Fake News Detection on News-Oriented Heterogeneous Information Networks through Hierarchical Graph Attention

Yuxiang Ren
IFM Lab
Department of Computer Science
Florida State University
Tallahassee, FL, USA
yuxiang@ifmlab.org

Jiawei Zhang
IFM Lab
Department of Computer Science
Florida State University
Tallahassee, FL, USA
jiawei@ifmlab.org

***Abstract*—The viral spread of fake news has caused great social harm, making fake news detection an urgent task. Current fake news detection methods rely heavily on text information by learning the extracted news content or writing style of internal knowledge. However, deliberate rumors can mask writing style, bypassing language models and invalidating simple text-based models. In fact, news articles and other related components (such as news creators and news topics) can be modeled as a heterogeneous information network (HIN for short). In this paper, we propose a novel fake news detection framework, namely Hierarchical Graph Attention Network(HGAT), which uses a novel hierarchical attention mechanism to perform node representation learning in HIN, and then detects fake news by classifying news article nodes. Experiments on two real-world fake news datasets show that HGAT can outperform text-based models and other network-based models. In addition, the experiment proved the expandability and generalizability of our for graph representation learning and other node classification related applications in heterogeneous graphs.**

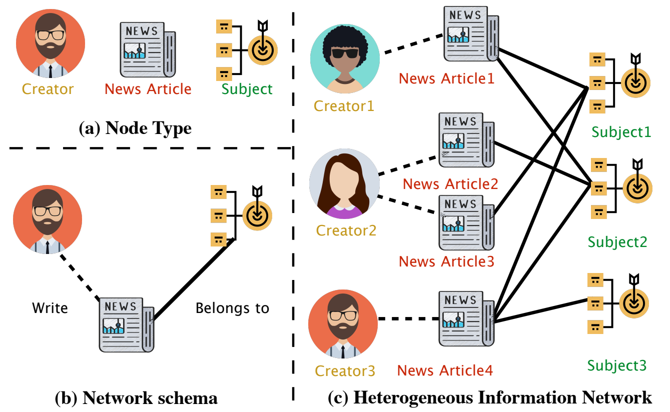


Figure 1. An illustrative example of a heterogenous information network based on PoliticFact data (News-HIN). (a) Three types of nodes (i.e., Creator, News article, Subject). (b) Network schema of News-HIN (c) A News-HIN consists three types of nodes and two types of links.

I. Introduction

With explosive growth, fake news has already caused severe threats to the public's factual judgment and governments' credibility. Especially with the wide use of social platforms, they facilitate the generation and dissemination of fake news. For example, during the 2016 US presidential election, numerous fake news about presidential candidates is spread on various social platforms [7]. For example, 115 pro-Trump fake stories shared on Facebook a total of 30 million times, and 41 pro-Clinton fake stories shared a total of 7.6 million times are observed in [1]. Such a massive amount of widely spread fake news has greatly destroyed candidates' public persona and misled voters' judgment. It has become very critical to detect fake news on social media in time to block the spread.

There are significant differences between fake news and traditional fraudulent information. First, fake news is intentionally edited by the creators to achieve the purpose of misleading the public. For instance, news about the same event published by different creators is highly similar in most content, but fake news carries malicious content in objective statements. Although the proportion of these malicious contents is negligible, this is enough to make the news a harmful fake one. Second, for traditional suspicious information, like spam [21], people instinctively have a precautionary mentality that makes themselves less likely to be deceived. But for news, people usually actively search, receive, and share without being on guard about authenticity. Third, spams usually are easier to be detected because of the abundant regular messages; yet, detecting fake news is incredibly challenging since news is very time-sensitive. The evidence collected about past news can not benefit the detection of emerging fake news apparently.

These characteristics of fake news make the detection more challenging. In order to detect fake news more effectively, it is necessary to mine meaningful information from different views instead of focusing on the news contents solely. In fact, fake news does not exist independently in the form of articles. For example, news creators and news subjects also exist in online social media. The information from news creators and news subjects describes the news in a more comprehensive view and helps us more thoroughly eliminate fake news and relating components. In detail, for the news creators, we can collect profile information and other supplementary knowledge. As for the news subjects, background knowledge and other related information can be collected to support fake news detection. News articles and other related components can be modeled as a heterogeneous information network (HIN for short) [13]. HINs have a powerful capability of representing rich information, and we formulate fake news detection as the node classification problem in the HIN in this paper. We present an illustrative example of a news-oriented heterogeneous information network (News-HIN) in Figure 1.

Problem Studied: In this paper, we propose to study the HIN-based fake news detection problem. We model the fake news detection problem as a node classification task in the HIN, which requires us to learn the more comprehensive and discriminative representation of news article nodes.

The main challenges of the fake news detection problem in the HIN lie in the following three points:

- *Heterogeneity*: There exist various types of heterogeneous information related to news articles. Learning effective node representations in a HIN in a unified way is not an easy task.
- *Hierarchy*: Representation learning in heterogeneous graphs will be a multi-level work because node contents and the information of the schema are contained at different levels.
- *Generalizability*: To ensure the proposed model's applicability to different types of HINs, we need to propose a general learning model that can be extensible to various learning settings.

To handle these challenges aforementioned, we propose a novel **H**ierarchical **G**raph **A**ttention **N**etwork (HGAT) to detect fake news. HGAT employs a hierarchical attention mechanism to learn the representation of news article nodes. Based on the learned node representation, fake news can be identified through the node classification task. In particular, for each news article node, we use the node-level attention mechanism to learn a set of weights for its neighbors of the same type. Using these sets of weights, we aggregate neighbor nodes of the same type into a schema node. The schema-level attention works to learn the attention weights of different schema nodes. Based on the two-level attention, HGAT can get the optimal combination of different types of neighbors in a hierarchical manner. The learned node representations capture the features from different heterogeneous information sources. HGAT can be optimized in an end-to-end manner by backpropagation.

The contributions of our work are summarized as follows:

- We attempt to detect the fake news in the heterogeneous information network with the support of the heterogeneous graph neural network, while without handcrafted features (e.g., meta-path).
- We propose the novel HGAT model, which takes different types of node contents and diverse categories of connections into consideration simultaneously. HGAT, as a general model for representation learning, has excellent potential to be applied to other applications in the HIN.
- We conduct extensive experiments on two real-world datasets to demonstrate the effectiveness of HGAT.

II. Related Work

A. Fake News Detection

As an emerging topic, some research works have been proposed. Among them, the knowledge-based approach aims to assess the authenticity of news by comparing the knowledge extracted from the news contents with real knowledge [2]. Yet, the timeliness and integrity of the knowledge map remain an unresolved issue [24]. Another typical way is based on writing style, such as discourse level by employing rhetorical structure theory [12], and sentiment & readability [10]. Based on relationships among news articles, users (spreaders) and user posts, matrix factorization [14], tensor factorization [5], hierarchical word encoder [3], and Recurrent Neural Networks (RNNs) [23] have been developed for fake news detection.

B. GNNs and Network Embedding

Graph Neural Networks (GNNs) for representation learning of graphs learn nodes' effective feature vectors through a recursive neighborhood aggregation scheme [22]. Kipf et al. [8] propose Graph Convolutional Network (GCN). Graph Attention Network (GAT) [17] first imports the attention mechanism into graphs. However, the above graph neural networks are presented for the homogeneous graphs. Wang et al. [20] consider the attention mechanism in heterogeneous graph learning through the model HAN. However, meta-path as a handcrafted feature limits HAN, and HAN ignores node contents carried by other types of nodes. The learned embeddings from network embedding methods can be applied to the downstream tasks [20]. Some models have been proposed to deal with homogeneous networks, including the random walk based methods [6], the matrix factorization based methods [19], the deep learning-based methods [18]. In order to handle the heterogeneity, metapath2vec [4] samples random walks under the guidance of meta-path on heterogeneous graphs.

III. Concept and Problem Definition

In this section, we use the *PolitiFact* data as an example to introduce some terminologies used in this paper. These concepts are equally applicable to other datasets.

A. Terminology Definition

PolitiFact dataset contains three types of entities: news articles, subjects, and creators. They can be modeled as three types of nodes in a heterogeneous network, and different types of links are constructed according to the connections between them.

Definition *1:* (News Articles): News articles refer to the news contents post on social media or public platforms. News articles can be represented as set $\mathcal{N} = \{n_1, n_2, \cdots, n_m\}$. For each news article $n_i \in \mathcal{N}$, it contains textual contents.

Definition *2:* (Subject): Subjects usually denote news articles' central ideas, which are the main objectives of writing news articles. The set of subjects can be denoted as $\mathcal{S} = \{s_1, s_2, \cdots, s_k\}$. For each subject $s_i \in \mathcal{S}$, it contains the textual description.

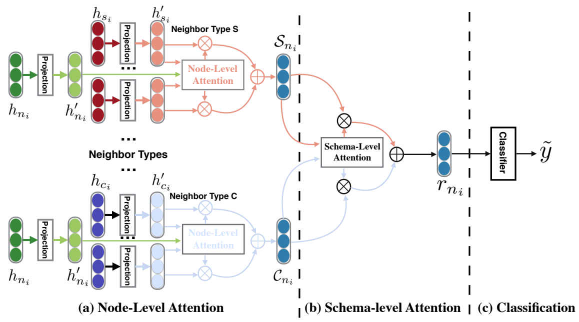


Figure 2. The overall framework of HGAT. (a) All types of nodes are projected into a unified feature space and the weights of node pairs can be learned via node-level attention. (b) Joint learning the weight of each type of schema nodes and fuse representations via schema-level attention. (c) Calculate the loss and end-to-end optimization.

DEFINITION *3:* (Creator): Creators denote users who write news articles. The set of creators can be represented as $\mathcal{C} = \{c_1, c_2, \cdots, c_n\}$. For each creator $c_i \in \mathcal{C}$, it contains the profile information, including titles, political party membership, and geographical residential locations.

A formal definition of News Oriented Heterogeneous Information Networks can be proposed as follows:

DEFINITION *4:* (News Oriented Heterogeneous Information Networks (News-HIN)): The news oriented heterogeneous information network (News-HIN) can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V} = \mathcal{C} \cup \mathcal{N} \cup \mathcal{S}$, and the link set $\mathcal{E} = \mathcal{E}_{c,n} \cup \mathcal{E}_{n,s}$ involves the "Write" links between creators and news articles, and the "Belongs to" links between news articles and subjects.

In order to better understand the News-HIN and utilize type information, it is necessary to define the schema-level description.

DEFINITION *5:* (News-HIN Schema): Formally, the schema of the given News-HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be represented as $S_{\mathcal{G}} = (\mathcal{V}_{type}, \mathcal{E}_{type})$, where $\mathcal{V}_{type}$ and $\mathcal{E}_{type}$ denote the set of node types and link types respectively. Here, $\mathcal{V}_{type} = \{\phi_n, \phi_c, \phi_s\}$ and $\mathcal{E}_{type} = \{Write, Belongs\ to\}$. ϕ_n, ϕ_c, and ϕ_s represent the node type of news article, subject, and creator respectively.

We present the schema of News-HIN based on the PolitiFact dataset in Figure 1(b), where the exact node and link types can be found intuitively.

IV. PROPOSED METHOD

Hierarchical Graph Attention Network (HGAT) follows a hierarchical attention structure including node-level attention and schema-level attention. The structure of HGAT is shown in Figure 2. The node-level attention is proposed to learn the weights of same-typed neighbors and aggregate them to get the type-specific neighbor representation. Then HGAT can learn the information of node types via schema-level attention and achieve the optimal weighted combination for the final fake news detection task. We will further discuss these components in this section.

A. Node-level attention

Node-level attention can learn the importance of neighbors belonging to the same type, respectively, for each news article node $n_i \in \mathcal{N}$. It then aggregates the representations of same-typed neighbors to form an integrated representation that we define as a schema node.

The inputs of the node-level attention layer are the initial feature vectors of nodes. Because multiple types of nodes exist in the News-HIN, the initial feature vectors belong to feature spaces with different dimensions. In order to enable the attention mechanism to output comparable and meaningful weights, we first utilize a type-specific transformation matrix to project features with different dimensions into the same feature space. We take the news article node $n_i \in \mathcal{N}$ as an example. The transformation matrix for type ϕ_n is $\mathbf{M}^{\phi_n} \in \mathbb{R}^{F \times F^{\phi_n}}$, where F^{ϕ_n} is the dimension of the initial feature $h_{n_i} \in \mathbb{R}^{F^{\phi_n}}$ and F is the dimension of the feature space mapped to. The F is the same for all type-specific transformation matrices. The projection process can be shown as follows:

$$h'_{n_i} = \mathbf{M}^{\phi_n} \cdot h_{n_i}; h'_{c_i} = \mathbf{M}^{\phi_c} \cdot h_{c_i}; h'_{s_i} = \mathbf{M}^{\phi_s} \cdot h_{s_i} \quad (1)$$

Through the type-specific projection operation, the feature space of nodes with different types can be unified. The node-level attention will learn the importance of same-typed neighbor nodes, respectively. In the face of detecting fake news, the target node is the news article node $n_i \in \mathcal{N}$, and the neighbors of it belong to $\mathcal{N} \cup \mathcal{S} \cup \mathcal{C}$. It should be noted that we also regard the target node itself as a neighbor node to cooperate with the self-attention mechanism. We let $T \in \{\mathcal{N}, \mathcal{S}, \mathcal{C}\}$ and nodes in T have the same type ϕ_t, then for n_i's neighbor nodes in T, the node-level attention can learn the importance $e_{ij}^{\phi_t}$ which means how important node $t_j \in T$ will be for n_i. The importance $e_{ij}^{\phi_t}$ can be formulated as follows:

$$e_{ij}^{\phi_t} = attention(h'_{n_i}, h'_{t_j}; \phi_t) \quad (2)$$

Here, *attention* denotes the same deep neural network as [17] conducting the node attention. *attention* is shared for all neighbor nodes with the same type ϕ_t. The masked attention keeps the network structure information. Only node $t_j \in neighbor_{n_i}$ being neighbors of node n_i with the type ϕ_t will be calculated and recorded as $e_{ij}^{\phi_t}$. Otherwise, the attention weight will be 0. We normalize them to get the weight coefficient $\alpha_{ij}^{\phi_t}$ via softmax function:

$$\alpha_{ij}^{\phi_t} = \text{softmax}_j(e_{ij}^{\phi_t}) = \frac{exp(e_{ij}^{\phi_t})}{\sum_{t_k \in neighbor_{n_i}} e_{ik}^{\phi_t}} \quad (3)$$

The schema node T_{n_i} can be aggregated as follows:

$$T_{n_i} = \sigma\Big(\sum_{t_j \in neighbor_{n_i}} \alpha_{ij}^{\phi_t} \cdot h'_{t_j}\Big) \quad (4)$$

A multi-head attention mechanism can be used to stabilize the learning process of self-attention in node-level attention. In details, K independent node-level attentions execute the transformation of Equation (4), and then the features achieved by K heads will be concatenated, resulting in the output representation of the schema node:

$$T_{n_i} = \Big\Vert_{k=1}^{K} \sigma\Big(\sum_{t_j \in neighbor_{n_i}} \alpha_{ij}^{\phi_t} \cdot h'_{t_j}\Big) \quad (5)$$

where $\Vert$ represents concatenation. In the problem we face, every target node n_i has 3 schema nodes $\mathcal{N}_{n_i}$, $\mathcal{C}_{n_i}$, $\mathcal{S}_{n_i}$ corresponding to 3 different types neighbors (include itself) based on the Definition 5.

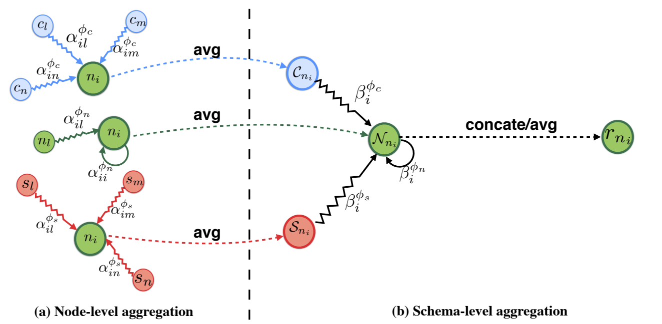


Figure 3. Explanation of aggregating process in both node-level and schema-level.

B. Schema-level attention

Through the node-level attention, we aggregate the neighbors of news article nodes as several schema nodes. Essentially it is equivalent to fusing information from same-typed neighbor nodes into the representation of a schema node. What we need to do in this stage is to learn the representation of news article nodes from all schema nodes. Different schema nodes contain type information, which requires us to differentiate the importance of node types. Here we introduce schema-level attention to automatically learn the importance of different schema nodes and use the learned coefficients for weighted fusion.

In order to obtain sufficient expressive power to calculate the attention weights between schema nodes as higher-level features, we apply one learnable linear transformation to the features of schema nodes from node-level attention. The linear transformation is parametrized by a weight matrix $\mathbf{W} \in \mathbb{R}^{F' \times KF}$. K is the number of heads in node-level attention. The schema-level attention mechanism $schema$ is a single-layer feedforward neural network applying the activating function Sigmoid with the dimension $2F'$. For the schema node T_{n_i}, the importance of it can be denoted as $w_i^{\phi_t}$:

$$w_i^{\phi_t} = schema(\mathbf{W}T_{n_i}, \mathbf{W}\mathcal{N}_{n_i}) \tag{6}$$

We normalize the importance of each schema nodes through a softmax function. Then the coefficients of the final fusion are denoted as $\beta_i^{\phi_t}$, which can be calculated as follows:

$$\beta_i^{\phi_t} = \text{softmax}_{\phi_i}(w_i^{\phi_t}) = \frac{\exp(w_i^{\phi_t})}{\sum_{\phi \in \mathcal{V}_T} \exp(w_i^{\phi})} \tag{7}$$

The learned coefficients can indicate the importance of different schema nodes to the final representation. Based on the coefficients, we can aggregate all schema nodes to get the final representation r_{n_i} of the target node n_i:

$$r_{n_i} = \sum_{\phi_t \in \mathcal{V}_T} \beta_i^{\phi_t} \cdot T_{n_i} \tag{8}$$

The set of learned final node representations is denoted as $\mathcal{R}$. Figure 3 describes the two-level aggregating for reference.

C. Loss Functions

Once achieving the final representation, we can use labeled news article nodes to train a classifier. In our experiments, a logistic regression layer is used to make predictions. We define the set of labeled news article nodes as $\mathcal{N}_l$. For the fake news detection tasks, our optimization objective function is set as a cross-entropy loss minimization, and it can be optimized through the backpropagation.

In the binary-class fake news detection, the loss is:

$$Loss(\mathcal{R}, \mathcal{N}_l) = -\sum_{n_i \in \mathcal{N}_l} \left((y_{n_i}\log(p_{n_i}) + (1-y_{n_i})\log(1-p_{n_i})\right) \tag{9}$$

Here, y is a binary indicator (0 or 1) indicating if the label is the correct classification for the news article node. p_{n_i} is the predicted probability of the representation of news article node n_i.

For the multi-class fake news detection, the cross-entropy based loss can be represented as:

$$Loss(\mathcal{R}, \mathcal{N}_l) = -\sum_{n_i \in \mathcal{N}_l} \sum_{j \in \mathcal{Y}} y_{n_i,j} \log(p_{n_i,j}) \tag{10}$$

where y is also a binary indicator (0 or 1) indicating whether class label j is the correct classification for the news article node n_i. A multi-class logistic regression layer will be trained to output the predicted probability $p_{n_i,j}$ of each class.

Table I
PROPERTIES OF THE HETEROGENEOUS NETWORKS

	PolitiFact Network		BuzzFeed Network	
# node	article	14,055	article	182
	creator	3,634	twitter user	15,257
	subject	152	publisher	9
# link	creator-article	14,055	publisher-article	182
	article-subject	48,756	article-twitter user	25,240

V. EXPERIMENTS

To test the effectiveness of HGAT, extensive experiments are designed and conducted on two real-world fake news datasets. Through the experimental results, we plan to evaluate our model by answering the following questions:

- **Question 1**: Can News-HIN and HGAT improve fake news detection performance?
- **Question 2**: Can the hierarchical attention handle the heterogeneity effectively?
- **Question 3**: Does HGAT possess generalizability so that it can work on different News-HINs and even promote to other applications on heterogeneous graphs?

A. Dataset Description

One of the datasets used in experiments is collected from the platform with fact-checking: *PolitiFact*, which is operated by Tampa Bay Times. Regarding news articles, *PolitiFact* provides the original contents, fact-checking results, and comprehensive fact-checking reports on the website. The platform categorizes them into different subjects based on contents and topics. A brief description of each subject is provided as well. The fact-checking results can indicate the credibility of corresponding news articles and take values from {True, Mostly True, Half True, Mostly False, False, Pants on Fire!}. We have established a News-HIN based on the original dataset following the descriptions in the previous sections. *BuzzFeed*[1] from Shu et al.[15]. *BuzzFeed* contains 91 real news articles and 91 fake news articles. We also construct a HIN based on *BuzzFeed* dataset following a similar way in Section III. There exist three types of nodes: article, twitter user, and publisher.

[1]https://github.com/KaiDMML/FakeNewsNet/tree/old-version

The key statistical data describing two HINs can be found in Table I.

B. Experimental Settings

1) Experimental Setup: For the *PolitiFact* dataset, we can acquire the set of news article nodes, which are the target nodes, to conduct the classification. The set of news article nodes are divided into 10 folds. Among them, 8 folds will be used as the training set, and 1 fold will be used as the validation set. The remaining 1 fold is left as the testing set. In order to conduct sufficient experiments with the setting of the different numbers of training data, we further make use of 2, 4, 6, 8 of 8 folds as the training set, respectively. In this way, experiments will be conducted with training ratios $\theta \in \{20\%, 40\%, 60\%, 80\%\}$, and the testing ratio is fixed as 10%. The fact-checking results corresponding to news articles are used as the ground truth for model learning and evaluation. In order to fit the non-sequential models, we have to transform the input features of each type of node into a vector with a fixed length. We use *TfidfVectorizer* in *Sklearn* package to extract features. The dimensions of news articles, creators, and subjects' initial features are 3000, 3109, and 191, respectively. We will not make use of comprehensive fact-checking reports in our experiments. We train models to work on both the multi-class classification task and the binary-class classification task. In the multi-class classification task, 6 kinds of different fack-checking results correspond to 6 classes. Meanwhile, in the binary-class classification, we group fact-checking results {Pants on fire, False, Mostly False} as a Fake class and group {True, Mostly True, Half True} as a Real class. Because our target is to detect fake news, we treat the Fake class as the positive class and the Real class as the negative class. In the binary-class classification task, we evaluate the results with Accuracy, Precision, Recall, and F1. Meanwhile, when the model works on the multi-class classification task, the performance is evaluated by Accuracy, Macro Precision, Macro Recall, and Macro F1, respectively.

For the *BuzzFeed* dataset, it has only two types of labels: True and fake, which can be used directly. In this way, we can only test the binary-class classification task models on *BuzzFeed*. The multi-class classification task will not be conducted on this dataset. The rest settings, including training&testing sets and initial features extraction, are the same as the *PolitiFact* dataset.

2) Comparison Methods:

Graph neural network methods

- **HAN** [20]: HAN employs node-level attention and semantic-level attention to capture the information from all meta-paths. In our experiments, we utilize two meta-paths (article-creator-article, article-subject-article) in HAN.
- **GAT** [17]: GAT is also an attention-based graph neural network for the node classification, but it is designed for homogeneous graphs. The News-HIN is treated as a homogeneous graph (ignore the type information) when testing the model.
- **GCN** [8]: GCN is a semi-supervised methods for homogeneous graphs. The News-HIN is treated as a homogeneous graph when testing it.
- **FakeDetector** [23]: FAKEDETECTOR is a deep diffusive network based fake news credibility inference model.

Text classification methods

- **SVM**: SVM is a classic supervised learning model. The feature vector used for building the SVM model is extracted merely based on the news article contents with TF-IDF.
- **LIWC** [9]: LIWC stands for Linguistic Inquiry and Word Count, which is widely used to extract the lexicons falling into psycho-linguistic categories. It learns a feature vector from a psychology and deception perspective.

Network embedding methods

- **Label Propagation (LP)** [25]: LP is merely based on the network structure. The prediction scores will be rounded and cast into labels.
- **DeepWalk** [11]: DeepWalk is a random walk based embedding method, which is designed to deal with the homogeneous network. Based on the embedding results, we then train a logistic regression model to perform the classification of news articles.
- **LINE** [16]: LINE optimizes the objective function that preserves the local and the global network structure simultaneously. We also learn a logistic regression model to conduct the classification based on the learned embeddings.

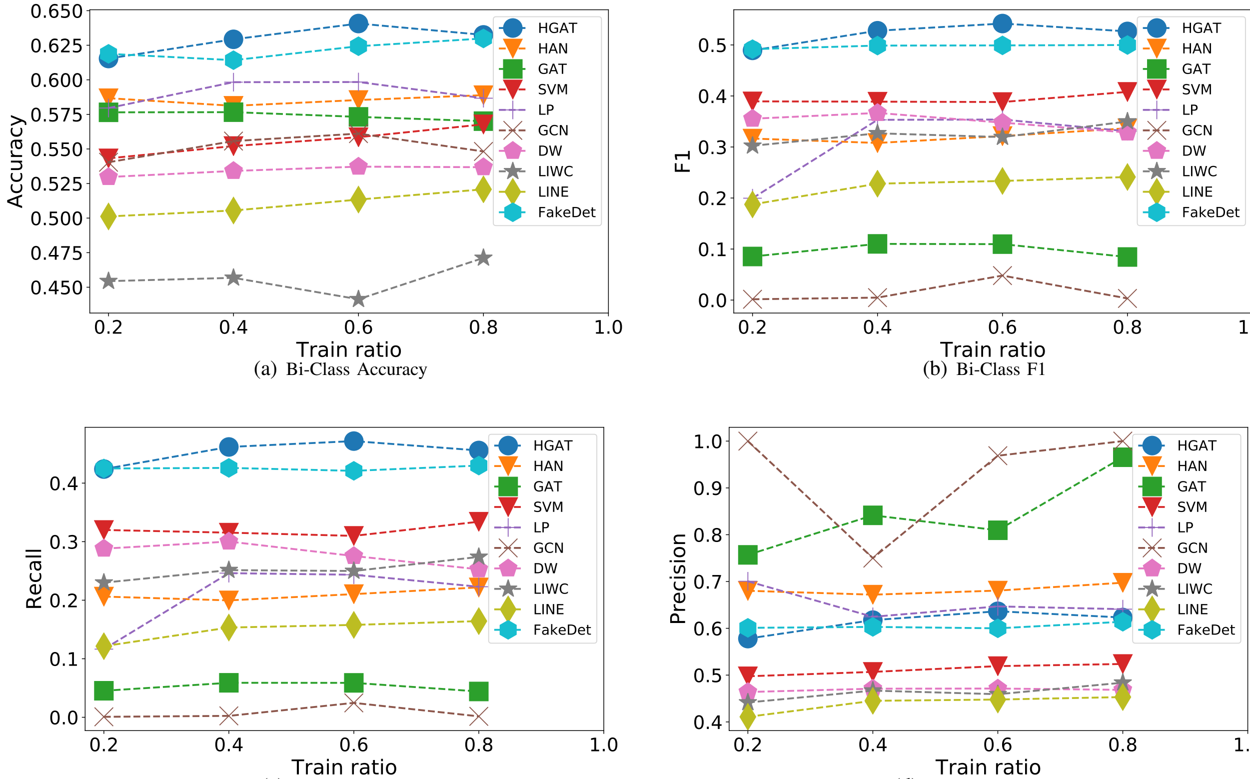
We have also noticed some recently appeared methods for fake news detection [3], [14], but did not compare them. The primary consideration is the difference between the scenarios we face. In [3], [14], they all utilize social context like user comments, but HGAT aims at detecting fake news in a relatively early stage. We will not utilize user comments about the news because when many users have started to discuss one fake news, the harmful influence of fake news has spread.

C. Reproducibility

The dimension of node-level representations is set as 12 and the dimension of schema-level is set as $(8 * K)$. Here, the attention head K is set as 1. For HAN, we set the dimension of node-level representations to 12 the same as HGAT, and the number of semantic-level hidden units is 8. For GAT, we set the embedding dimension as 12 and use just 1 attention head for a fair consideration. For GCN, the embedding dimension is set as 512. In the DeepWalk, we set the window size to 5, length of the random walk to 30, the number of walks per node to 10, and the embedding dimension to 128. We run the experiments on the Server with 3 GTX-1080 ti GPUs, and all codes are implemented in Python3. Code is available at: https://github.com/YuxiangRen/Hierarchical-Graph-Attention-Network.

D. Experimental Results with Analysis

1) Evaluate the effectiveness of News-HIN: To answer **Question 1**, we can analyze the experimental result from



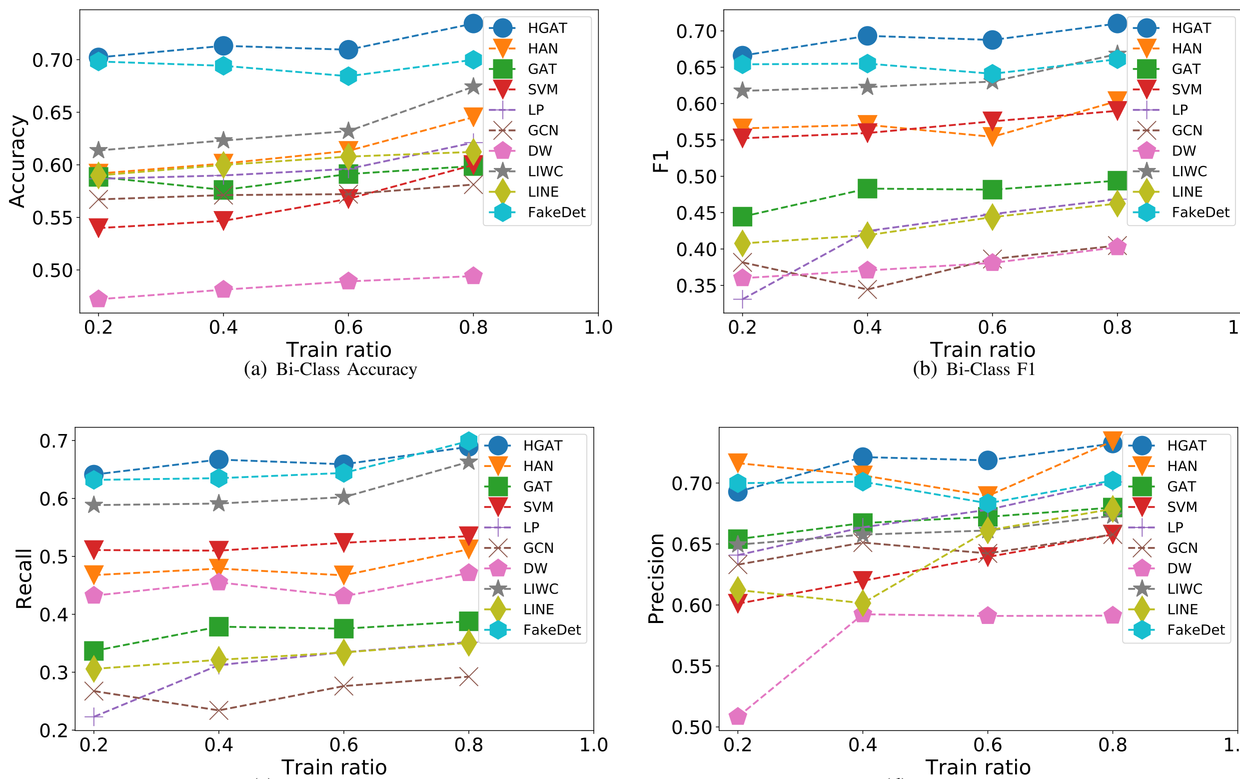
(a) Bi-Class Accuracy (b) Bi-Class F1

(c) Bi-Class Recall (d) Bi-Class Precision

Figure 4. The results of bi-class news articles classification on *PolitiFact* dataset

both binary-class and multi-class classification tasks. Based on results in Figure 4, our model HGAT achieves the best performance when focusing on Accuracy, F1, and Recall. Here we need to point out the reason for the inconsistency between the experimental results of FakeDetector shown in Figure 4 and the reported results in [23]. Because the fake news is set to positive class in our experiments, while the real news is set to positive in [23], the different settings lead to inconsistency. Our main task is to detect fake news, so we set fake news as positive to make the experimental results more intuitive. When considering precision, we can observe from Figure 4(d) that the performance of HGAT is lower than that of GCN and GAT. Through careful analysis, it can be found that GAT and GCN tend to judge most instances as 'Real' in the face of fake news detection, so the higher precision is related to very low Recall. In this case, higher precision is not practical because much fake news can not be detected. By comparing the performance between HGAT and network embedding methods, we can conclude that textual information is quite important, and merely utilizing graph structure is insufficient. Simultaneously, through the comparison between HGAT and text classification methods, we can find that graph structure is powerful for fake news detection. As a data structure that can model the graph structure and textual information simultaneously, news-HIN achieves better results. HGAT also shows an advantage over HAN, which is also proposed for heterogeneous graphs. More important. HGAT is a meta-path-free model without the limitation of handcrafted features. Figure 5, showing the results from the other dataset, also validates our conclusions. Next, we continue to analyze the multi-class classification task results to answer **Question 1** further. Due to the uncertainty of emerging news, it is often difficult to judge news directly as absolutely true and false. Besides, it is also not conducive to subsequent operations (e.g., final verification). Carrying out finer granularity multi-classification tasks according to the credibility of news is very meaningful. The experimental results from 6-labels classification are shown in Table II. Analyzing from the results, HGAT also achieves satisfactory performance. Compared with text classification methods and network embedding methods, HGAT outperforms them with an obvious advantage. On the one hand, this set of results illustrates the significance of News-HIN. On the other hand, this also shows that HGAT has a stronger learning ability in the heterogeneous graph, and the learned representation is also more comprehensive and discriminative. As we utilize a News-HIN as source data, the heterogeneity should be handle in an effective manner. We will evaluate the performance of HGAT in handling heterogeneity detailedly in the next section.

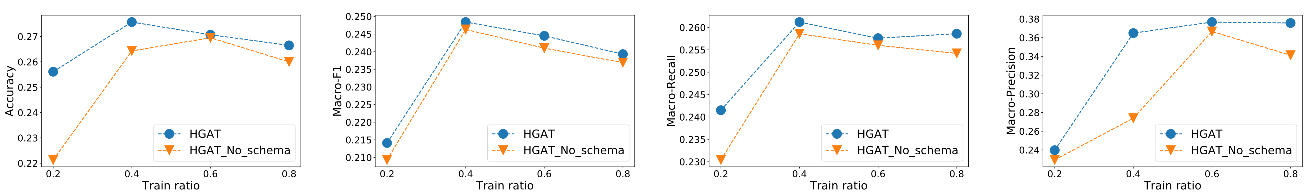
2) Access the performance in handling heterogeneity: In order to answer **Question 2**, we further compare the performance of GNN methods. From Figure 4 and Figure 5 to Table II, different tasks on two datasets verify that the heterogeneity of graphs should be dealt with in a more effective way. If we treat a heterogeneous graph as a homogeneous graph by ignoring the type (as we did in GCN and GAT), the result would be very disappointing. Compared with the method HAN proposed



(a) Bi-Class Accuracy (b) Bi-Class F1

(c) Bi-Class Recall (d) Bi-Class Precision

Figure 5. The results of bi-class news articles classification on *BuzzFeed* dataset



(a) Accuracy (b) Macro-F1 (c) Macro-Recall (d) Macro-Precision

Figure 6. The comparison of multi-class news article classification between HGAT and HGAT without schema-level attention on *PolitiFact* dataset

for heterogeneous graphs, HGAT also shows an advantage.

To further demonstrate the effectiveness of schema-level attention, we replace the schema-level attention of HGAT with fixed and equal weights for schema nodes. In experiments, all three schema nodes are assigned with the weight $1/3$, and we denote this comparison model as HGAT_No_schema. In Figure 6, we present the comparison results between HGAT and HGAT_No_schema. It is obvious that HGAT achieves better performance than HGAT_No_schema according to various metrics on two different datasets. This comparison verifies that the importance of schema nodes worth distinguishing, and HGAT can differentiate the importance through attention weights effectively. In contrast, the simple average operation harms the performance, which is equivalent to dropping the type information of schema nodes.

3) Verify the generalizability of HGAT: To answer **Question 3**, we design experiments for HGAT to test performance when facing different News-HINs. *PolitiFact* and *BuzzFeed* provide two different News-HINs. What's more, heterogeneous graph attention network [20] (HAN), as a general representation learning method proposed for heterogeneous graphs, is limited to handcrafted features (i.e., meta-path). HGAT can beat HAN without the limitation of any manual features, which fully illustrates the generalizability of HGAT. When facing other node classification-related applications on heterogeneous graphs, HGAT can be utilized as a general node representation learning method and be transferred without any obstacle. Although we only focus on fake news detection in this paper, HGAT is not restricted by the graph structure and is essentially a general method. Other potential applications of HGAT will be left for future discussion.

VI. Conclusion

In this paper, we study the HIN-based fake news detection problem and propose a novel graph neural network HGAT to solve it. HGAT employs a hierarchical attention mechanism considering both node-level and schema-level attention to learn the comprehensive representations of news article nodes. These discriminative representations can be used to detect fake news. Extensive experiments on two real-world News-

Table II
THE RESULTS OF MULTI-CLASS NEWS ARTICLES CLASSIFICATION ON *PolitiFact* DATASET

		Text Classification		Network Embedding			GNNs				
Train	**Metric**	SVM	LIWC	LP	DeepWalk	LINE	GCN	GAT	HAN	FakeDetector	HGAT
20%	Accuracy	0.1967	0.1432	0.2218	0.1932	0.1532	0.1986	0.2110	0.2181	0.2363	**0.2561**
	F1	0.1624	0.1225	0.1925	0.1562	0.0765	0.0711	0.1054	0.1234	**0.2204**	0.2141
	Recall	0.1801	0.0965	0.2153	0.1718	0.1433	0.1654	0.1975	0.1884	0.2307	**0.2415**
	Precision	0.1905	0.1409	0.2859	0.1742	0.0326	0.0674	0.1687	0.2467	**0.3123**	0.2397
40%	Accuracy	0.2042	0.1543	0.2278	0.1952	0.1567	0.1971	0.2237	0.2240	0.2399	**0.2757**
	F1	0.1775	0.1314	0.1944	0.1646	0.0798	0.0735	0.1103	0.1441	0.2238	**0.2484**
	Recall	0.1892	0.0987	0.2183	0.1742	0.1505	0.1697	0.1987	0.1853	0.2428	**0.2616**
	Precision	0.2047	0.1491	0.3037	0.1745	0.0401	0.0685	0.1815	0.2572	0.3421	**0.3649**
60%	Accuracy	0.2061	0.1513	0.2373	0.1969	0.1453	0.1927	0.2214	0.2256	0.2416	**0.2707**
	F1	0.1871	0.1321	0.2099	0.1647	0.0653	0.1023	0.1162	0.1475	0.2215	**0.2445**
	Recall	0.1976	0.1002	0.2222	0.1764	0.1410	0.1702	0.1954	0.1852	0.2401	**0.2576**
	Precision	0.2118	0.1561	0.2955	0.1966	0.0307	0.1053	0.1870	0.2792	0.3311	**0.3767**
80%	Accuracy	0.2186	0.1567	0.2407	0.2013	0.1623	0.1955	0.2212	0.2207	0.2576	**0.2665**
	F1	0.1962	0.1305	0.2187	0.1669	0.0875	0.1029	0.1037	0.1218	0.2221	**0.2393**
	Recall	0.2081	0.0954	0.2341	0.1830	0.1512	0.1940	0.1975	0.1840	0.2544	**0.2586**
	Precision	0.2233	0.1553	0.3149	0.1896	0.0468	0.0977	0.1819	0.2497	0.3510	**0.3757**

HIN demonstrate that HGAT has outstanding performance compared with the state-of-the-art methods. HGAT is also a general graph representation learning model that does not require any handcrafted features (e.g. meta-path) or other prior knowledge, so it is highly extensible for node classification-related problems on heterogeneous graph other than fake news detection.

REFERENCES

[1] Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. Journal of economic perspectives (2017)

[2] Ciampaglia, G.L., Shiralkar, P., Rocha, L.M., Bollen, J., Menczer, F., Flammini, A.: Computational fact checking from knowledge networks (2015)

[3] Cui, L., Shu, K., Wang, S., Lee, D., Liu, H.: defend: A system for explainable fake news detection. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (2019)

[4] Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (2017)

[5] Gupta, S., Thirukovalluru, R., Sinha, M., Mannarswamy., S.: Cimtdetect: A community infused matrix-tensor coupled factor- ization based method for fake news detection. In: arXiv:1809.05252 (2018)

[6] Hamilton, W., Bajaj, P., Zitnik, M., Jurafsky, D., Leskovec, J.: Embedding logical queries on knowledge graphs. In: Advances in Neural Information Processing Systems (2018)

[7] Jin, Z., Cao, J., Guo, H., Zhang, Y., Wang, Y., Luo, J.: Detection and analysis of 2016 us presidential election related rumors on twitter. In: International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation. Springer (2017)

[8] Kipf, T.N., Welling, M.: Semi-supervised classi cation with graph convolutional networks. In: ICLR (2017)

[9] Pennebaker, J.W., Boyd, R.L., Jordan, K., Blackburn, K.: The development and psychometric properties of liwc2015. Tech. rep. (2015)

[10] Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R.: Automatic detection of fake news. arXiv:1708.07104 (2017)

[11] Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (2014)

[12] Rubin, V.L., Lukoianova, T.: Truth and deception at the rhetorical structure level. Journal of the Association for Information Science and Technology (2015)

[13] Shi, C., Li, Y., Zhang, J., Sun, Y., Philip, S.Y.: A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering (2016)

[14] Shu, K., Wang, S., Liu, H.: Beyond news contents: The role of social context for fake news detection. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (2019)

[15] Shu, K., Wang, S., Liu, H.: Beyond news contents: The role of social context for fake news detection. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (2019)

[16] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web (2015)

[17] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph Attention Networks. ICLR (2018)

[18] Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (2016)

[19] Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., Yang, S.: Community preserving network embedding. In: Thirty-first AAAI conference on artificial intelligence (2017)

[20] Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The World Wide Web Conference (2019)

[21] Xie, S., Wang, G., Lin, S., Yu, P.S.: Review spam detection via temporal pattern discovery. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (2012)

[22] Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: ICLR (2019)

[23] Zhang, J., Dong, B., Philip, S.Y.: Fakedetector: Effective fake news detection with deep diffusive neural network. In: ICDE (2020)

[24] Zhou, X., Zafarani., R.: Fakenews: A survey of research, detection methods, and opportunities. In: arXiv:1812.00315 (2018)

[25] Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation (2002)