enhance the embedding of short text.

[5] https://github.com/yao8839836/text_gcn
[6] https://github.com/ytc272098215/HGAT
[7] https://github.com/yzhihao/STGCN
[8] https://github.com/tata1661/SHINE-EMNLP21

Table 3
Test the accuracy and F1 scores of different models of the X-shpaed structure edge enhancement method.

Model	Metrics	TagMyNews	Snippets	Ohsumed	MR	Twitter
HGAT-X	ACC	62.45(+0.73)	83.11(+0.75)	43.35(+0.67)	63.47(+0.72)	63.88(+0.67)
	F1	54.42(+0.61)	75.08(+0.64)	25.40(+0.58)	63.05(+0.69)	63.09(+0.73)
STGCN-X	ACC	35.15(+0.41)	70.50(+0.49)	34.28(+0.37)	58.54(+0.36)	64.75(+0.42)
	F1	34.39(+0.38)	70.28(+0.35)	27.51(+0.29)	58.45(+0.34)	64.67(+0.38)
SHINE-X	ACC	62.71(+0.21)	82.63(+0.24)	45.74(+0.17)	64.81(+0.23)	72.79(+0.25)
	F1	56.36(+0.15)	81.79(+0.17)	30.87(+0.11)	64.08(+0.19)	72.40(+0.21)
ST-Text-GCN-X	ACC	65.84(+0.41)	86.21(+0.43)	46.76(+0.34)	68.74(+0.30)	75.68(+0.45)
	F1	59.03(+0.31)	81.02(+0.39)	32.43(+0.29)	66.82(+0.28)	74.78(+0.42)
WC-HGCN-X	ACC	67.89(+0.63)	87.00(+0.67)	48.30(+0.58)	69.54(+0.61)	76.22(+0.59)
	F1	60.78(+0.59)	82.81(+0.61)	36.11(+0.52)	67.92(+0.57)	75.35(+0.54)
EMGAN(ours)	ACC	**70.13**	**88.06**	**50.85**	**71.04**	**77.82**
	F1	**65.68**	**84.52**	**37.79**	**70.38**	**76.64**

- **ST-Text-GCN**[9] (Cui, Wang, Li, & Welsch, 2022): The model utilizes self-training on text data, incorporating keywords into the training dataset. The tagged information propagates along the structure of the manifold to the target samples.
- **Bert+ TextRNN** (Cui et al., 2023): This method uses a fusion model that combines Bert and TextRNN to finally generate a deep bidirectional language representation that can integrate the context of both parties.
- **WC-HGCN** (Yang, Liu, Zhang, & Zhu, 2023): It Introduces the concept of word information to enhance the feature representation of short texts and construct a text-level heterogeneous graph for each sentence by using words and relevant concepts as nodes and updating the nodes through the designed strategy.

For all the baseline methods mentioned above, we first preprocess our dataset and run the source code provided by the authors. Some may choose to present the results reported in previous research papers (The results are directly displayed, including some baseline data from HGAT and SHINE, while the remaining data is acquired by running the source code.). Entity information is obtained from Wikipedia. For example, CNN and LSTM deep neural networks use entity embeddings trained on the same Wikipedia corpus. TextGCN, HGAT, STGCN, SHINE, STHCN, ST-Text-GCN, and WC-HGCN choose to capture feature information by constructing graphs for better classification performance. We select these baseline methods to perform better comparisons.

4.1.3. Parameter settings

Our approach has been validated by selecting optimal parameter values for g, T, and δ to achieve the best performance. For constructing the heterogeneous graph, we set the similarity threshold δ between entities to 0.5 for all datasets, select the top $F = 2$ words with the highest probabilities as the topic words, and assign documents to these high-probability words. In the LDA topic model, we set the number of topics to $g = 20$ for the Snippets dataset, $g = 15$ for the TagMyNews, MR, and Twitter datasets, and $g = 40$ for the Ohsumed dataset. We implement EMGAN in PyTorch and use the Louvain partitioning method. For all datasets, we set μ to 1.0, signifying that the number of sampled margins equals the degree of each node. By defining the maximum value of c, we can control the size of the single-layer receptive field. We set the maximum value of c to 3 in the first layer and 2 in the second layer. The learning rate is established at 0.005, the dropout rate is set to 0.5, and the number of iteration steps is fixed at 8. We employ the Adam optimizer for training, and if the validation cross-entropy loss does not decrease continuously for 10 consecutive epochs, the training process is halted. The experiment utilized two evaluation metrics, namely accuracy and F1 score, to measure the performance of short text classification. All methods were executed on a computer with an i7-9700kF CPU and an RTX3090 GPU.

[9] https://github.com/wanggangkun/ST-Text-GCN

4.2. Experimental results and analysis

In the comparative experiments, to verify the excellent classification performance of the proposed short text classification method, we compared it with CNN, LSTM, Text GCN, HGAT, STHCN, STGCN, SHINE, ST-Text-GCN, and WC-HGCN. Table 2 demonstrates the classification outcomes of various techniques across five benchmark datasets. Our approach surpasses all baseline methods across all datasets, showcasing the effectiveness and superiority of our proposed method in the domain of short text classification with sparse features.

Upon careful analysis, we observed varied performance among CNN-Rand, CNN-pretrain, LSTM-Rand, and LSTM-pretrain. While both CNN and LSTM utilize pre-trained word embeddings, CNN excels in capturing contiguous and close-range semantics. Therefore, pre-training on the Snippets dataset is more effective for CNN. TextGCN and STGCN models, based on graph neural networks, have achieved results comparable to the deep models CNN-Pretrain and LSTM-Pretrain. ST-Text-GCN is an enhancement of the TextGCN model. It augments the training set with self-training, thereby incorporating keywords and leading to significantly higher accuracy. This is attributed to the ability of the text graph to capture both document-word relationships and global word-word relationships. However, when we compare TextGCN with HGAT, the overall accuracy is relatively lower. This is because HGAT incorporates heterogeneous information network structure (HIN) and attention mechanisms, allowing it to learn the weights of neighboring nodes adaptively. This highlights the superiority of heterogeneous graphs and attention mechanisms. Consequently, the accuracy of STHCN, which combines attention networks, also performs well. SHINE has demonstrated strong performance on numerous datasets, and the analysis suggests that its dynamic learning of short document graphs can facilitate effective label propagation. Bert+ TextRNN achieves remarkable performance by leveraging Bert pre-training and TextRNN model to capture temporal information and long-distance dependencies in the text. Especially on the Snippets dataset, it achieves an accuracy of 87.54%, second only to our model. In contrast, WC-HGCN introduces conceptual information about words to enrich the feature representation of short texts, constructing a text-level heterogeneous graph for each sentence. Compared to the models above, it has achieved superior results. Furthermore, by comparing TextGCN, STGCN, and SHINE, we observe that models based on graph neural networks can achieve excellent results in short texts, indicating that graph structures can extract advanced semantic features from sentences. Meanwhile, a comparison of the performance between HGAT and WC-HGCN demonstrates that incorporating external knowledge can bolster the semantic richness of sentences, effectively addressing the issue of sparsity in short-text features. As a result, our EMGAN model outperforms other state-of-the-art models in terms of performance on five different datasets, with improvements in accuracy of 2.24%, 1.73%, 3.13%, 2.11%, and 2.19%, underscoring the efficacy of our approach. This can be attributed to several factors: (1) We utilize a variety of crucial pieces of information to construct a heterogeneous graph, incorporating external knowledge bases to enrich

Table 4
Test accuracy and F1 score for different models of Min-margin graph attention network.

Model	Metrics	TagMyNews	Snippets	Ohsumed	MR	Twitter
HGAT-MM	ACC	62.17(+0.45)	82.83(+0.47)	43.00(+0.32)	63.16(+0.41)	63.65(+0.44)
	F1	54.22(+0.41)	74.88(+0.44)	25.10(+0.28)	62.72(+0.36)	62.86(+0.38)
STGCN-MM	ACC	35.37(+0.63)	70.69(+0.68)	34.48(+0.57)	58.79(+0.61)	64.97(+0.64)
	F1	34.60(+0.59)	70.53(+0.60)	27.73(+0.51)	58.67(+0.56)	65.17(+0.58)
STHCN-MM	ACC	64.18(+0.74)	84.23(+0.78)	46.72(+0.65)	65.50(+0.69)	73.96(+0.72)
	F1	56.95(+0.67)	78.70(+0.73)	31.87(+0.59)	65.06(+0,62)	73.66(+0.65)
ST-Text-GCN-MM	ACC	66.00(+0.57)	86.42(+0.64)	46.91(+0.49)	68.99(+0.55)	75.79(+0.56)
	F1	59.20(+0.48)	81.24(+0.61)	32.56(+0.42)	67.01(+0.47)	75.87(+0.48)
EMGAN(ours)	ACC	**70.13**	**88.06**	**50.85**	**71.04**	**77.82**
	F1	**65.68**	**84.52**	**37.79**	**70.38**	**76.64**

Table 5
Ablation experiment of EMGAN.

TextGCN	HIG	X-shaped	MMGAN	ACC	F1
✓				77.82	71.95
✓	✓			79.13	73.58
✓		✓		78.95	72.34
✓			✓	80.42	75.15
✓	✓	✓		84.91	78.38
✓	✓	✓	✓	88.06	84.52

semantics. (2) We employ an edge enhancement approach based on heterogeneous graphs, enriching inter-node connectivity by restructuring edge structures. This procedure facilitated the acquisition of higher-order relationships within the heterogeneous graph. (3) We introduce a network model based on the Minimum Margin Graph Attention Network. This model employs an attention mechanism to comprehensively explore the structure of a heterogeneous graph at minimal cost. It aggregates feature information from distant, high-order neighbors, effectively addressing the issue of sparse features in short texts.

4.3. Ablation study

To verify the impact of the proposed X-shaped structure Edge Enhancement approach on our method, we conducted the following experiments by applying the X-shaped structure Edge Enhancement approach to the HGAT, STGCN, SHINE, ST-Text-GCN, and WC-HGCN methods. The experimental outcomes are illustrated in Table 3, and based on these results, we can observe the following performances:

The X-shaped structure edge enhancement approach can restructure the edge relationships between nodes, thereby enriching the edge connections. This approach has significantly optimized HGAT, STGCN, SHINE, ST-Text-GCN and WC-HGCN. Both STGCN and ST-Text-GCN do not take into account the heterogeneity of nodes. To address this, we consider their nodes homogeneous, although this approach may result in the loss of some feature information. However, our X-shaped structure edge enhancement method can establish rich edge relationships, preserving core node features and their interrelationships, such as entities, topics, keywords, and other essential characteristics. However, the performance improvement on SHINE was not significant. Our analysis suggests that this is due to the use of hierarchical graph construction in SHINE, where nodes of the same class are present in each layer and have already formed close relationships. Our proposed method for edge enhancement shows significant improvements in HGAT and WC-HGCN, especially with HGAT-X achieving a classification accuracy of 83.11% on Snippets. The experimental results demonstrate that our X-shaped structure edge enhancement method effectively addresses the issue of sparse edge relationships in a short text, significantly improving model performance and validating the effectiveness of our approach.

To demonstrate the effectiveness of our proposed minimum margin graph attention network, we compared our EMGAN model with four variant models, namely HGAT, STGCN, STHCN and ST-Text-GCN. The comparison results are presented in Table 4.

In the model for the short text classification task, we designed a minimum margin graph attention network to achieve the purpose of enriching feature information. This model was used for the first time in short text tasks, and the model improves well in HGAT, STGCN, STHCN, and ST-Text-GCN. Firstly, Our model excels in STGCN, STHCN, and ST-Text-GCN, mainly because ST-Text-GCN builds a text graph based on word co-occurrence and document-word relationships. However, this graph only includes word and document nodes, limiting the available information. In the STGCN method, a topic model extracts the short text graph of topic words. The node information in this graph only has topic information, the node type is missing, and the word node representation of the short text plays a vital role. STHCN employs dual channel hypergraph learning to extract two distinct representations of short-text features. Subsequently, we enhance short-text embeddings by utilizing our minimal margin attention network. This integration with our proposed model allows for more effective exploration within the graph, facilitating the capture of additional node information and the enrichment of feature data. Furthermore, HGAT itself incorporates an attention mechanism, resulting in no significant performance improvements. In summary, our proposed minimum margin graph attention network can thoroughly explore the structure of heterogeneous graphs at minimal cost and aggregate feature information from distant neighbors.

The above two ablation experiments demonstrate that EMGAN not only incorporates the idea of X-shaped structure edge enhancement but also proposes a minimum margin graph attention network, which further enriches the feature information of short texts and effectively addresses the problem of sparse feature information in short texts, thus improving classification performance. However, we noticed that the F1 value is still relatively low on the Ohsumed dataset, which may be because the original Ohsumed dataset may contain multiple labels for each data, and the text information is complex. Nevertheless, our method still has room for improvement, indicating that EMGAN significantly outperforms all variants.

With respect to the effects of the three mechanisms involved by EMGAN, i.e., the heterogeneous information graph, X-shaped structure enhancement, and minimum margin graph attention network, and some combinations of them, we take TextGCN as the baseline of performance, and the experimental results are shown in Table 5. To facilitate a systematic comparison, we enumerate the results one by one. The table reveals that all mechanisms contribute to the enhancement of TextGCN, with the EMGAN fusion mechanism showing the most pronounced effect, boosting the classification accuracy from 77.82% to 88.06%. The individual application of each mechanism on TextGCN results in respective improvements of 1.31%, 1.13%, and 2.60%. We discover that using the X-shaped enhancement in isolation yields minimal improvements. This can be attributed to TextGCN graph construction being based on word co-occurrence. While we have introduced the X-shaped enhancement in TextGCN to strengthen edge structures, the information type in its graph construction remains singular. This has a certain impact, albeit relatively minor. However, when combined with the other two mechanisms, it can significantly

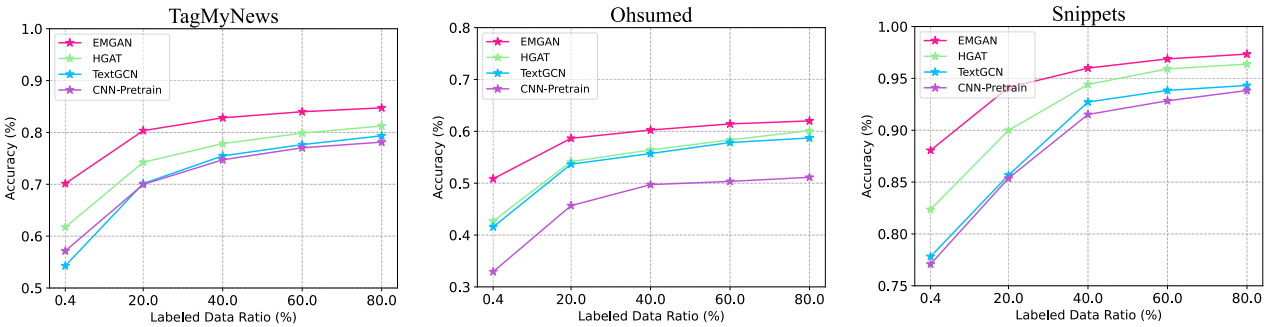
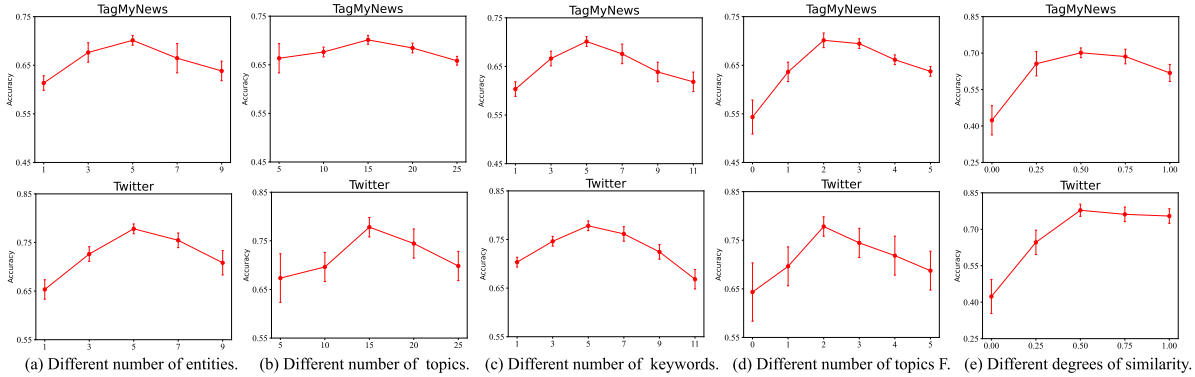


Fig. 6. The test accuracy with different numbers of labeled documents.



(a) Different number of entities. (b) Different number of topics. (c) Different number of keywords. (d) Different number of topics F. (e) Different degrees of similarity.

Fig. 7. The average accuracy with different numbers of entities, keywords, topics, top F relevant topics and similarity threshold δ between entities on TagMyNews and Twitter datasets.

enhance performance. Subsequently, we incrementally introduce mechanisms on an individual basis, achieving 84.91% effectiveness when simultaneously employing HIG and the X-shaped method. This is because HIG extracts three types of feature information, enriching the information on the graph. Subsequently, the X-shaped method enhances the graph structure by establishing higher-order connections. The EMGAN, utilizing all three mechanisms simultaneously, demonstrates the best performance. This underscores the effectiveness of MMGAN, built upon the foundations of HIG and the X-shaped structure. By employing attention to minimize edge distances on high-order heterogeneous graphs, it comprehensively explores their structure. Furthermore, it aggregates feature information from distant high-order neighbors, effectively addressing the issue of sparse features in short texts.

4.4. Labeled data

In order to evaluate the impact of labeled data size, we selected four relevant algorithms for testing, including CNN-Pretrain, TextGCN, HGAT, and EMGAN. We systematically manipulated the proportion of annotated documents across different datasets. We assessed their respective test accuracies on the TagMyNews, Snippets, MR, Ohsumed, and Twitter datasets. Each method was executed ten times, and the average performance was computed to yield the results. As shown in Fig. 6, all algorithms performed well on these datasets, with accuracy increasing as the proportion of labeled data increased. TextGCN, HGAT, and EMGAN based on graph convolutional networks achieved accuracy. This indicates that methods based on graph convolutional networks can effectively enhance information propagation through X-shaped structure edge augmentation and the minimum margin graph attention network model, enabling better utilization of limited labeled data. When the proportion of labeled documents provided is relatively small, the performance of baseline methods decreases significantly. In contrast, our method still achieves relatively high accuracy. There is a noticeable improvement when the proportion of labeled documents is relatively large. This is attributed to our EMGAN method, which connects more nodes to obtain more node feature information, effectively propagating the labeled data and maximizing its utilization to achieve accurate short text classification performance.

4.5. Parameter analysis

This section examines the parameter impact on our method through analysis. Selecting topics, entities, and keywords is crucial for our composition method, as it determines semantic capture and algorithm runtime. To verify our hypothesis, we experimented and visualized the results for reference. Fig. 7 shows the test accuracy on the TagMyNews and Twitter datasets for different numbers of topics, top-related topics, entities, and keywords. For the number of topics as the number of topics increases, the accuracy also improves. However, this trend continues until 15, after which the accuracy decreases as the number of topics increases. The top F related topics assigned to a specified document work best when $F = 2$ and show a downward trend when F exceeds 2. We have also experimented with the performance of different numbers of entities and keywords. We have noticed that as the number of selected entities and keywords grows, the testing accuracy initially improves. However, once the count exceeds 5, the accuracy starts to decline. We hypothesize that this may be because the number of entities in the document is inherently much smaller than the number of topics and keywords, and selecting too many keywords can increase the complexity of the heterogeneous information network. This may cause redundant edge relationships between unrelated nodes, making model classification more challenging. We set these four parameters in our experiments based on each dataset's validation set. For the three hyperparameters within our model: the sampling rate for margins μ, the depth of margins C, and the number of iterations $Iter$, we vary each to analyze our model's sensitivity to these factors. As shown in Fig. 8, all these outcomes are derived from the Snippets dataset. Regarding

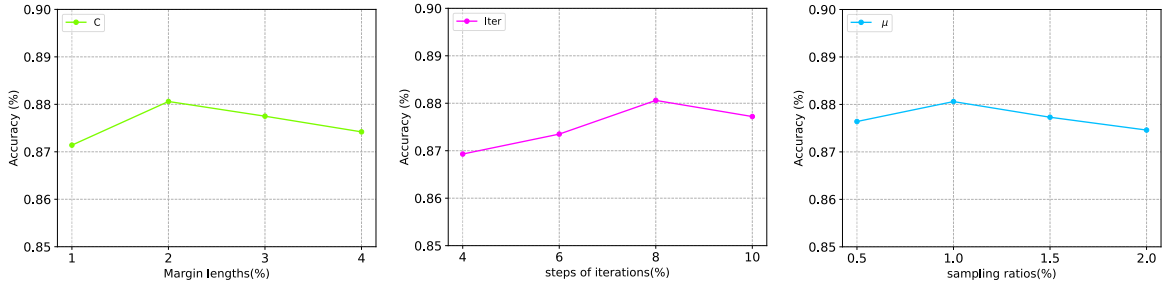


Fig. 8. In the context of the MMGAN model, a sensitivity analysis is conducted regarding the three hyperparameters: margin length C, number of iterations $Iter$, and sampling ratio μ.

the sampling rate, we held the Iter constant at 8 and C at 2, achieving optimal performance at 1.0. This signifies that we employed the same number of paths as the degree of each node. Subsequently, we fixed the sampling ratio at 1.0 and varied the number of iterations. The outcome reveals that achieving satisfactory performance requires only eight iterations. Furthermore, we have adjusted the maximum distance of the path from 2 to 5, resulting in an optimal performance of 2.

4.6. Computing complexity

Many real-world sparse graphs can be represented with values and indices within an $O(B)$ space complexity, where B represents the number of edges in the graph. The computed edge distance matrix R is truncated by C and further simplified through sampling. In the experiment, the spatial complexity of the attention layer is approximately 2 to 3 times that of a first-order attention layer. Taking the Snippets dataset as an example, the GAT model requires about 800 megabytes of GPU memory, whereas our EMGAN model only incurs approximately 300 megabytes of GPU memory. Regarding running time, the running time complexity of the shortest path algorithm Dijkstra we adopted is $O(V'BlogB)$, where V' is the number of nodes in the graph. According to the index and value of R, the sparse operator (Fey, Lenssen, Weichert, & Müller, 2018) is used to implement the path attention mechanism, making full use of the computing power of the GPU. On the RTX3090 GPU, the runtime for epochs with margin attention on the Snippets dataset is 0.3 s.

5. Conclusion

This paper proposes a novel Edge-Enhanced Minimum-Margin Graph Attention Network (EMGAN) for short text classification. This method optimizes the global topological structure to capture high-order feature information accurately. Specifically, we introduce a novel heterogeneous information graph (HIG) methodology to address the limitations of external knowledge by extracting themes, entities, and keywords as feature extensions. Subsequently, we incorporate an edge enhancement method based on an X-shaped structure, which reconstructs the edge structure between nodes, thus reinforcing edge relationships and obtaining a high-order heterogeneous graph with an X-shaped structure. Furthermore, we devise a Minimum-Margin Graph Attention Network (MMGAN) for short text classification. This model aggregates feature information from high-order neighbors and captures their rich relationships to mitigate the issue of sparse short text features. Extensive experimental results demonstrate the superiority of our model across various short text datasets compared to existing methods. It effectively overcomes the sparsity of short text data and the inadequacy of semantic features, yielding significant improvements in short text classification tasks.

First, we would like to highlight that the Heterogeneous Information Graph (HIG) technology integrates entities, topics, and keywords, enhancing search results in information retrieval and providing deeper insights in network analysis. Edge Enhancement enriches relationships between nodes, benefiting network analysis and financial modeling. MMGAN improves tasks like text summarization and sentiment analysis and enhances personalized recommendations in recommendation systems.

Therefore, EMGAN, combining HIG, Edge Enhancement, and MMGAN, offers a comprehensive understanding of short text content and finds applications in various domains beyond classification, including information retrieval, recommendation systems, social media analysis, and customer feedback. However, although our scheme achieves good results in short text classification, there are still areas for optimization. In future work, we plan to optimize from three perspectives: further enriching short text HIG, reducing information redundancy problems, and improving algorithm performance.

CRediT authorship contribution statement

Wei Ai: Supervision, Investigation, Writing – review & editing. **Yingying Wei:** Conceptualization, Methodology, Data curation, Writing – original draft. **Hongen Shao:** Supervision, Writing – review & editing. **Yuntao Shou:** Supervision, Investigation, Writing – review & editing. **Tao Meng:** Supervision, Investigation, Writing – review & editing. **Keqin Li:** Supervision, Investigation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors deepest gratitude goes to the anonymous reviewers and AE for their careful work and thoughtful suggestions that have helped improve this paper substantially. This work is supported by National Natural Science Foundation of China (Grant No. 69189338), Excellent Young Scholars of Hunan Province of China (Grant No. 22B0275), and Changsha Natural Science Foundation, China (Grant No. kq2202294).

References

Ai, W., Wang, Z., Shao, H., Meng, T., & Li, K. (2023). A multi-semantic passing framework for semi-supervised long text classification. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 1–17.

Balomenos, T., Raouzaiou, A., Ioannou, S., Drosopoulos, A., Karpouzis, K., & Kollias, S. (2005). Emotion analysis in man-machine interaction systems. *Machine Learning for Multimodal Interaction*, *3361*, 318–328.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., et al. (2008). Fast unfolding of community hierarchies in large networks.

Chakraborty, S., & Singh, A. (2022). Active sampling for text classification with subinstance level queries. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 36*, (6), (pp. 6150–6158).

Chen, Q., Yao, L., & Yang, J. (2016). Short text classification based on LDA topic model. In *2016 international conference on audio, language and image processing* (pp. 749–753). IEEE.

Cui, H., Wang, G., Li, Y., & Welsch, R. E. (2022). Self-training method based on GCN for semi-supervised short text classification. *Information Sciences*, *611*, 18–29.

Cui, H., Wang, C., & Yu, Y. (2023). News short text classification based on bert model and fusion model. *Highlights in Science, Engineering and Technology*, *34*, 262–268.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, *29*.

Dijkstra, E. W. (2022). A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His life, work, and legacy* (pp. 287–290).

Fey, M., Lenssen, J. E., Weichert, F., & Müller, H. (2018). Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 869–877).

Flisar, J., & Podgorelec, V. (2020). Improving short text classification using information from DBpedia ontology. *Fundamenta Informaticae*, *172*(3), 261–297.

Graves, A., & Graves, A. (2012). Long short-term memory. *Supervised Sequence Labelling with Recurrent Neural Networks*, 37–45.

Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. *Science*, *349*(6245), 261–266.

Hua, J., Sun, D., Hu, Y., Wang, J., Feng, S., & Wang, Z. (2024). Heterogeneous graph-convolution-network-based short-text classification. *Applied Sciences*, *14*(6), 2279.

Jin, L., Sun, Z., & Ma, H. (2022). Short text classification method with dual channel hypergraph convolution networks. In *2022 8th international conference on systems and informatics* (pp. 1–6). IEEE.

Joachims, T. (2005). Text categorization with support vector machines: Learning with many relevant features. In *Machine learning: ECML-98: 10th European conference on machine learning chemnitz, Germany, April 21–23, 1998 proceedings* (pp. 137–142). Springer.

Kateb, F., & Kalita, J. (2015). Classifying short text in social media: Twitter as case study. *International Journal of Computer Applications*, *111*(9), 1–12.

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Li, P., Liu, Y., Hu, Y., Zhang, Y., Hu, X., & Yu, K. (2022). A drift-sensitive distributed LSTM method for short text stream classification. *IEEE Transactions on Big Data*, *9*(1), 341–357.

Linmei, H., Yang, T., Shi, C., Ji, H., & Li, X. (2019). Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 4821–4830).

Liu, Y., Li, P., & Hu, X. (2022). Combining context-relevant features with multi-stage attention network for short text classification. *Computer Speech and Language*, *71*, Article 101268.

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101.

Lu, S.-H., Chiang, D.-A., Keh, H.-C., & Huang, H.-H. (2010). Chinese text classification by the Naïve Bayes Classifier and the associative classifier with multiple confidence threshold values. *Knowledge-Based Systems*, *23*(6), 598–604.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*(2), Article 026113.

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. arXiv preprint cs/0506075.

Pham, P., Nguyen, L. T., Pedrycz, W., & Vo, B. (2023). Deep learning, graph-based text representation and classification: a survey, perspectives and challenges. *Artificial Intelligence Review*, *56*(6), 4893–4927.

Phan, X.-H., Nguyen, L.-M., & Horiguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web* (pp. 91–100).

Ragesh, R., Sellamanickam, S., Iyer, A., Bairi, R., & Lingam, V. (2021). Hetegcn: heterogeneous graph convolutional networks for text classification. In *Proceedings of the 14th ACM international conference on web search and data mining* (pp. 860–868).

Rousseau, F., Kiagias, E., & Vazirgiannis, M. (2015). Text categorization as a graph classification problem. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 1702–1712).

Sidhu, D., Nair, R., & Abdallah, S. (1991). Finding disjoint paths in networks. In *Proceedings of the conference on communications architecture & protocols* (pp. 43–51).

Vitale, D., Ferragina, P., & Scaiella, U. (2012). Classification of short texts by deploying topical annotations. In *ECIR* (pp. 376–387). Springer.

Wang, X., Chen, R., Jia, Y., & Zhou, B. (2013). Short text classification using wikipedia concept based document representation. In *2013 international conference on information technology and applications* (pp. 471–474). IEEE.

Wang, C., Jiang, H., Chen, T., Liu, J., Wang, M., Jiang, S., et al. (2022). Entity understanding with hierarchical graph learning for enhanced text classification. *Knowledge-Based Systems*, *244*, Article 108576.

Wang, Z., Liu, X., Yang, P., Liu, S., & Wang, Z. (2021). Cross-lingual text classification with heterogeneous graph neural network. arXiv preprint arXiv:2105.11246.

Wang, C., Song, Y., Li, H., Zhang, M., & Han, J. (2016). Text classification with heterogeneous information network kernels. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 30*, (1).

Wang, Y., Wang, S., Yao, Q., & Dou, D. (2021). Hierarchical heterogeneous graph representation learning for short text classification. arXiv preprint arXiv:2111.00180.

Wang, Y., Wang, H., Zhang, X., Chaspari, T., Choe, Y., & Lu, M. (2019). An attention-aware bidirectional multi-residual recurrent neural network (abmrnn): A study about better short-term text classification. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing* (pp. 3582–3586). IEEE.

Wang, J., Wang, Z., Zhang, D., & Yan, J. (2017). Combining knowledge with deep convolutional neural networks for short text classification.. In *IJCAI*: *Vol. 350*, (pp. 3172077–3172295).

Wu, M. (2023). Commonsense knowledge powered heterogeneous graph attention networks for semi-supervised short text classification. *Expert Systems with Applications*, *232*, Article 120800.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

Xia, S., Peng, D., Meng, D., Zhang, C., Wang, G., Giem, E., et al. (2020). A fast adaptive k-means with no bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xia, S., Wang, G., Chen, Z., Duan, Y., et al. (2018). Complete random forest based class noise filtering learning for improving the generalizability of classifiers. *IEEE Transactions on Knowledge and Data Engineering*, *31*(11), 2063–2078.

Yang, T., Hu, L., Shi, C., Ji, H., Li, X., & Nie, L. (2021). HGAT: Heterogeneous graph attention networks for semi-supervised short text classification. *ACM Transactions on Information Systems (TOIS)*, *39*(3), 1–29.

Yang, S., Liu, Y., Zhang, Y., & Zhu, J. (2023). A word-concept heterogeneous graph convolutional network for short text classification. *Neural Processing Letters*, *55*(1), 735–750.

Yao, D., Bi, J., Huang, J., & Zhu, J. (2015). A word distributed representation based framework for large-scale short text classification. In *2015 international joint conference on neural networks* (pp. 1–7). IEEE.

Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 33*, (01), (pp. 7370–7377).

Ye, Z., Jiang, G., Liu, Y., Li, Z., & Yuan, J. (2020). Document and word representations generated by graph convolutional network and bert for short text classification. In *ECAI 2020* (pp. 2275–2281). IOS Press.

Yu, H.-F., Ho, C.-H., Arunachalam, P., Somaiya, M., & Lin, C.-J. (2012). Product title classification versus text classification. *Csie. Ntu. Edu. Tw*, 1–25.

Zhang, B., He, Q., & Zhang, D. (2022). Heterogeneous graph neural network for short text classification. *Applied Sciences*, *12*(17), 8711.

Zhang, W., Yoshida, T., & Tang, X. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, *21*(8), 879–886.

Zhou, Y., Li, J., Chi, J., Tang, W., & Zheng, Y. (2022). Set-CNN: A text convolutional neural network based on semantic extension for short text classification. *Knowledge-Based Systems*, *257*, Article 109948.

Zhou, Y., Xu, B., Xu, J., Yang, L., & Li, C. (2016). Compositional recurrent neural networks for chinese short text classification. In *2016 IEEE/WIC/aCM international conference on web intelligence* (pp. 137–144). IEEE.