HGAT: Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification

TIANCHI YANG, LINMEI HU, CHUAN SHI*, and HOUYE JI, Beijing University of Posts and Telecommunications, China
XIAOLI LI, Institute for Infocomm Research, Singapore
LIQIANG NIE, Shan Dong University, China

Short text classification has been widely explored in news tagging to provide more efficient search strategies and more effective search results for information retrieval. However, most existing studies, concentrating on long text classification, deliver unsatisfactory performance on short texts due to the sparsity issue and the insufficiency of labeled data. In this article, we propose a novel heterogeneous graph neural network-based method for semi-supervised short text classification, leveraging full advantage of limited labeled data and large unlabeled data through information propagation along the graph. Specifically, we first present a flexible heterogeneous information network (HIN) framework for modeling short texts, which can integrate any type of additional information and meanwhile capture their relations to address the semantic sparsity. Then, we propose Heterogeneous Graph Attention networks (HGAT) to embed the HIN for short text classification based on a dual-level attention mechanism, including node-level and type-level attentions. To efficiently classify new coming texts that do not previously exist in the HIN, we extend our model HGAT for inductive learning, avoiding re-training the model on the evolving HIN. Extensive experiments on single-/multi-label classification demonstrates that our proposed model HGAT significantly outperforms state-of-the-art methods across the benchmark datasets under both transductive and inductive learning.

CCS Concepts: • **Theory of computation** → **Semi-supervised learning**; • **Information systems** → Document representation; • **Computer systems organization** → *Neural networks*; • **Computing methodologies** → **Natural language processing**;

Additional Key Words and Phrases: Short texts, graph neural networks, semi-supervised learning, heterogeneous information network, inductive learning

ACM Reference format:
Tianchi Yang, Linmei Hu, Chuan Shi*, Houye Ji, Xiaoli Li, and Liqiang Nie. 2021. HGAT: Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification. *ACM Trans. Inf. Syst.* 39, 3, Article 32 (May 2021), 29 pages.
https://doi.org/10.1145/3450352

This work is supported in part by the National Natural Science Foundation of China (Grants No. U20B2045, No. 61806020, and No. 61772082) and the National Key Research and Development Program of China (Grant No. 2018YFB1402600).

Authors' addresses: T. Yang, L. Hu, C. Shi, and H. Ji, Beijing University of Posts and Telecommunications, Haidian District, Beijing, China 100876; emails: {yangtianchi, hulinmei, shichuan, jhy1993}@bupt.edu.cn; X. Li, Institute for Infocomm Research, 1 Fusionopolis Way #21-01 Connexis, Singapore 138632; email: xlli@i2r.a-star.edu.sg; L. Nie, Shandong University (Tsingtao Campus), 72 Binhai Road, Jimo Qingdao, Shandong Province, China 266237; email: nieliqiang@gmail.com.

1046-8188/2021/05-ART32 $15.00
https://doi.org/10.1145/3450352

1 INTRODUCTION

With the rapid development of online social media and e-commerce, short texts, such as online news, queries, reviews, and tweets, are increasingly widespread on the Internet [35]. Short text classification can be widely applied in many domains accordingly, including sentiment analysis, psychometric measures, news tagging/categorization and query intent classification [1, 2, 23, 56], which is one of the most important research fields in **Information Retrieval (IR)** [19, 24, 31, 50]. For example, short text classification can solve the mess of information, and it also provides more efficient search strategies and more effective search results for information retrieval [19, 24]. However, in many practical scenarios, the labeled data is scarce, while manual labeling is time-consuming and may require expert knowledge [1]. As a consequence, there is a pressing need for studying semi-supervised short text classification with a relatively small number of labeled training data.

Nevertheless, semi-supervised short text classification is nontrivial due to the following challenges. First, short texts are usually semantically sparse and ambiguous, lacking contexts [20, 26, 35, 48]. While some methods have been proposed to incorporate additional information such as entities [42, 45], they fail to consider the relational data such as the semantic relations among entities. Second, the labeled training data is limited, which leads to traditional and neural supervised methods ineffective [16, 44, 54]. As such, how to make full use of the limited labeled data and the large number of unlabeled data has become a key task for short text classification [1]. Third, we need to capture the importance of different information that is incorporated to address sparsity at multiple granularity levels and reduce the weights of noisy information to achieve more accurate classification results.

In this work, we propose a novel *heterogeneous graph neural network-based method* for semi-supervised short text classification, which makes full use of both limited labeled data and numerous unlabeled data by allowing information propagation through our automatically constructed graph. Particularly, we first present a flexible HIN framework for modeling the short texts[1] as shown in Figure 1, which is able to incorporate any additional information (e.g., entities and topics) as well as capture the rich relations among the texts and the additional information. Since the HIN for short texts is information-rich, traditional network embedding methods that only focus on the network topology will lead to severe information loss, such as GNetMine [15], node2vec [13], and metapath2vec [8]. Consequently, we propose **Heterogeneous Graph Attention networks (HGAT)** to embed the HIN for short text classification based on a new dual-level attention mechanism including node-level and type-level attentions. Our HGAT method considers the heterogeneity of different node types. Additionally, the dual-level attention mechanism captures both the importance of both different neighboring nodes (reducing the weights of noisy information) and different node (information) types to a current node. To address the new coming texts that do not previously exist in the HIN, we extend our model HGAT for inductive learning, which can avoid re-training the model on the evolving HIN and address the new coming texts efficiently. Specifically, we first construct an inductive graph for the new coming texts, which takes full advantage of the information from existing labeled and unlabeled data. Sampling strategies are also explored to reduce the time complexity. Then, we apply the trained HGAT on the newly constructed inductive graph to predict the labels for the new coming texts. Furthermore, we improve our HGAT by introducing orphan categories to match the non-text categories, thus reducing the classification interference of the entity and topic categories in the HIN. The main contributions of this article can be summarized as follows:

[1] For ease of expression, we will use the terms *document* and *short text* interchangeably.

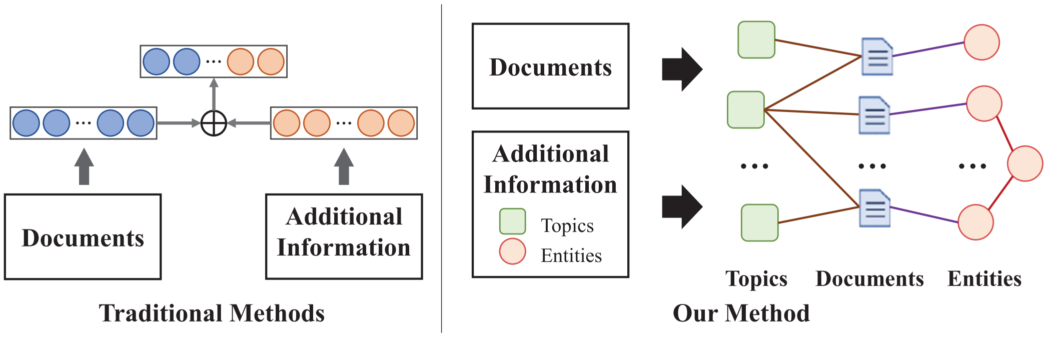


Fig. 1. Comparison between traditional and our method. Traditional methods usually directly merge the representations of the additional information and text. Our method builds an HIN for short texts to flexibly incorporate any additional information and capture the rich relations among the texts and the additional information.

(1) To the best of our knowledge, *this is the first attempt* to model short texts as well as additional information with an HIN and adapt graph neural networks on the HIN for semi-supervised classification.
(2) We propose novel HGAT for the HIN embedding based on a new dual-level attention mechanism, which can learn the importance of both different neighboring nodes and different node (information) types to a current node.
(3) We propose an inductive learning approach with sampling strategy for our HGAT to handle new coming texts efficiently, which fully leverages the information from not only the new coming texts but also the existing labeled and unlabeled texts.
(4) Extensive experimental results have demonstrated that our proposed HGAT model significantly outperforms seven state-of-the-art methods across benchmark datasets.

Please notice that the preliminary work has been accepted at the 2019 Conference on Empirical Methods in Natural Language Processing [21]. Based on the conference paper, we substantially extend the original work from the following aspects:

(1) To predict the labels of new coming texts that do not previously exist in the constructed HIN for short texts, we propose a new inductive learning approach for HGAT in Section 3.4, which avoids re-training the model on the evolving HIN and addresses the new coming texts efficiently. Besides, sampling strategies are also explored to reduce the time complexity. It fully leverages the information from both the new coming texts and the existing labeled and unlabeled texts.
(2) Considering the requirements of multi-label classification in real applications, we extend the original HGAT with new objective functions for multi-label classification in Section 3.5.
(3) We significantly enrich the experiments to demonstrate the superior performance of the proposed methods. Specifically, we have added the experiments to validate the effectiveness of the proposed inductive learning for HGAT (Sections 4.2.1, 4.2.4, and 4.2.5). We add the state-of-the-art baseline models Bert and GraphSAGE to further validate our proposed model. We further conduct experiments to test the effectiveness of our model on multi-label classification (Section 4.2.2). In addition, we add experiment to justify the effectiveness of our model under the setting where more labeled data are available (Section 4.2.6).
(4) A more comprehensive discussion of related work is provided. Besides, we carefully polish our article and improve the language quality.

The rest of the article is organized as follows. In Section 2, we discuss and summarize the related work. Section 3 describes our proposed method including the construction of HIN for short texts and HGAT model in detail as well as the proposed inductive learning approach. Extensive experiments are conducted to validate the proposed models in Section 4. Finally, we conclude the article in Section 5.

2 RELATED WORK

In this section, we first introduce the related work on short text classification including traditional methods, deep neural network methods and graph-based semi-supervised methods. Then, we discuss the recent work on graph neural networks.

2.1 Traditional Text Classification

Traditional text classification methods, such as SVM [9], require a feature engineering step for text representation. The most commonly used manual features are BoW and TF-IDF vectors [3], where each element is corresponding to the occurrence/weight of each term in the word dictionary. Besides, features of bi-grams and tri-grams are included to capture the co-occurrence information between words [33, 44]. Some recent studies [28, 41] model texts as graphs and extract path-based features for classification. Despite their initial success on formal and well-edited texts, all these methods usually fail to deliver satisfactory performance on short text classification, due to the insufficient features incurred by short texts [10]. To address this problem, efforts have been made to enrich the semantics of *short texts* [7, 43]. For example, Phan et al. [26] extracted the latent topics of the short texts with the help of an external corpus. Wang et al. [45] introduced external entity information from Knowledge Bases. Di Yao et al. [7] enriched short texts with word semantic similarity information. However, these methods cannot obtain good performance easily as the feature engineering step relies on domain knowledge.

2.2 Deep Neural Networks for Text Classification

Deep neural networks that automatically represent texts as embeddings have been widely used for text classification. RNNs [22, 34] capture the sequential information of words, while CNNs [11, 16, 32] capture the N-gram information. The above two representative deep neural models have shown their power in many NLP tasks, including text classification. To adapt them to *short text classification*, several methods have been proposed. For example, Zhang et al. [54] designs a character-level CNN that alleviates the sparsity by mining different levels of information within the texts. Wang et al. [42] incorporates the entities and concepts from KBs to enrich the semantics of short texts. Ghadery et al. [11] integrates multilingual n-gram information to enrich the text representation. Recently, BERT [6] uses a multi-layer bidirectional Transformer encoder and trains with a masked language model, leading to the state-of-the-art performance on several NLP tasks. However, these methods cannot capture the semantic relations (e.g., entity relations) and rely heavily on the large number of training data. Clearly, lacking training data is still a key bottleneck that prohibits them from successful practical applications.

2.3 Graph-based Semi-supervised Text Classification

Since the manual labeling is very expensive and the unlabeled texts also contains valuable information, a large number of semi-supervised methods have been proposed [23, 37, 52]. Here, we will only introduce the topic relevant to our task scenario, i.e., graph-based semi-supervised text classification. Generally, most graph-based methods construct an affinity graph for both labeled and unlabeled texts based on local features, such as similarity between samples, shared words or phrases, and so on. For example, Zhu et al. [57] defined the problem with a Gaussian random field model with respect to the graph. Zhou et al. [55] proposed to spread label information of each point to neighbors to ensure both local and global consistency. However, these traditional graph-based methods are based on homogeneous graphs, thus they cannot distinguish the different semantic meaning of multi-typed nodes and edges. Then, some efforts extend the graph-based learning framework for heterogeneous networked data. GNetMine [15] models the link structure in information networks with arbitrary network schema. Rossi et al. [27] constructed a bipartite heterogeneous network, containing documents and terms. Then label propagation is performed from documents to terms and then from terms to documents iteratively. While PTE [36] models the documents, words, and labels with graphs and learns text (node) embeddings for classification. Recently, **graph convolutional networks (GCN)** have received wide attention for semi-supervised classification [17]. Some researchers began to explore GCN for text classification. For example, TextGCN [51] models the whole text corpus as a document-word graph with word co-occurrence relations and applies GCN for classification. However, all these methods focus on long texts. In addition, they fail to use attention mechanisms, leaving important information missed.

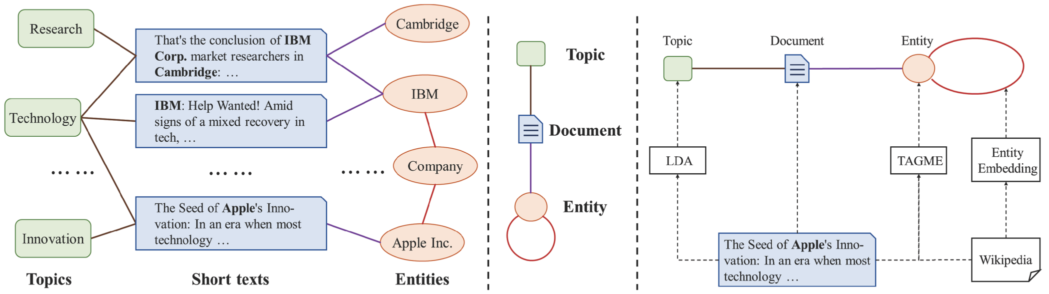
2.4 Graph Neural Networks

Graph neural networks (GNNs), first introduced by Gori et al. [12] and Scarselli et al. [30], extend the deep neural networks (especially convolutional network) from dealing with European spatial data to arbitrary non-European spatial data, i.e., graph-structured data. The graph convolutional neural networks can be generally divided into two categories, namely, spectral domain and spatial domain. The former uses the Spectral Graph Theory to implement convolution operations on graphs. Bruna et al. [4] defined convolution on general graphs through the corresponding Fourier basis. To avoid expensive computation, Defferrard et al. [5] approximated the convolution operator with K-order Chebyshev polynomials. Finally, Kipf and Welling [17] designed a GCN with only first-order approximation of spectral graph convolutions, which inspired researchers' enthusiasm for GNN. In terms of spatial approaches, convolutions are defined directly on the graph by the adjacency. Hamilton et al. [14] performed a neural network-based aggregator over fixed-size node neighbors, namely, GraphSAGE, which learns a function to generate embeddings by aggregating features from local neighborhood. Then, attention mechanisms are introduced into graph neural networks to automatically learn the importance of different neighbors, such as Graph Attention Network [39], HAN [46]. GNNs have been successfully applied in a wide range of applications ranging from social networks [17], computer vision [47], to text classification [51].

Different from the above existing studies, in this article, we propose to construct an HIN for short texts that can flexibly integrate any additional information, and develop a novel heterogeneous graph attention network model considering both the heterogeneity and importance of different nodes in the HIN for semi-supervised short text classification.

3 OUR PROPOSED METHOD

In this article, we propose a novel heterogeneous graph neural network-based method for semi-supervised short text classification, which takes full advantage of both limited labeled data and large unlabeled data by allowing information propagation along the graph. Our method includes two steps. Particularly, to alleviate the sparsity of short texts, we first present a flexible HIN framework for modeling the short texts, which can incorporate any additional information and capture the rich relations among the short texts and the added information. Then, we propose a novel model HGAT to embed the HIN for short text classification based on a new dual-level attention mechanism. HGAT considers the heterogeneity of different types of information. In addition, the attention mechanism can learn the importance of both different nodes (reducing the weights of noisy information) and different node (information) types. Furthermore, to address the new


Research
Technology
Innovation
That's the conclusion of IBM Corp. market researchers in Cambridge: …
IBM: Help Wanted! Amid signs of a mixed recovery in tech, …
The Seed of Apple's Innovation: In an era when most technology …
Cambridge
IBM
Company
Apple Inc.
…… …… ……
Topics
Short texts
Entities
Topic
Document
Entity
Topic
Document
Entity
LDA
TAGME
Entity Embedding
Wikipedia
The Seed of Apple's Innovation: In an era when most technology …

(a) HIN for short texts (b) HIN schema (c) Construction of HIN for short texts

Fig. 2. (a) An example of HIN for short texts on AGNews. (b) Network schema of the HIN for short texts. (c) Illustration of the construction of the HIN for short texts.

coming texts, we propose a new inductive learning approach for HGAT to leverage the information from both the new coming texts and the existing labeled and unlabeled texts, where sampling strategies are also explored to reduce the time complexity.

3.1 HIN for Short Texts

We first present the HIN framework for modeling the short texts, as shown in Figures 2(a) and 2(b), which enables the integration of any additional information and captures the rich relations among the texts and the added information. In this way, the sparsity of the short texts is alleviated.

Previous studies have exploited latent topics [53] and external knowledge (e.g., entities) from Knowledge Bases to enrich the semantics of the short texts [42, 45]. However, they fail to consider the semantic relation information, such as entity relations. In contrast, our HIN framework for short texts can flexibly integrate any additional information and model their rich relations.

Here, we consider two types of additional information i.e., topics and entities. As shown in Figure 2(c), we construct the HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ containing the short texts $D = \{d_1, \ldots, d_m\}$, topics $T = \{t_1, \ldots, t_K\}$, and entities $E = \{e_1, \ldots, e_n\}$ as nodes, i.e., $\mathcal{V} = D \cup T \cup E$. The set of edges $\mathcal{E}$ represents their relations. The details of constructing the network are described as follows.

First, LDA [3] is applied to mine the latent topics T for enriching the semantics of short texts. Each topic $t_i = (\theta_1, \ldots, \theta_w)$ (w denotes the vocabulary size) is represented by a probability distribution over the words. For reducing the noise in irrelevant topics, we assign each document to the top P topics with the largest probabilities. Thus, the edge is built between a document and a topic if the document is assigned to the topic.

Second, we recognize the entities E in the documents D and link them to Wikipedia with the open entity linking tool TAGME.[2] If a document contains an entity, then the edge between them will be built. We take an entity as a whole word and run word2vec[3] based on the Wikipedia corpus to learn the entity embeddings. To further enrich the semantics of short texts and advance the information propagation, we consider the relations between entities. Particularly, we first calculate the similarity (cosine similarity) between all the entities based on the entity embeddings. Then the edge between two entities will be built if their similarity score is larger than a predefined threshold δ.

By incorporating the topics, entities and the relations, we enrich the semantics of the short texts and thus greatly benefit the following classification task. For example, as shown in Figure 2, the short text "the seed of Apple's Innovation: In an era when most technology..." is semantically

[2] https://sobigdata.d4science.org/group/tagme/.
[3] https://code.google.com/archive/p/word2vec/.

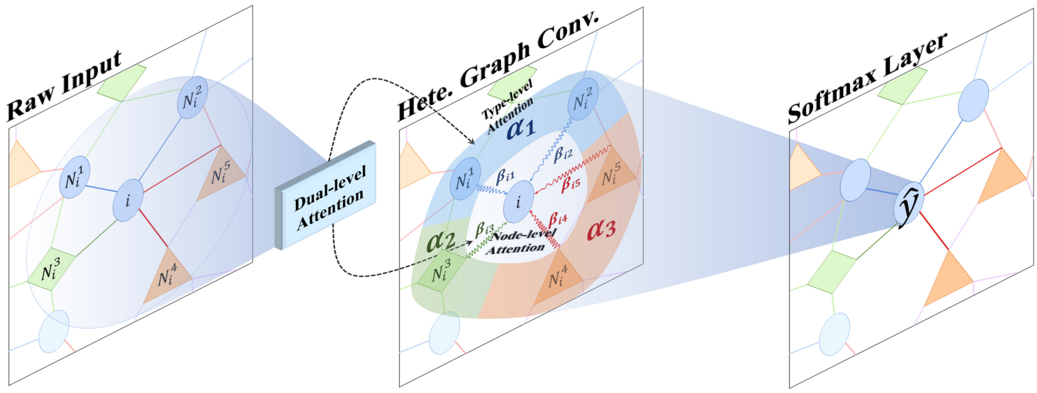


Fig. 3. Illustration of our model HGAT.

enriched by the relations with the entities "Apple Inc." and "company," as well as the topic "technology." Thus, it can be correctly classified into the category of "business" with high confidence.

3.2 HGAT

We then propose HGAT model (shown in Figure 3) to embed the HIN for short text classification based on a new dual-level attention mechanism including node level and type level. HGAT considers the heterogeneity of different types of information with heterogeneous graph convolution. In addition, the dual-level attention mechanism captures the importance of both different neighboring nodes (reducing the weights of noisy information) and different node (information) types to a specific node. Furthermore, we improve our HGAT by introducing orphan categories to match the non-text categories, thus reducing the classification interference of the entity and topic categories in the HIN. Finally, it predicts the labels of documents through a softmax/sigmoid layer.

To predict the labels of new coming texts that do not previously exist in the constructed HIN for short texts, we propose a new inductive learning approach for HGAT, which avoids re-training the model on the evolving HIN and addresses the new coming texts efficiently. It fully leverages the information from both the new coming texts and the existing labeled and unlabeled texts.

3.2.1 Heterogeneous Graph Convolution. We first describe the heterogeneous graph convolution in HGAT, considering the heterogeneous types of nodes (information).

As known, GCN [17] is a multi-layer graph neural network that operates directly on a homogeneous graph and induces the hidden embeddings of nodes based on the relationships of their neighborhoods. Formally, consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ represent the set of nodes and edges, respectively. Let $X \in \mathbb{R}^{|\mathcal{V}| \times q}$ be a matrix containing the nodes with their features $x_v \in \mathbb{R}^q$ (each row x_v is a feature vector for a node v). For the graph $\mathcal{G}$, we introduce its adjacency matrix $A' = A + I$ with self-connections added and its corresponding degree matrix M, where $M_{ii} = \sum_j A'_{ij}$. Then the layer-wise propagation rule can be summaried as follows:

$$H^{(l+1)} = \sigma(\tilde{A} \cdot H^{(l)} \cdot W^{(l)}), \tag{1}$$

where $\tilde{A} = M^{-\frac{1}{2}} A' M^{-\frac{1}{2}}$ denotes the symmetric normalized adjacency matrix. $W^{(l)}$ is a layer-specific trainable transformation matrix. $\sigma(\cdot)$ denotes an activation function such as ReLU. $H^{(l)} \in \mathbb{R}^{|\mathcal{V}| \times q}$ denotes the hidden representations of nodes in the l^{th} layer. Initially, $H^{(0)} = X$.

Nevertheless, GCN cannot be directly applied to the HIN for short texts due to the node heterogeneity issue. Specifically, in the HIN, we have three types of nodes: documents, topics and entities with different feature spaces. For a document $d \in D$, we take the TF-IDF vector as its feature vector x_d. For a topic $t \in T$, we take its the word distribution over the vocabulary to represent the topic $x_t = \{\theta_i\}_{i=[1,w]}$. For each entity, to make full use of relevant information, we represent the entity x_v by concatenating its embedding and TF-IDF vector of its Wikipedia description text.

A straightforward way to adapt GCN for the HIN containing different types of nodes $\mathcal{T} = \{\tau_1, \tau_2, \tau_3\}$ is to construct a new large feature space by concatenating the feature spaces of different types of nodes. For example, each node is denoted as a feature vector with 0 values for the irrelevant dimensions for other types, i.e., the feature spaces of the other types of nodes. We name this basic method of adapting GCN to HIN as *GCN-HIN*. However, it suffers from reduced performance, since it ignores the heterogeneity of different information types.

To address the issue, we propose the heterogeneous graph convolution, which considers the difference of various types of information and projects them into an implicit common space with their respective transformation matrices as

$$H^{(l+1)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \tilde{A}_\tau \cdot H_\tau^{(l)} \cdot W_\tau^{(l)}\right), \tag{2}$$

where $\tilde{A}_\tau \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}_\tau|}$ is the submatrix of $\tilde{A}$, whose rows represent all the nodes and columns represent their neighboring nodes with the type τ. The representation of the nodes $H^{(l+1)}$ is obtained by aggregating information from the features of their neighboring nodes $H_\tau^{(l)}$ with different types τ via different transformation matrix $W_\tau^{(l)} \in \mathbb{R}^{q^{(l)} \times q^{(l+1)}}$. The transformation matrix $W_\tau^{(l)}$ considers the difference of feature spaces and projects them into an implicit common space $\mathbb{R}^{q^{(l+1)}}$. Initially, $H_\tau^{(0)} = X_\tau$.

3.2.2 Dual-level Attention Mechanism. Typically, given a specific node, different types of neighboring nodes may have different impacts on it. For example, the neighboring nodes of the same type as the current node may carry more useful information. Additionally, different neighboring nodes of the same type could also have different importance. To capture the different importance at both node level and type level, we design a new dual-level attention mechanism as shown in Figure 3.

Type-level Attention. Given a specific node v, the type-level attention learns the weights of different types of neighboring nodes. Specifically, we first represent the embedding of the type τ as $h_\tau = \sum_{v'} \tilde{A}_{vv'} h_{v'}$, which is the sum of the neighboring node features $h_{v'}$ where the nodes $v' \in \mathcal{N}_v$ and are with the type τ. Then, we calculate the type-level attention scores based on the current node embedding h_v and the type embedding h_τ:

$$a_\tau = \sigma(\mu_\tau^T \cdot [h_v || h_\tau]), \tag{3}$$

where μ_τ is the attention vector for the type τ, $||$ means "concatenate," and $\sigma(\cdot)$ denotes the activation function, such as Leaky ReLU.

Then, we normalize the attention scores across all the types with the softmax function and obtain the type-level attention weights as

$$\alpha_\tau = \frac{\exp(a_\tau)}{\sum_{\tau' \in \mathcal{T}} \exp(a_{\tau'})}. \tag{4}$$

Node-level Attention. Node-level attention is designed to capture the importance of different neighboring nodes and reduce the weights of noisy nodes. Formally, given a specific node v with the type τ and its neighboring node $v' \in \mathcal{N}_v$ with the type τ', we compute the node-level attention scores based on the node embeddings h_v and $h_{v'}$ with the type-level attention weight $\alpha_{\tau'}$ for the node v':

$$b_{vv'} = \sigma(\nu^T \cdot \alpha_{\tau'}[h_v || h_{v'}]), \tag{5}$$

where ν is the attention vector. Then, we obtain the node-level attention weights by normalizing the node-level attention scores with the softmax function as

$$\beta_{vv'} = \frac{\exp(b_{vv'})}{\sum_{i \in \mathcal{N}_v} \exp(b_{vi})}. \tag{6}$$

Afterward, we incorporate the dual-level attention mechanism including type-level and node-level attentions into the heterogeneous graph convolution by replacing Equation (2) with the following layer-wise propagation rule:

$$H^{(l+1)} = \sigma\left(\sum_{\tau \in \mathcal{T}} \mathcal{B}_\tau \cdot H_\tau^{(l)} \cdot W_\tau^{(l)}\right), \tag{7}$$

where $\mathcal{B}_\tau$ represents the attention matrix, whose element in the v^{th} row v'^{th} column is $\beta_{vv'}$ in Equation (6).

3.2.3 Alleviation of Overfitting. Due to lack of information in semi-supervised short text classification, the attention mechanism is prone to overfitting. Considering that the prior knowledge of the node importance can guide the attention mechanism, we make a trade-off for each node type τ between the original graph convolution matrix $\tilde{A}_\tau$ in GCN and the dual-level attention matrix $\mathcal{B}_\tau$ with a hyper-parameter λ to alleviate overfitting. Formally,

$$f(\tilde{A}_\tau, \mathcal{B}_\tau; \lambda) = (1 - \lambda) \cdot \tilde{A}_\tau + \lambda \cdot \text{diag}(\tilde{A}_\tau \cdot \mathbf{1}) \cdot \mathcal{B}_\tau, \tag{8}$$

where $\mathbf{1}$ represents the vector with all elements equal to 1. Note that the sum of each row of the matrix $\mathcal{B}$ equals to 1 because of the softmax normalization, while the sum of each row of the matrix $\tilde{A}$ does not because of the symmetric normalization. Therefore, we use the the term $\text{diag}(\tilde{A}_\tau \cdot \mathbf{1})$ for better fusion.

Finally, the layer-wise propagation rule of our proposed heterogeneous graph convolution can be summarized as follows:

$$H^{(l+1)} = \sigma\left(\sum_{\tau \in \mathcal{T}} f(\tilde{A}_\tau, \mathcal{B}_\tau; \lambda) \cdot H_\tau^{(l)} \cdot W_\tau^{(l)}\right). \tag{9}$$

3.3 Orphan Categories

In our original HGAT model [21], the output of the model corresponds to the probabilities of the short text belonging to each text category. Inspired by Sabour et al. [29] and Yang et al. [49], an additional "orphan" category can capture the "background" information of the images or "stop words" of the words that are unrelated to specific categories, helping improve the classification accuracy. Therefore, to further improve our model performance for short text classification, we introduce two "orphan" categories to match the non-text categories in the HIN including "entity" and "topic." They can be seen as the "background" information of the HIN for short texts, thus helping HGAT better embed the HIN and reducing the classification interference caused by the non-text categories in the HIN. For example, as shown in Figure 2, the short text "the seed of Apple's Innovation: In an era when most technology..." is semantically enriched by an entity "Apple Inc." If there is no orphan category for entity, then the model may attempt to classify this entity node "Apple Inc." into the category of "business" that is the same category as the neighboring text. However, "Apple Inc." is also connected to the short text "iPod Rivals Square Off Against Apple (Reuters). The next wave of iPod competitors is coming," whose category is "entertainment." This phenomenon increases the difficulty of fitting the data. If there is an orphan category for entity, then the model may attempt to classify "Apple Inc." into this orphan category, namely, "entity."

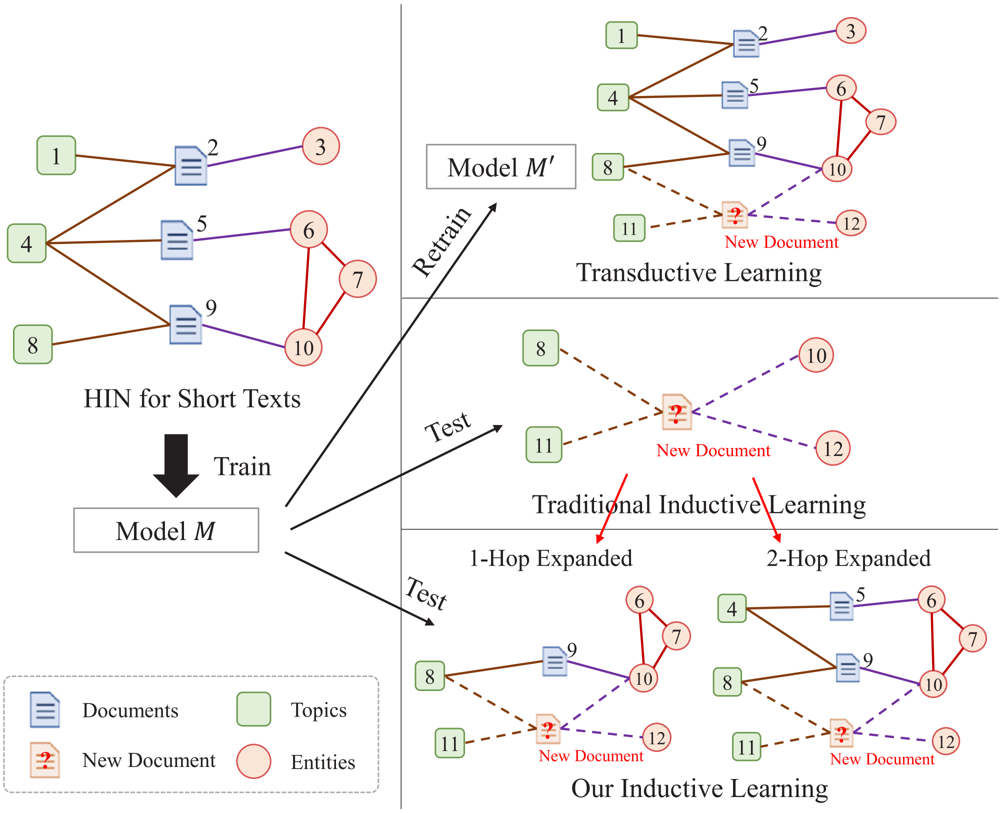


Fig. 4. Comparison among transductive learning, traditional inductive learning, and our inductive learning.

This forces the entity "Apple Inc." to only provide its carried information without disturbing the propagation of the label on the graph. It could reduce the impact of this confusion about categories. Consequently, orphan categories providing placeholders for entity and topic nodes can effectively help HGAT better embed the HIN. Formally, the dimension of the output probabilities of the model increases by 2, indicating "entity" and "topic."

3.4 Inductive Learning

In real applications, massive text data is generated every day. As shown in Figure 4, when a new coming short text is to be classified, for our original HGAT model [21], we need to construct a new HIN involving the new text and retrain the model to predict the label for the new text in a transductive learning way. This is impractical for real applications due to time efficiency. Therefore, we extend our model HGAT for inductive learning, which can address the new coming texts efficiently. As shown in Figure 4, traditional inductive methods deal with new coming texts by applying the trained model on the graph constructed only for the new coming texts. It significantly improves efficiency at the expense of a small performance drop.

In this article, to take full advantage of not only the new coming texts but also the existing labeled and unlabeled data, we propose a new inductive learning approach for our model HGAT to better deal with the lack of information in semi-supervised short text classification. Specifically, we first construct an inductive graph for the new coming texts, which is expanded by the existing labeled and unlabeled data. Formally, given an HIN for the existing short texts $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and a set of new coming texts D_{new}, we fist construct a new HIN $\mathcal{G}_{new} = \{\mathcal{V}_{new}, \mathcal{E}_{new}\}$ for the new coming texts D_{new} following the process described in Section 3.1[4]. Afterward, as shown in Figure 4, we can construct one-hop or two-hop expanded inductive graph $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$ by expanding the new text graph $\mathcal{G}_{new}$ with neighbors from existing HIN $\mathcal{G}$ within one hop or two hops, respectively.

Moreover, the number of edges might be billion level in real-world applications, such as Twitter, FaceBook, and so on. In practice, it is not necessary to aggregate all the neighboring nodes to obtain an optimal performance, while a sub-optimal performance is absolutely acceptable if the time consuming could be greatly reduced. Therefore, it is important to apply an efficient neighborhood sampling strategy. Note that the time complexity without sampling strategy is $O(\#New \cdot \#Degree^{\#Hop})$, while the time complexity with sampling strategy is $O(\#New \cdot \#Sample^{\#Hop})$, where $\#New$, $\#Degree$, $\#Hop$, and $\#Sample$ represent the number of new coming texts, average node degrees, hops and number of sampling neighbors, respectively. A simple way is to utilize uniform random sampling over the whole neighbors, but some nodes of low-relevance may bring noise and harm the performance. Another straight forward solution is to directly choose some TOP relevant neighboring nodes, where the relevance can be measured by the attention mechanism. However, high relevance usually means the information of each other is highly overlapped, thus leading to limited access to complementary additional information. Therefore, weighted random sampling is more appropriate, because it make a trade-off between the above, which not only ensures that some complementary information can be introduced but also ensures that the noise is not introduced too much. We can utilize the dual-level attention mechanism to calculate the weights of each neighboring node. We will verify these analysis in Section 4.2.5.

The detailed expansion process is illustrated in Algorithm 1. Finally, we apply the trained HGAT model on the inductive graph to predict the labels of new coming texts.

3.5 Model Training

After going through an L-layer HGAT, we can get the embeddings of nodes (including short text embeddings $H^{(L)}$) in the HIN. For single-label classification, the node embeddings are then fed to a softmax layer for classification, while for multi-label classification, the node embeddings are fed to a sigmoid layer. Formally,

$$Z_{\text{single}} = \text{softmax}(H^{(L)}), \tag{10}$$

$$Z_{\text{multi}} = \text{sigmoid}(H^{(L)}). \tag{11}$$

During model training, we exploit the cross-entropy loss over training data with the L2-norm for single-label classification while we use a separate margin loss [29, 49] for multi-label classification. The margin loss allows independent training of each category and ensures the training does not focus too much on the samples that have been correctly predicted with high confidence, thereby alleviating overfitting. Formally,

$$\mathcal{L}_{\text{single}} = -\sum_{i \in D_{\text{train}}} \sum_{j=1}^{|C|} Y_{ij} \cdot \log Z_{ij} + \eta \, \|\Theta\|_2, \tag{12}$$

$$\mathcal{L}_{\text{multi}} = \sum_{i \in D_{\text{train}}} \sum_{j=1}^{|C|} (Y_{ij} \max(0, m^+ - Z_{ij})^2 + (1 - Y_{ij}) \max(0, Z_{ij} - m^-)^2) + \eta \, \|\Theta\|_2, \tag{13}$$

[4] Note that for the construction of topic nodes in new graph $\mathcal{G}_{new}$, we use the LDA model pre-trained on the training data to ensure that the original and the new topic feature spaces are consistent.

ALGORITHM 1: Construction of the Expanded Inductive Graph

```
Input: New coming texts $D_{new}$, HIN for existing short texts $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, hops $H$
Output: $H$-hop expanded inductive graph $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$
Construct a new HIN $\mathcal{G}_{new} = \{\mathcal{V}_{new}, \mathcal{E}_{new}\}$ for $D_{new}$ following the process described in Section 3.1
Cross nodes $\mathcal{N}_{\text{cross}} \leftarrow \mathcal{V} \cap \mathcal{V}_{new}$
Queue $q$, Involved node set $v$
$q$.push($\mathcal{N}_{\text{cross}}$), $v$.add($\mathcal{N}_{\text{cross}}$)
for $i = 1$ to $H$ do
    Queue $p$
    while not $q$.empty() do
        $n_1 \leftarrow q$.pop()
        for $(n_1, n_2) \in \mathcal{E}$ do
            if $n_2 \notin \mathcal{V}$ and $n_2$ is sampled then
                $\mathcal{V}$.add($n_2$), $p$.push($n_2$)
            end if
        end for
    end while
    $q \leftarrow p$
end for
Involved Edge Set
        $e \leftarrow \{(n_1, n_2) | (n_1, n_2) \in \mathcal{E}$ and $n_1, n_2 \in \mathcal{V}\}$
$\mathcal{G}' \leftarrow \{\mathcal{V}_{new} \cup v, \mathcal{E}_{new} \cup e\}$
return $\mathcal{G}'$
```

where D_{train} is the set of short text indices for training, Y is the corresponding label indicator matrix, Θ is model parameters, and η is regularization factor. $m^+ = 0.9$ and $m^- = 0.1$ discourage classifiers from overconfidence [49]. For model optimization, we adopt the gradient descent algorithm.

4 EXPERIMENTS

In this section, we evaluate the empirical performance of different methods for semi-supervised short text classification.

4.1 Experimental Setup

4.1.1 Datasets. For single-label classification, we conducted extensive experiments on 6 short text benchmark datasets: AGNews, Snippets, Ohsumed, TagMyNews, MR, and Twitter.

AGNews: This dataset is adopted from Zhang et al. [54]. We randomly selected 6,000 pieces of news from AGNews, evenly distributed into four categories.

Snippets: This dataset is released by Phan et al. [26]. It is composed of the snippets returned by a web-search engine.

Ohsumed: We used the benchmark bibliographic classification dataset released by Yao et al. [51], where the documents with multiple labels are removed. In this work, we used the titles for short text classification.

TagMyNews: We applied the news titles as instances from the benchmark classification dataset released by Vitale et al. [40], which contains English news from **really simple syndication (RSS)** feeds of three newspapers. They are annotated with seven categories.

Table 1. Statistics of the Datasets

	#docs	#tokens	#entities	docs with entities (%)	#categories
AGNews	6,000	18.4	0.9	72%	4
Snippets	12,340	14.5	4.4	94%	8
Ohsumed	7,400	6.8	3.1	96%	23
TagMyNews	32,549	5.1	1.9	86%	7
MR	10,662	7.6	1.8	76%	2
Twitter	10,000	3.5	1.1	63%	2
Ohsumed-multi	13,929	7.2	1.4	100%	23

MR: It is a movie review dataset, in which each review only contains one sentence [25]. Each sentence is annotated with positive or negative for binary sentiment classification. This corpus contains 5,331 positive and 5,331 negative movie reviews.

Twitter: This dataset is provided by NLTK,[5] a library of Python, which is also a binary sentiment classification dataset. It has 5,000 positive tweets and 5,000 negative tweets.

Ohsumed-multi[6]: We used the whole standard dataset of Ohsumed for multi-label classification. It is a mixture of 7,400 single-label samples and 6,529 multi-label samples. We only adopted the titles for short text classification to categorize the 23 cardiovascular disease categories.

For each dataset, we randomly selected 40 labeled documents per class, half for training and the other half for validation. For transductive learning, following Kipf and Welling [17], all the left documents are taken as unlabeled documents in the HIN for model training. We tested the models on these unlabeled documents. For inductive learning, except for the labeled documents, we randomly selected 1,000 unlabeled documents that are also included in the HIN for model training, and the left are taken as new coming texts. We tested the models on these new coming texts.

We preprocessed all the datasets as follows. Specifically, we removed non-English characters, stop words, and low-frequency words appearing less than five times. Table 1 shows the statistics of the datasets, including the number of documents, the number of average tokens and entities, the number of categories, and the proportion of documents containing entities in parentheses. In our datasets, most of the texts (around 80%) contain entities.

4.1.2 Evaluation Metrics. We applied the following standard metrics to evaluate the performance of all the methods.

For single-label classification, we use the well-known two metrics: Accuracy and F1-score (Macro-F1).

- **Accuracy** is simply a ratio of correctly predicted observation to the total observations.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}. \tag{14}$$

- **F1-score** is the harmonic mean of Precision and Recall.

$$\text{F1-score} = \frac{1}{|C|}\sum_{c\in C}\frac{2P_cR_c}{P_c+R_c}, \tag{15}$$

[5] https://www.nltk.org/.
[6] http://disi.unitn.it/moschitti/corpora.html.

$$\text{where } P_c = \frac{TP_c}{TP_c + FP_c}, \quad R_c = \frac{TP_c}{TP_c + FN_c}, \tag{16}$$

where TP_c, TN_c, FP_c, FN_c denote the true positives, true negatives, false positives and false negatives for the category c in label set C, respectively.

For multi-label classification, following Yang et al. [49], we adopted **Exact Match Ratio (ER)**, Micro-Precision, Micro-Recall, Micro-F1, Macro-Precision, Macro-Recall and Macro-F1 as the evaluation metrics.

- **ER** considers partially correct prediction as incorrect and only counts fully correct samples:

$$\text{ER} = \frac{\text{exactly correct instances}}{\text{total instances}}. \tag{17}$$

- **Micros** calculate metrics globally by counting the total true positives, false negatives, and false positives:

$$\text{Micro-Precision} = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C}(TP_c + FP_c)}, \tag{18}$$

$$\text{Micro-Recall} = \frac{\sum_{c \in C} TP_c}{\sum_{c \in C}(TP_c + FN_c)}, \tag{19}$$

$$\text{Micro-F1} = \frac{2 * \text{Micro-Precision} * \text{Micro-Recall}}{\text{Micro-Precision} + \text{Micro-Recall}}. \tag{20}$$

- **Macros** calculate metrics for each label, and find their unweighted mean, where the labeled imbalance is not considered:

$$\text{Macro-Precision} = \frac{1}{|C|} \sum_{c \in C} P_c, \tag{21}$$

$$\text{Macro-Recall} = \frac{1}{|C|} \sum_{c \in C} R_c, \tag{22}$$

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{c \in C} \frac{2P_c R_c}{P_c + R_c}. \tag{23}$$

4.1.3 Baselines. To comprehensively evaluate our proposed method for semi-supervised short text classification, we compared it with the following state-of-the-art methods:

SVM: SVM classifiers based on the classic manual features TF-IDF features, LDA features [3] and Doc2Vec [18], are denoted as SVM+TFIDF, SVM+LDA, and SVM+Doc2Vec, respectively.

CNN: CNN [16] with 2 variants: CNN-rand, whose word embeddings are randomly initialized; CNN-pretrain, whose word embeddings are pre-trained with Wikipedia Corpus.

LSTM: LSTM [22] with and without pre-trained word embeddings, named LSTM-rand and LSTM-pretrain, respectively.

BERT: BERT [6] uses a multi-layer bidirectional Transformer encoder and trains with a masked language model, leading to the state-of-the-art performance on several tasks. We applied the original BERT-base and its 2 variants: BERT-CNN with a CNN added to the task-specific layers of BERT model; BERT-LSTM with an LSTM added. The models are fine-tuned. Note that we have not preprocessed the datasets for BERT like other baselines, since the Tokenizer of BERT is designed for the raw corpus.

PTE: A semi-supervised representation learning method for text data [36]. It first learns word embedding based on the heterogeneous text networks containing three bipartite networks of words, documents and labels, then averages word embeddings as document embeddings for text classification.

TextGCN: TextGCN [51] models the text corpus as a graph containing documents and words as nodes with relations of document-word and word-word co-occurrence, and applies GCN for text classification.

HAN: HAN [46] embeds HINs by first converting an HIN to several homogeneous sub-networks through pre-defined meta-paths and then applying graph attention networks.

GCN-HIN: GCN [17] is a homogeneous graph embedding method for node classification. Here, as mentioned in Section 3.2.1, we adapt GCN to HIN by concatenating the feature spaces of different types of nodes. We name this basic method as GCN-HIN.

GAT-HIN: GAT [39] introduces attention mechanism into graph convolutional network to capture the node importance. Since GAT is also based on homogeneous graph, we similarly adjust GAT to adapt to HIN, namely, GAT-HIN. Besides, the attention mechanism allows it to perform inductive learning.

GraphSAGE: GraphSAGE [14] can be viewed as a stochastic generalization of homogeneous graph convolutions for inductive learning. There are four popular variants: GraphSAGE-GCN extends graph convolution network to the inductive setting; GraphSAGE-mean takes the element-wise mean value of feature vectors; GraphSAGE-LSTM conductd aggregation by feeding the neighborhood features into an LSTM; GraphSAGE-pool takes the element-wise max-pool of feature vectors transformed by a shared MLP. We have did the same adjustment as above for HIN and modified these models to run in the transductive mode, denoted with mark " * ."

All of the above baselines have used entity information from Wikipedia. Specifically, for SVMs, the mentions are replaced by the entity names. Deep neural models, including CNN and LSTM, have used the same entity embeddings (which are trained using Wikipedia corpus) as our proposed methods. Network embedding models, including GCN-HIN, GAT-HIN, HAN, and GraphSAGE, use the same graph (HIN for short texts) as our HGAT.

4.1.4 Parameter Settings. We applied the same parameters in transductive learning and inductive learning as follows.

We chose the parameter values of K, T, and δ that achieve the best results on the validation set. To construct HIN for short texts, we set the number of topics $K = 15$ in LDA for the datasets AGNews, TagMyNews, MR, and Twitter. We set $K = 20$ for Snippets and $K = 40$ for Ohsumed. For all the datasets, each document is assigned to top $P = 2$ topics with the largest probabilities. The similarity threshold δ between entities is set $\delta = 0.5$.

Following previous studies [38], we set the hidden dimension of our model HGAT and other neural models (Doc2Vec, LDA, CNN, LSTM, PTE, TexgGCN, HAN, GraphSAGE) to $d = 512$ and the dimension of pre-trained word embeddings to 100 (CNN, LSTM) . We set the layer number L of graph neural networks as 2 (i.e., HGAT, GCN-HIN, TextGCN, HAN, and GraphSAGE). The fusion hyper-parameter λ is also chosen according to the results on the validation set: We set $\lambda = 0.1$ for AGNews, Snippets and Twitter, $\lambda = 0.2$ for TagMyNews and MR, and $\lambda = 0.4$ for Ohsumed. For model training, we set the learning rate as 0.005, dropout rate as 0.8 and the regularization factor $\eta = 5e\text{-}8$. Early stopping is applied to avoid overfitting.

4.2 Experimental Results

4.2.1 Single-label Classification. Tables 2 and 3 report the results of our comparative performance evaluation in transductive learning and inductive learning, respectively.

Table 2. **Transductive Learning:** Test Accuracy (%) and F1-score (%) of Different Models on Six Standard Datasets

Dataset	Metrics	AGNews	Snippets	Ohsumed	TagMyNews	MR	Twitter
SVM+TFIDF	Accuracy	59.45	64.70	39.02	39.91	54.29	53.69
	F1-score	59.79	59.17	24.78	32.05	48.13	52.45
SVM+LDA	Accuracy	65.16	62.54	38.61	40.40	54.40	54.34
	F1-score	64.79	56.40	25.03	30.40	48.39	53.97
SVM+doc2vec	Accuracy	39.05	23.21	16.06	24.39	57.34	54.19
	F1-score	38.11	10.26	2.69	13.36	54.60	49.98
CNN-rand	Accuracy	32.65	48.34	35.25	28.76	54.85	52.58
	F1-score	32.00	42.12	13.95	15.82	51.23	51.91
CNN-pretrain	Accuracy	67.24	77.09	32.92	57.12	58.32	56.34
	F1-score	66.72	69.28	12.06	45.37	57.99	55.86
LSTM-rand	Accuracy	34.97	30.74	23.30	25.89	53.13	54.81
	F1-score	34.23	25.04	5.20	17.01	52.98	53.85
LSTM-pretrain	Accuracy	65.77	75.07	29.05	53.96	59.73	58.20
	F1-score	63.53	67.31	5.09	42.14	59.19	58.16
BERT	Accuracy	69.45	81.53	21.76	58.17	53.48	52.00
	F1-score	69.31	**79.03**	4.81	41.04	46.99	43.34
BERT+CNN	Accuracy	69.50	81.13	21.77	59.71	55.55	58.98
	F1-score	69.37	78.36	3.36	50.82	54.33	57.95
BERT+LSTM	Accuracy	67.65	80.85	21.46	60.83	58.67	57.27
	F1-score	63.48	74.69	3.27	45.34	57.70	56.77
PTE	Accuracy	36.00	63.10	36.63	40.32	54.74	54.24
	F1-score	35.41	58.96	19.24	33.56	52.36	53.17
TextGCN	Accuracy	67.61	77.82	41.56	54.28	59.12	60.15
	F1-score	67.12	71.95	**27.43**	46.01	58.98	59.82
GCN-HIN	Accuracy	70.87	76.69	40.25	56.33	60.81	61.59
	F1-score	69.23	74.85	16.70	50.18	59.03	59.99
GAT-HIN	Accuracy	70.92	71.56	39.88	56.49	60.24	60.19
	F1-score	69.43	70.37	16.95	50.31	59.11	59.82
*GraphSAGE-GCN	Accuracy	61.66	61.67	28.20	48.57	56.72	58.92
	F1-score	60.38	60.07	11.18	41.59	56.46	58.73
*GraphSAGE-pool	Accuracy	63.83	65.22	36.41	48.47	58.80	59.79
	F1-score	62.13	62.11	22.88	39.80	58.75	59.04
*GraphSAGE-mean	Accuracy	64.99	66.93	38.64	50.62	56.63	58.65
	F1-score	63.55	64.94	23.65	43.09	56.06	58.42
*GraphSAGE-LSTM	Accuracy	58.08	55.70	36.53	40.23	55.17	54.18
	F1-score	56.98	51.15	25.32	33.60	53.95	52.01
HAN-transductive	Accuracy	62.64	58.38	36.97	42.18	57.11	53.75
	F1-score	61.23	55.80	26.88	35.05	56.46	53.09
HGAT-transductive	Accuracy	**72.10***	**82.36***	**42.68***	**61.72***	**62.75***	**63.21***
	F1-score	**71.61***	74.44*	24.82*	**53.81***	**62.36***	**62.48***

The note * means our model significantly outperforms the baselines based on t-test ($p < 0.05$).

Table 3. **Inductive Learning:** Test Accuracy (%) and F1-score (%) of Different Models on Six Standard Datasets

Dataset	Metrics	AGNews	Snippets	Ohsumed	TagMyNews	MR	Twitter
GraphSAGE-GCN	Accuracy	63.92	62.94	29.43	48.15	58.64	57.72
	F1-score	62.46	60.75	11.39	41.88	58.47	57.60
GraphSAGE-pool	Accuracy	63.75	65.93	31.93	48.34	58.59	60.43
	F1-score	62.39	64.43	17.38	37.06	58.38	59.13
GraphSAGE-mean	Accuracy	63.33	65.40	31.24	50.25	57.88	59.33
	F1-score	62.13	64.41	15.79	43.01	57.63	58.11
GraphSAGE-LSTM	Accuracy	56.19	63.47	35.23	39.89	55.03	54.42
	F1-score	55.15	61.59	22.67	33.54	54.31	49.78
GAT-HIN-inductive	Accuracy	69.55	70.14	37.77	53.21	58.97	59.60
	F1-score	68.03	68.87	24.66	44.46	58.83	58.98
HAN-inductive	Accuracy	61.58	56.83	35.12	40.84	55.51	53.27
	F1-score	61.04	54.01	25.03	33.55	54.91	52.78
HGAT-inductive-0	Accuracy	61.85	62.82	39.58	37.19	56.26	52.67
	F1-score	61.23	57.11	23.94	27.16	50.73	50.15
HGAT-inductive-1	Accuracy	69.03*	79.00*	40.90*	**58.20***	59.80*	**62.60***
	F1-score	67.64*	72.12*	24.37*	**49.55***	59.31*	**60.47***
HGAT-inductive-2	Accuracy	**70.23***	**79.40***	**42.08***	57.83*	**61.18***	61.69*
	F1-score	**68.43***	**77.69***	**25.71***	46.80*	**59.77***	60.01*

The note * means our model significantly outperforms the baselines and traditional inductive learning based on t-test ($p < 0.05$).

For transductive learning, Table 2 reports the classification accuracy and F_1 score of different methods on six benchmark datasets. We can see that our methods significantly outperform all the baselines by a large margin, which shows the effectiveness of our proposed method on semi-supervised short text classification. The traditional methods SVMs based on the human-designed features, outperform the deep models with random initialization, i.e., CNN-rand and LSTM-rand in most cases. But CNN-pretrain and LSTM-pretrain based on the pre-trained vectors achieve significant improvements and outperform SVMs. The graph-based model PTE achieves inferior performance compared to CNN-pretrain and LSTM-pretrain. The reason could be that the word-occurrences are the key information for PTE to learn text embeddings, however, word-occurrences are very sparse during short text classification. Graph neural network-based models, i.e., TextGCN, GCN-HIN, GAT-HIN, HAN, and GraphSAGE, achieve comparable results with the deep models CNN-pretrain and LSTM-pretrain. Our model HGAT consistently outperforms all the state-of-the-art models by a large margin, verifying the effectiveness of our proposed method. The reasons are as follows: First, the constructed flexible HIN framework for modeling the short texts enables integration of additional information to enrich the semantics; Second, the novel proposed model HGAT embeds the HIN for short text classification based on a new dual-level attention mechanism, which not only captures the importance of different neighboring nodes (reducing the weights of noisy information) but also the importance of different types of nodes.

For inductive learning, Table 3 shows the classification accuracy and F_1 score of different methods on the six benchmark datasets. HGAT-inductive-0 represents HGAT using the traditional inductive setting. HGAT-inductive-1, 2 represent HGAT with inductive learning based on our proposed one-hop and two-hop expanded inductive graphs, respectively. Note that all the baseline models are run on the same inductive graph as HGAT-inductive-2, since the two-hop expanded

Table 4. Multi-label Classification Performance on Ohsumed-multi

Metrics	ER	Micro-P	Micro-R	Micro-F1	Macro-P	Macro-R	Macro-F1
CNN-pretrain	18.38	48.13	44.91	46.46	37.54	31.47	33.29
LSTM-pretrain	12.22	42.28	49.39	45.56	30.13	27.99	25.21
BERT	22.15	53.83	45.95	49.57	44.21	32.20	35.71
TextGCN	10.09	28.74	**82.69**	42.65	21.65	**86.18**	30.60
HAN-transductive	15.67	33.91	39.52	36.50	23.73	30.28	24.39
HGAT-transductive	**24.34***	**54.58***	46.46	**50.19***	**49.98***	37.73	**42.41***
GraphSAGE-GCN	19.04	47.49	29.29	36.23	48.99	15.88	19.53
GraphSAGE-pool	22.13	49.81	42.82	46.04	47.30	30.48	34.58
GraphSAGE-mean	22.15	50.32	41.83	45.68	**57.20**	28.27	34.10
GraphSAGE-LSTM	21.16	50.72	39.92	44.67	48.92	28.92	34.05
HAN-inductive	16.21	31.56	38.97	34.88	22.85	31.79	24.16
HGAT-inductive	**23.90***	**55.25***	**46.41***	**50.44***	51.93	**38.68***	**43.67***

The note * means our model significantly outperforms the baselines and traditional inductive learning based on t-test ($p < 0.01$).

inductive graph is more informative. From Table 3, we obtained the following observations. First, both HGAT-inductive-1 and HGAT-inductive-2 largely outperform all the baselines, verifying the effectiveness of our proposed HGAT for inductive learning. Second, HGAT-inductive-1 outperforms HGAT-inductive-0 by a large margin, confirming the necessity of the proposed expanded inductive graph for the new coming texts in semi-supervised short text classification. Third, HGAT-inductive-2 achieves the best performance in most cases, which indicates that as the number of hops increases, more information is introduced, thus better performance is achieved. However, if the number of hop becomes too large (e.g., TagMyNews and Twitter), some irrelevant information will be introduced, which may bring noise. A more detailed analysis about the impact of the number of hops for the expanded inductive graph is presented in Section 4.2.4.

4.2.2 Multi-label Classification. We also extended our model HGAT and the baseline models for multi-label classification and studied the performance of the models on a multi-label dataset, i.e., Ohsumed-multi. For fair comparison, we use the same margin-loss for all the models for multi-label classification. Table 4 presents the results of our model and baselines, in both transductive learning and inductive learning. As shown in Table 4, our HGAT greatly outperforms all the baselines in both transductive and inductive learning settings, validating the effectiveness of HGAT for multi-label classification. We also found that TextGCN performs very poorly on multi-label classification. This may be caused by the following two reasons: First, the nodes in TextGCN have no features but only one-hot vectors; second, the edges are very dense, especially between documents and words. These make it easy for TextGCN to predict more categories for the documents. The low Precision and extremely high Recall also verify this speculation. Our method, based on the HIN integrating the additional topic and entity information, can better distinguish the difference between the categories. Thus our model obtains superior performance for multi-label classification.

4.2.3 Comparison of Variants of HGAT. We further conduct several variants to compare with our model HGAT to validate the effectiveness of our model under the setting of transductive learning. As shown in Table 5, seven variant models are compared with our HGAT. The basic model GCN-HIN directly applies GCN on our constructed HIN for short texts by concatenating the feature spaces of different types of information. Another basic model GAT-HIN introduces attention mechanism into graph convolution. The two models both do not consider the heterogeneity of

Table 5. Test Accuracy (%) of Our Variants

Dataset	AGNews	Snippets	Ohsumed	TagMyNews	MR	Twitter
GCN-HIN	70.87	76.69	40.25	56.33	60.81	61.59
GAT-HIN	70.92	71.56	39.88	56.49	60.24	60.19
HGAT w/o ATT	70.97	80.42	41.31	59.41	62.13	62.35
HGAT-Type	71.54	81.68	41.95	60.78	62.27	62.95
HGAT-Node	71.76	81.93	42.17	61.29	62.31	62.45
HGAT w/o Orphan	71.79	82.01	42.23	61.27	62.40	62.03
HGAT	**72.10**	**82.36**	**42.68**	**61.72**	**62.75**	**63.21**

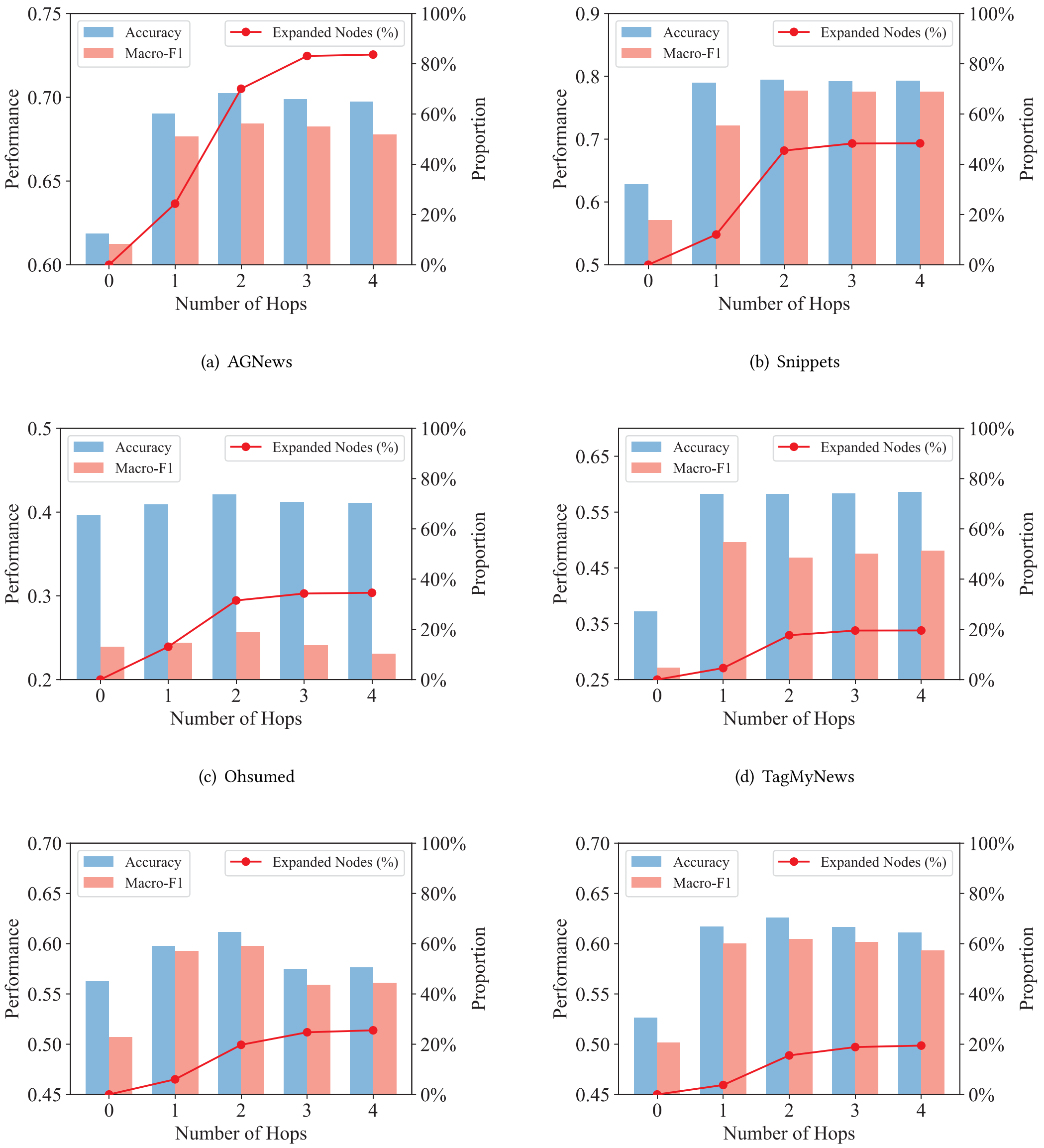
various information types. HGAT w/o ATT considers the heterogeneity through our proposed heterogeneous graph convolution, which projects different types of information to an implicit common space with respective transformation matrices. Based on HGAT w/o ATT, HGAT-Type, and HGAT-Node further consider only the type-level attention and node-level attention, respectively. HGAT w/o Orphan removes the module of orphan categories compared with the complete model HGAT.

We can see from Table 2, GAT-HIN sometimes performs better than GCN-HIN due to the attention mechanism, while sometime performs worse caused by overfitting. HGAT w/o ATT consistently outperforms these two basic models on all datasets, demonstrating the necessity of our proposed heterogeneous graph convolution, which considers the heterogeneity of various information types. HGAT-Type and HGAT-Node further improve HGAT w/o ATT by capturing the importance of different information (reducing the weights of noisy information). HGAT-Node surpassed HGAT-Type, indicating that node-level attention is more important. Besides, HGAT w/o Orphan is slightly inferior to the HGAT, which shows the effectiveness of the consideration of non-text categories, i.e., “entity” and “topic.” Finally, HGAT significantly outperforms all the variants by considering the heterogeneity and applying dual-level attention mechanism with node-level and type-level attentions.

4.2.4 Impact of Number of Hops for Inductive Graph. In this subsection, we studied the impact of the number of hops for the expanded inductive graph. Figure 5 illustrates the performance on the six single-label datasets. We can see that HGAT-inductive-0 (i.e., when the number of hops is 0) achieves the worst performance. When the number of hops increases to 1, the performance is substantially improved, which is caused by the use of the information from existing labeled and unlabeled short texts instead of information from only new coming texts. Moreover, as the hop increases, the accuracy and F1 score first go up, reach the optimal value when the hop is 1 or 2, and then tend to moderate or decrease. The reason is that at the beginning, our model benefits from the additional information from existing data. However, when the number of the hops becomes too large, some noise may be introduced.

4.2.5 Impact of Sampling Strategy. In this subsection, we conduct a series of experiments on the six datasets to evaluate the effects of three common sampling strategies (compared with no sampling) and the number of sampling neighbors. Specifically, as illustrated in Figure 6, we vary the number of samples from 1 to 20. *Random*, *TopK* and *Weighted* represent uniform random sampling, top-K pruning and weighted random sampling, respectively. *None* means no sampling strategy is applied.

As reported in Figure 6, as the number of sampling neighbors increases, the performance of all sampling strategies consistently becomes better on all datasets, because more information from



(a) AGNews

(b) Snippets

(c) Ohsumed

(d) TagMyNews

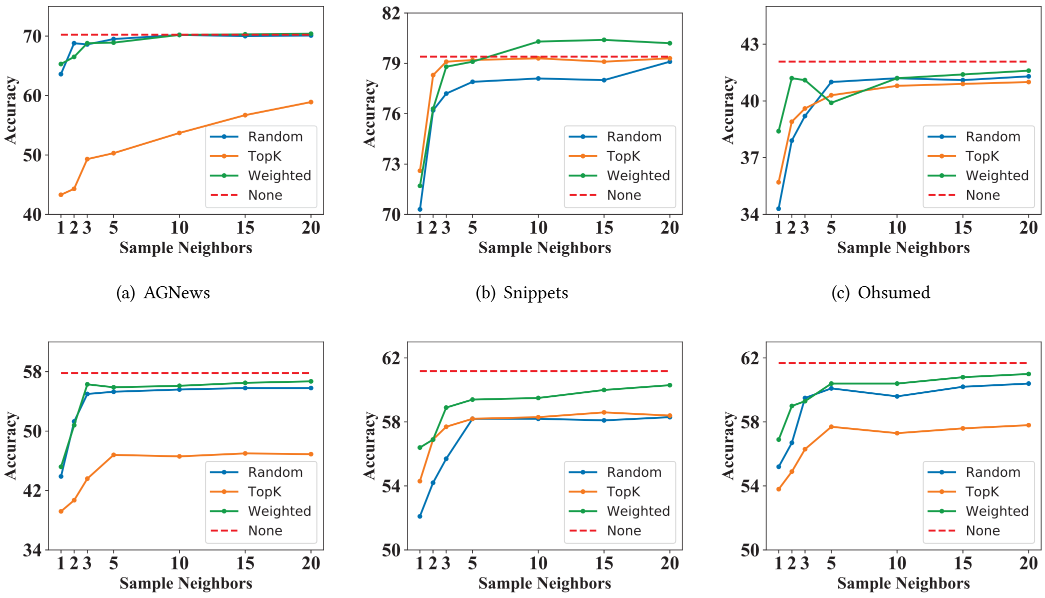
(e) MR

(f) Twitter

Fig. 5. The bars show the test accuracy (blue bars) and F1 score (red bars) with different numbers of hops for the expanded inductive graph. Red lines show the proportion of the expanded nodes in the existing data.

neighboring nodes is integrated. Compared with no sampling, there is an obvious performance drop in most cases, due to the information loss caused by sampling. However, note that applying a sampling strategy will greatly improve the efficiency, which is more suitable for real-world applications.

Detailed discussions are as follows. Compared with *TopK*, the performance of *Random* is better in most cases. It proves our assumption mentioned in Section 3.4 that TopK pruning directly



(a) AGNews

(b) Snippets

(c) Ohsumed

(d) TagMyNews

(e) MR

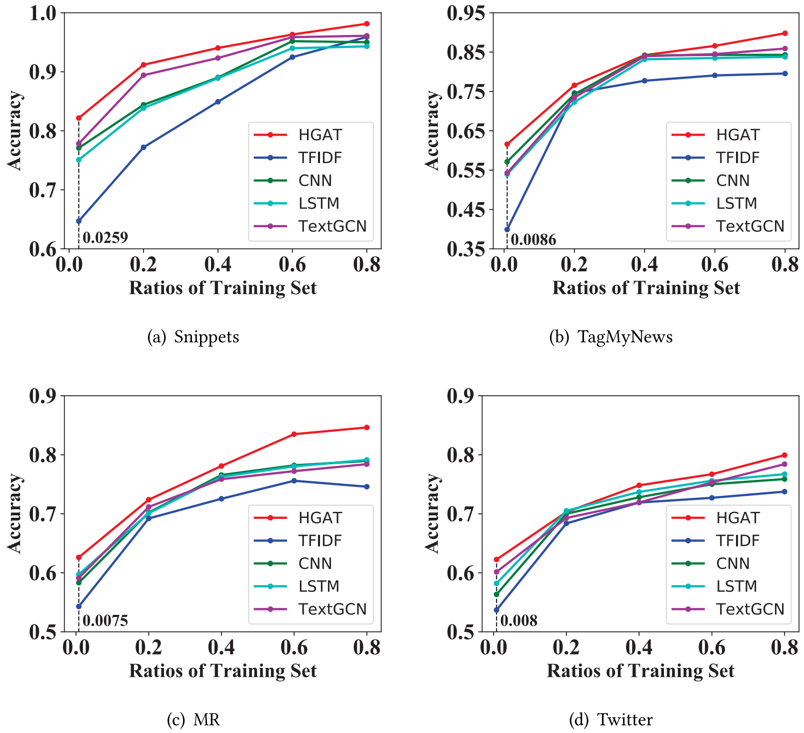
(f) Twitter

Fig. 6. The test accuracy with different sampling strategies and various numbers of sample neighbors. "None" means no sampling strategy is applied.

choosing the most similar neighbors (i.e., they have the biggest attention scores) will lose some relevant but not similar information. Compared with *Weighted*, *Random* obtains inferior performance. We analyze that this phenomenon is caused by a similar reason: Although *Random* involved more relevant but not similar information, it also introduces too much noise. Therefore, *Weighted* making a trade-off between noise and effective information can get better results. Therefore, a more appropriate sampling strategy can be explored in the future to reduce complexity while ensuring as good a performance as possible.

4.2.6 Impact of Training Set Ratios. We chose five representative methods with the best performance: HGAT, SVM+TFIDF, CNN-pretrain, LSTM-pretrain, TextGCN, to study the impact of the number of training set ratios. Particularly, we varied the ratios of training set and compared their performance on four datasets[7]: Snippets, TagMyNews, MR, and Twitter. We ran each method 10 times and report the average performance. As shown in Figure 7, with the increase of training data, all the methods achieve better results in terms of accuracy. Generally, the graph-based methods TextGCN and HGAT perform better, indicating that graph-based methods can make better use of limited labeled data through information propagation. Our method outperforms all the other methods consistently. When fewer labeled documents are provided, the baselines exhibit obvious performance drop, especially the traditional method SVM+TFIDF, while our model still achieves relatively high performance. It demonstrates that our method can more effectively exploit the limited labeled data for short text classification. Moreover, when the ratios of the training set becomes normal, i.e., not under the extreme semi-supervised settings, one can observe our model

[7] The other two datasets have different settings compared with previous work. Specifically, for AGNews, we randomly selected 6,000 pieces from the whole dataset due to its large scale. For Ohsumed, we remove the abstracts in the corpus for short text classification.



(a) Snippets

(b) TagMyNews

(c) MR

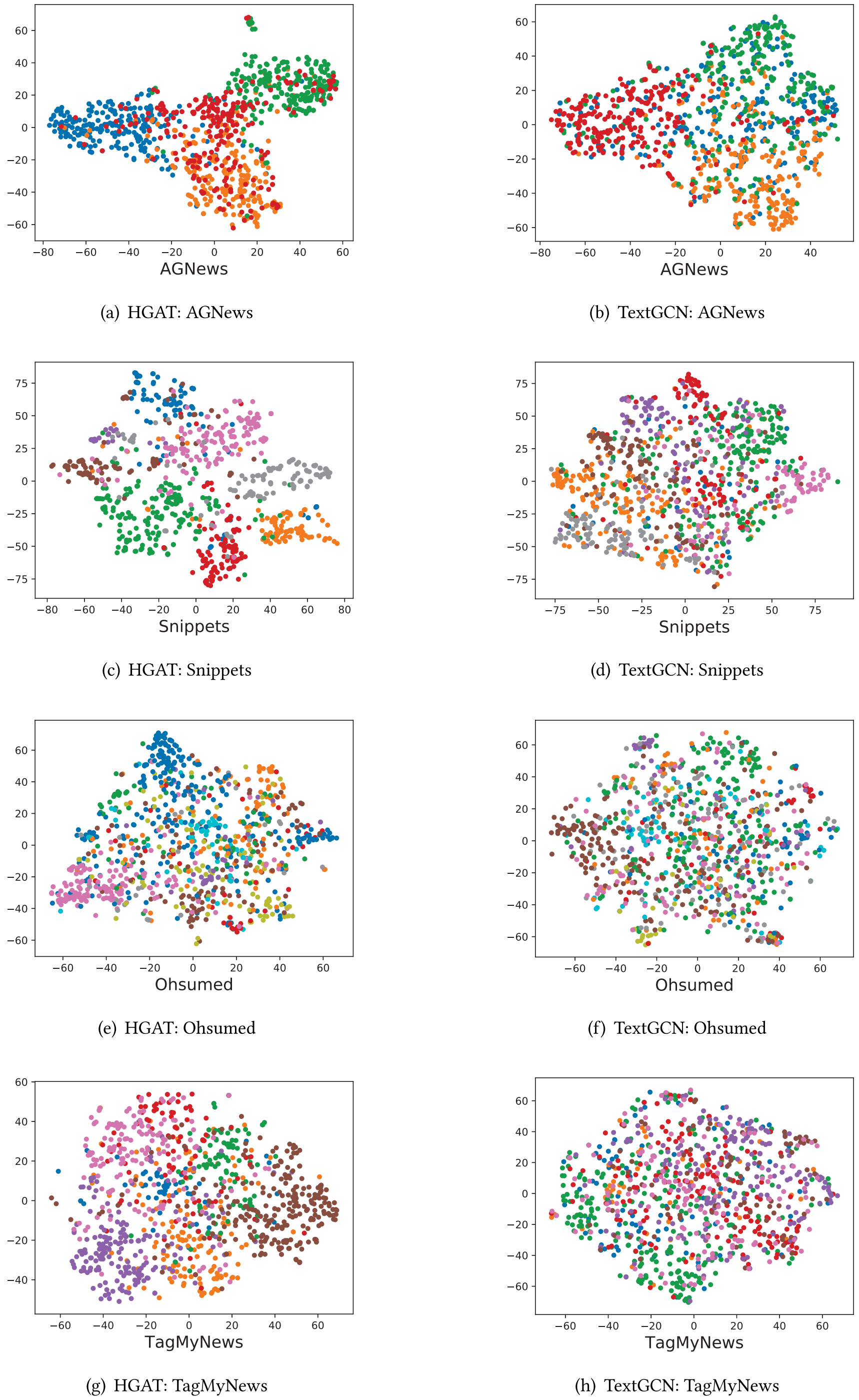
(d) Twitter

Fig. 7. The test accuracy with different ratios of training set. Note that the lines are started with the settings in Section 4.2.1, rather than ratio = 0. Besides, the standard split ratios of the four datasets are 0.8, 0.8, 0.66, 0.8, respectively.

consistently keep superior. We believe our method benefits from the flexible HIN and the proposed heterogeneous graph attention networks with dual-level attention.

4.2.7 Visualization of Document Embeddings. Figure 8 visualises the short text embeddings of HGAT and TextGCN. We chose four datasets and visualized the embeddings of randomly-sampled 1,000 short texts from the test sets of the four datasets. From Figure 8, we observed that compared to TextGCN, HGAT learns closer document embeddings of the same category, and it is easier to distinguish documents of different categories. Even for more difficult datasets such as Ohsumed and TagMyNews, it is still more clear to observe the clustering phenomenon of document embeddings for HGAT, compared to TextGCN.

4.2.8 Parameter Analysis. In this subsection, we study the impact of different numbers of topics K and top relevant topics P assigned to a document, and different values of the hyper-parameter λ. The accuracy of our model on the six single-label datasets is illustrated in Figure 9. It is clear that for the number of topics, the test accuracy first increases with the increase of the number of topics, reaching the highest value at 15 on most datasets; then it falls when the number is larger. We also tried the different number of topics for the baselines, and observed that the best K is the same as that in our model. This is consistent with the intuition that the number of topics should fit the dataset, i.e., it should be model-free. For the number of top relevant topics P assigned to the documents, the test accuracy first increases with the increase of P and then decreases when P is larger than 2 on all the datasets. For the hyper-parameter λ, we find that our model is insensitive



(a) HGAT: AGNews

(b) TextGCN: AGNews

(c) HGAT: Snippets

(d) TextGCN: Snippets

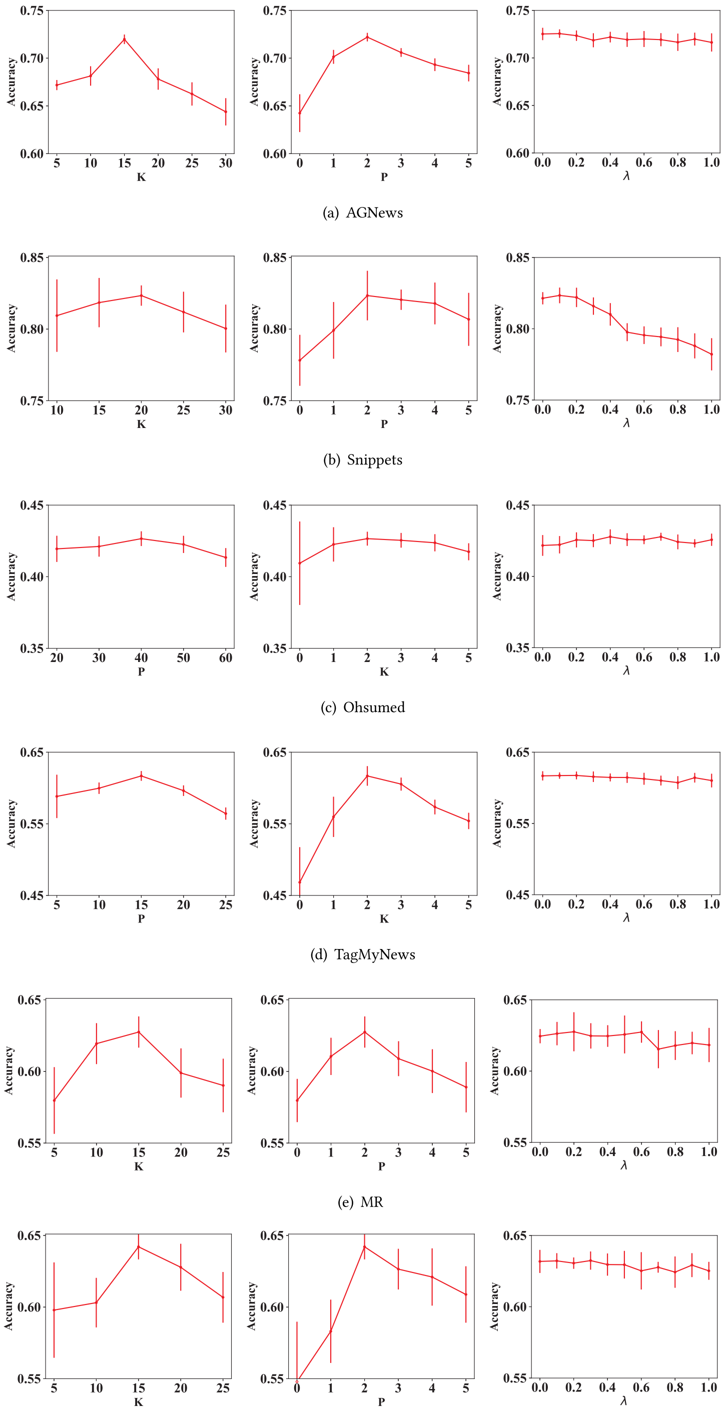
(e) HGAT: Ohsumed

(f) TextGCN: Ohsumed

(g) HGAT: TagMyNews

(h) TextGCN: TagMyNews

Fig. 8. Visualization of the short text embeddings on the test set of the four datasets where performance is relatively significant.



(a) AGNews

(b) Snippets

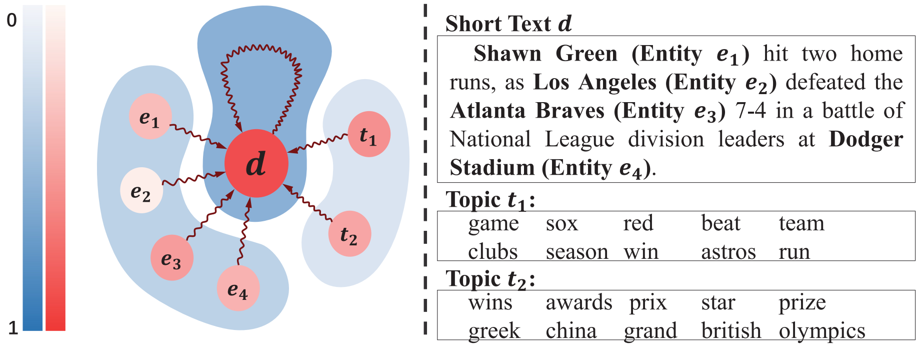
(c) Ohsumed

(d) TagMyNews

(e) MR

(f) Twitter

Fig. 9. The average accuracy with different number of topics K, top P relevant topics, and hyper-parameter λ to alleviate overfitting on the six single-label datasets.



Short Text d

Shawn Green (Entity e_1) hit two home runs, as **Los Angeles (Entity e_2)** defeated the **Atlanta Braves (Entity e_3)** 7-4 in a battle of National League division leaders at **Dodger Stadium (Entity e_4)**.

Topic t_1:

game	sox	red	beat	team
clubs	season	win	astros	run

Topic t_2:

wins	awards	prix	star	prize
greek	china	grand	british	olympics

Fig. 10. Visualization of the dual-level attention including node-level attention (shown in red) and type-level attention (shown in blue). Each topic t is represented by top 10 words with highest probabilities.

to λ on most datasets. The best performance usually occurs when λ is 0.1 or 0.2. Note that when λ is 1.0, it means the mechanism of the overfitting alleviation is completely removed, thus leading to a performance drop especially for dataset snippets. In our experiments, the three parameters are set based on the validation set of each dataset.

4.2.9 Case Study. As Figure 10 shows, we took a short text from AGNews as an example (which is classified to the category of sports correctly) to illustrate the dual-level attention of HGAT. The type-level attention assigns high weight (0.7) to the short text itself, while lower weights (0.2 and 0.1) to entities and topics. It means that the text itself contributes more to the classification, than the entities and topics. Besides, the node-level attention assigns different weights to neighboring nodes, and the node-level weights of nodes belonging to a same type sum to 1. As we see, the entities e_3 (Atlanta Braves, a baseball team), e_4 (Dodger Stadium, a baseball gym), e_1 (Shawn Green, a baseball player) have higher weights than e_2 (Los Angeles, referring to a city at most time). The topics t_1 (game) and t_2 (win) have almost the same importance in classifying the text to the category of sports. The case study shows that our proposed dual-level attention can capture key information at multiple granularities for classification and reduce the weights of noisy information.

5 CONCLUSION

In this article, we propose a novel heterogeneous graph neural network-based method for semi-supervised short text classification, which takes full advantage of both limited labeled and large unlabeled data by information propagation. Particularly, we first present a flexible HIN framework for modeling the short texts, which can integrate any additional information and capture their rich relations to address the semantic sparsity of short texts. Then, we propose a novel model HGAT to embed the HIN based on a dual-level attention mechanism including node-level and type-level attentions. HGAT considers the heterogeneity of various information types by projecting them into an implicit common space. Additionally, the dual-level attention captures the key information at multiple granularity levels and reduces the weights of noisy information. To deal with the new coming texts not previously existing in the HIN, we extend our model HGAT for inductive learning. Moreover, we improve our HGAT by introducing orphan categories to reduce the classification interference of the non-text categories in the HIN for short texts. Extensive experimental results on single-/multi-label classification have demonstrated that our proposed model HGAT consistently and significantly outperforms state-of-the-art methods across the benchmark datasets under both transductive and inductive learning.

For the future, since our model HGAT is a general HIN embedding approach, it would be interesting to apply it to other tasks, e.g., HIN-based recommendation. Besides, a more effective neighbor-sampling strategy is also worth exploring.

REFERENCES

[1] Charu C. Aggarwal and ChengXiang Zhai. 2012. A survey of text classification algorithms. In *Mining Text Data*. Springer, 163–222. DOI: https://doi.org/10.1007/978-1-4614-3223-4_6

[2] Faizan Ahmad, Ahmed Abbasi, Jingjing Li, David G. Dobolyi, Richard G. Netemeyer, Gari D. Clifford, and Hsinchun Chen. 2020. A deep learning architecture for psychometric natural language processing. *ACM Trans. Info. Syst.* 38, 1, Article 6 (Feb. 2020), 29 pages. DOI: https://doi.org/10.1145/3365211

[3] David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3 (May 2003), 993–1022. DOI: https://doi.org/10.1162/jmlr.2003.3.4-5.993

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR'14)*, Yoshua Bengio and Yann LeCun (Eds.). OpenReview.net. Retrieved from http://arxiv.org/abs/1312.6203.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the Annual Conference on Neural Information Processing Systems: Advances in Neural Information Processing Systems 29*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 3837–3845. Retrieved from http://papers.nips.cc/paper/6081-convolutional-neural-networks-on-graphs-with-fast-localized-spectral-filtering.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'19)*. Association for Computational Linguistics, 4171–4186. DOI: https://doi.org/10.18653/v1/n19-1423

[7] Di Yao, Jingping Bi, Jianhui Huang, and Jin Zhu. 2015. A word distributed representation-based framework for large-scale short text classification. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'15)*. IEEE, 1–7. DOI: https://doi.org/10.1109/IJCNN.2015.7280513

[8] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 135–144. DOI: https://doi.org/10.1145/3097983.3098036

[9] Harris Drucker, Donghui Wu, and Vladimir Vapnik. 1999. Support vector machines for spam categorization. *IEEE Trans. Neural Netw.* 10, 5 (1999), 1048–1054. DOI: https://doi.org/10.1109/72.788645

[10] Jernej Flisar and Vili Podgorelec. 2020. Improving short text classification using information from DBpedia ontology. *Fundamenta Informaticae* 172, 3 (Feb. 2020), 261–297. DOI: https://doi.org/10.3233/FI-2020-1905

[11] Erfan Ghadery, Sajad Movahedi, Heshaam Faili, and Azadeh Shakery. 2019. MNCN: A multilingual ngram-based convolutional network for aspect category detection in online reviews. *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), 6441–6448. DOI: https://doi.org/10.1609/aaai.v33i01.33016441

[12] Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, Vol. 2. IEEE, 729–734. DOI: https://doi.org/10.1109/IJCNN.2005.1555942

[13] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864. DOI: https://doi.org/10.1145/2939672.2939754

[14] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the Annual Conference on Neural Information Processing Systems: Advances in Neural Information Processing Systems 30*. 1024–1034. Retrieved from http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.

[15] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. 2010. Graph regularized transductive classification on heterogeneous information networks. In *Machine Learning and Knowledge Discovery in Databases*. Vol. 6321. Springer, Berlin, 570–586. DOI: https://doi.org/10.1007/978-3-642-15880-3_42

[16] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1746–1751. DOI: https://doi.org/10.3115/v1/d14-1181

[17] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR'17)*. OpenReview.net. Retrieved from https://openreview.net/forum?id=SJU4ayYgl.

[18] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of Machine Learning Research*, Vol. 32. PMLR, 1188–1196. Retrieved from http://proceedings.mlr.press/v32/le14.html.

[19] Chenliang Li, Shiqian Chen, Jian Xing, Aixin Sun, and Zongyang Ma. 2018. Seed-guided topic model for document filtering and classification. *ACM Trans. Info. Syst.* 37, 1, Article Article 9 (Dec. 2018), 37 pages. DOI: https://doi.org/10.1145/3238250

[20] Chenliang Li, Yu Duan, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2017. Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Trans. Info. Syst.* 36, 2, Article 11 (Aug. 2017), 30 pages. DOI: https://doi.org/10.1145/3091108

[21] Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. 2019. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics. DOI: https://doi.org/10.18653/v1/d19-1488

[22] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 2873–2879.

[23] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM'18)*. Association for Computing Machinery, 983–992. DOI: https://doi.org/10.1145/3269206.3271737

[24] Liqiang Nie, Yongqi Li, Fuli Feng, Xuemeng Song, Meng Wang, and Yinglong Wang. 2020. Large-scale question tagging via joint question-topic embedding learning. *ACM Trans. Info. Syst.* 38, 2 (2020). DOI: https://doi.org/10.1145/3380954

[25] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (ACL'05)*, Kevin Knight, Hwee Tou Ng, and Kemal Oflazer (Eds.). Association for Computer Linguistics, 115–124. Retrieved from https://www.aclweb.org/anthology/P05-1015/.

[26] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 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 (WWW'08)*. ACM Press. DOI: https://doi.org/10.1145/1367497.1367510

[27] Rafael Geraldeli Rossi, Alneu de Andrade Lopes, and Solange Oliveira Rezende. 2016. Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Info. Process. Manage.* 52, 2 (Mar. 2016), 217–257. DOI: https://doi.org/10.1016/j.ipm.2015.07.004

[28] François Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. 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 of the Asian Federation of Natural Language Processing (ACL'15)*. The Association for Computer Linguistics, 1702–1712. DOI: https://doi.org/10.3115/v1/p15-1164

[29] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Proceedings of the Annual Conference on Neural Information Processing Systems: Advances in Neural Information Processing Systems 30*. Curran Associates, 3856–3866. Retrieved from http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.

[30] Franco Scarselli, Marco Gori, Ah Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Trans. Neural Netw.* 20 (Jan. 2009), 61–80. DOI: https://doi.org/10.1109/TNN.2008.2005605

[31] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *Comput. Surveys* 34, 1 (Mar. 2002), 1–47. DOI: https://doi.org/10.1145/505282.505283

[32] Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. HFT-CNN: Learning hierarchical category structure for multi-label short text categorization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 811–816. DOI: https://doi.org/10.18653/v1/d18-1093

[33] Joao Silva, Luisa Coheur, Ana Cristina Mendes, and Andreas Wichert. 2011. From symbolic to sub-symbolic information in question classification. *Artific. Intell. Rev.* 35, 2 (Feb. 2011), 137–154. DOI: https://doi.org/10.1007/s10462-010-9188-4

[34] Koustuv Sinha, Yue Dong, Jackie Chi Kit Cheung, and Derek Ruths. 2018. A hierarchical neural attention-based text classifier. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 817–823. DOI: https://doi.org/10.18653/v1/d18-1094

[35] Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang, and Shifu Bie. 2014. Short text classification: A survey. *J. Multimedia* 9, 5 (May 2014), 635. DOI: https://doi.org/10.4304/jmm.9.5.635-643

[36] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Longbing Cao, Chengqi Zhang, Thorsten Joachims, Geoffrey I. Webb, Dragos D. Margineantu, and Graham Williams (Eds.). ACM, 1165–1174. DOI: https://doi.org/10.1145/2783258.2783307

[37] Jesper E. Van Engelen and Holger H. Hoos. 2020. A survey on semi-supervised learning. *Mach. Learn.* 109, 2 (Feb. 2020), 373–440. DOI: https://doi.org/10.1007/s10994-019-05855-6

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems: Advances in Neural Information Processing Systems 30*. 5998–6008. Retrieved from http://papers.nips.cc/paper/7181-attention-is-all-you-need.

[39] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*. OpenReview.net. Retrieved from https://openreview.net/forum?id=rJXMpikCZ.

[40] Daniele Vitale, Paolo Ferragina, and Ugo Scaiella. 2012. Classification of short texts by deploying topical annotations. In *Lecture Notes in Computer Science*. Springer, Berlin, 376–387. DOI: https://doi.org/10.1007/978-3-642-28997-2_32

[41] Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. 2016. Text classification with heterogeneous information network kernels. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 2130–2136. Retrieved from http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12392.

[42] Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Vol. 350. International Joint Conferences on Artificial Intelligence Organization. DOI: https://doi.org/10.24963/ijcai.2017/406

[43] Pu Wang and Carlotta Domeniconi. 2008. Building semantic kernels for text classification using wikipedia. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*. ACM Press, Las Vegas, NV, 713. DOI: https://doi.org/10.1145/1401890.1401976

[44] Sida I. Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. The Association for Computer Linguistics, 90–94. Retrieved from https://www.aclweb.org/anthology/P12-2018/.

[45] Xiang Wang, Ruhua Chen, Yan Jia, and Bin Zhou. 2013. Short text classification using Wikipedia concept-based document representation. In *Proceedings of the International Conference on Information Technology and Applications*. IEEE, 471–474. DOI: https://doi.org/10.1109/ita.2013.114

[46] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous graph attention network. In *Proceedings of the World Wide Web Conference (WWW'19)*. ACM Press. DOI: https://doi.org/10.1145/3308558.3313562

[47] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*. IEEE Computer Society, 6857–6866. DOI: https://doi.org/10.1109/CVPR.2018.00717

[48] Jingyun Xu, Yi Cai, Xin Wu, Xue Lei, Qingbao Huang, Ho-fung Leung, and Qing Li. 2020. Incorporating context-relevant concepts into convolutional neural networks for short text classification. *Neurocomputing* 386 (Apr. 2020), 42–53. DOI: https://doi.org/10.1016/j.neucom.2019.08.080

[49] Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 3110–3119. DOI: https://doi.org/10.18653/v1/D18-1350

[50] Yiming Yang and Christopher G. Chute. 1994. An example-based mapping method for text categorization and retrieval. *ACM Trans. Info. Syst.* 12, 3 (July 1994), 252–277. DOI: https://doi.org/10.1145/183422.183424

[51] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), 7370–7377. DOI: https://doi.org/10.1609/aaai.v33i01.33017370

[52] Chunyong Yin, Jun Xiang, Hui Zhang, Jin Wang, Zhichao Yin, and Jeong-Uk Kim. 2015. A new SVM method for short text classification based on semi-supervised learning. In *Proceedings of the 4th International Conference on Advanced Information Technology and Sensor Application (AITS'15)*. IEEE, 100–103. DOI: https://doi.org/10.1109/aits.2015.34

[53] Jichuan Zeng, Jing Li, Yan Song, Cuiyun Gao, Michael R. Lyu, and Irwin King. 2018. Topic memory networks for short text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 3120–3131. DOI: https://doi.org/10.18653/v1/d18-1351

[54] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the Annual Conference on Neural Information Processing Systems: Advances in Neural Information Processing Systems 28*. 649–657. Retrieved from http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification.

[55] Dengyong Zhou, Olivier Bousquet, Thomas N. Lal, Jason Weston, and Bernhard Schölkopf. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. MIT Press, 321–328. Retrieved from http://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf.

[56] Guang-You Zhou and Jimmy Xiangji Huang. 2017. Modeling and mining domain shared knowledge for sentiment analysis. *ACM Trans. Info. Syst.* 36, 2, Article 18 (Aug. 2017), 36 pages. DOI: https://doi.org/10.1145/3091995

[57] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on International Conference on Machine Learning (ICML'03)*. AAAI Press, 912–919.

Received May 2020; revised January 2021; accepted February 2021