國立臺灣大學電機資訊學院暨中央研究院

資料科學學位學程

碩士論文

Data Science Degree Program

College of Electrical Engineering and Computer Science

National Taiwan University and Academia Sinica

Master Thesis

偵測錯誤健康新聞的階層式圖注意力網路

Hierarchical Graph Attention Network for Fake Health News Detection

孫珮文

Pei-Wen Sun

指導教授: 王志宇 博士、謝宏昀 博士

Advisor: Chih-Yu Wang Ph.D., Hung-Yun Hsieh Ph.D.

中華民國 111 年 1 月

January 2022

doi:10.6342/NTU202200334

誌謝

能夠完成這篇論文，首先我要感謝的是兩位指導教授王志宇老師和謝宏昀老師，在這兩年的研究生涯中，老師們在研究方向和方法上給了我許多建議與啟發，每次的討論都讓我受益匪淺。同時，也感謝王釧茹老師及蘇黎老師撥空擔任我的口試委員，並提供諸多寶貴的意見，使本篇論文更臻於完善。除了師長之外，我也要謝謝兩位研究所的同學慶豐和信之，和我一起修課、討論作業與實驗，想起來都是碩士班生活的美好回憶。特別要感謝我的家人，因為有你們的鼓勵與支持，我得以繼續追逐自己的人生目標。最後要謝謝實驗室的大家、學程的承辦人湘郁、以及所有曾經幫助過我的人們。

doi:10.6342/NTU202200334

中文摘要

隨著社群平台的興起，大量錯誤的醫療健康新聞流傳於網際網路上，當人們採取健康假訊息建議的偏方後，他們的生命可能會受到威脅。為了避免假新聞造成的負面影響，許多偵測的方法已被提出，例如，自然語言處理技術（NLP）能夠根據新聞的文字來判斷其真實性，然而由於當今人們時常從社群媒體接收新聞資訊，用戶的背景以及其對新聞的參與模式或許有助於假新聞的偵測，因此，研究學者引入圖神經網路（GNN）到此任務上。通常在一個社群網路中，每個節點對他相鄰的節點有不同的影響力，每種關係也有獨特的意義，有鑑於此，我們提出一個新穎、以階層式注意力機制為基礎的圖學習框架，以捕抓重要的節點和交互作用。另外，因為圖神經網路在多層堆疊時表現不佳，我們設計了兩階段的訓練策略，以縮短傳遞用戶交友圈之訊息到新聞節點的路徑。在辨別健康假新聞的任務上，實驗結果顯示我們的模型優於現有的方法，並且基於注意力機制的圖神經網路能受益於兩階段的訓練。

關鍵字：健康新聞、假新聞偵測、社群網路、圖形神經網路、注意力機制

doi:10.6342/NTU202200334

Abstract

With the rise of social media, massive fake health news is flooding over the Internet. When people take the treatments suggested by health misinformation, their lives may be at risk. To prevent the negative impacts caused by fake news, numerous detection methods have been proposed. For example, Natural Language Processing (NLP) techniques have been utilized to debunk fake news based on the content in the story. However, since nowadays people often receive information from social media, the background of platform users and their engagement with news may be useful for news verification. Researchers thus introduce Graph Neural Networks (GNNs) to this task. Generally in a social network, each node has different influences on its adjacent nodes and each relationship represents a unique meaning. As a result, we propose a novel graph learning framework based on hierarchical attention to capture the important nodes and interactions. Furthermore, GNNs often perform poorly when stacking with multiple layers, so we design the two-stage training strategy to shorten the paths of delivering user network information to news nodes.

doi:10.6342/NTU202200334

The experimental results show that our model outperforms the existing methods in the task of health misinformation detection and attention-based GNNs can be benefited from the two-stage training.

Keywords: Health News, Fake News Detection, Social Networks, Graph Neural Networks, Attention Mechanism

doi:10.6342/NTU202200334

Contents

doi:10.6342/NTU202200334

doi:10.6342/NTU202200334

List of Figures

doi:10.6342/NTU202200334

List of Tables

doi:10.6342/NTU202200334

Chapter 1 Introduction

1.1 Motivation

Online fake news can lead to undesired consequences in many areas, such as in political and healthcare domains. For example, during a politician election, numerous malicious rumors are spreading over the Internet to defame an opponent's reputation and influence voters' decisions [2]. At the outbreak of COVID-19, virus misinformation and the unproven cures posted on social media have caused hundreds of deaths [1]. Among the spread of fake news, we are more interested in detecting misinformation in the healthcare domain as it is directly related to human living and may cause irreversible consequences.

To date, multiple websites have devoted efforts to debunk fake news. For example, fact-checking websites such as *FactCheck*[1] and *HealthNewsReview*[2] rely on editors manually checking the authenticity of news stories. Such an approach may not catch up with the growing speed of fake news today. To scale up with the increasing amount of misinformation, automated fact-checking methods are in demand. A straightforward solution is applying Natural Language Processing (NLP) techniques to detect fake news based on news content, such as headline, body text, and other news features. Another approach

[1] https://www.factcheck.org/

[2] https://www.healthnewsreview.org/

for identifying fake news is through analyzing the information propagation pattern on the social network. For instance, users with domain knowledge in medicine are more likely to share the correct healthcare stories and stop the propagation of health misinformation. Also, people connected on social media often behave in some similarity, such as having the same political stance or similar characteristics. As a result, existing works [15, 4, 6] have utilized Graph Neural Networks (GNNs) to capture user information and the social engagements of news articles to determine news authenticity.

However, current graph-based methods for fake news detection have some drawbacks. For example, FANG [15] models two homogeneous sub-graphs, thus omitting various types of social entities and interactions. Though SAFER [4] considers the heterogeneity in the social context, it simply averages the representations of all engaged users when learning the contextual embedding of each news article. Hetero-SCAN [6] applies attention weights to meta-paths, but this approach requires domain knowledge to design the effective meta-paths and needs a large amount of memory space for storing all instances of meta-paths.

To overcome the above limitations, we propose a novel graph learning framework that captures the rich interactions in a heterogeneous graph, considers the different influence of the connected entities, and does not rely on pre-defined meta-paths. We achieve these goals by designing a hierarchical attention mechanism that automatically learns the important social entities and relationships.

Furthermore, GNNs are susceptible to the general problems of training deep neural networks [4], so they often contain a small number of layers [3]. To model GNNs with the complete structure of social context, we propose a novel training technique that divides

doi:10.6342/NTU202200334

end-to-end learning into two stages.

1.2 Contribution

Our main contributions can be listed as follows:

- We improve the first engagement-driven fake health news repository for future research to work on a more complete dataset. With this updated dataset, we perform exploratory analyses and report the useful features for identifying health misinformation.
- We propose the Hierarchical Graph Attention Network (HiGAT) that effectively captures the critical nodes and engagements, thus improving the detection results.
- We propose two-stage training for graph entities to receive messages from their distant important nodes, and validate that this training technique helps boost the performance of attention-based GNNs.

1.3 Organization of Thesis

In Chapter 2, we provide the background information about Graph Neural Networks, and review some popular GNNs and the current methods for fake news detection. In Chapter 3, we describe and refine the dataset that will be used in this study, and then perform feature engineering and selection for the prediction task. In Chapter 4, we describe the proposed framework and training technique in detail. In Chapter 5, we conduct experiments to evaluate our approach to fake health news detection. Finally, we conclude and identify future work in Chapter 6.

doi:10.6342/NTU202200334

Chapter 2 Preliminary and Related Work

We first define some important terminologies related to graphs. Then we review several famous Graph Neural Networks (GNNs) and the existing works for debunking misinformation.

2.1 Preliminary

2.1.1 Heterogeneous Graph

A heterogeneous graph is a graph that contains multiple node types or edge types. The formal definition is shown as follows.

Definition 2.1.1 (Heterogeneous Graph)**.** Heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$ is a graph that consists of node set $\mathcal{V}$, edge set $\mathcal{E}$, and two mapping functions $\phi : \mathcal{V} \to \mathcal{A}$ and $\psi : \mathcal{E} \to \mathcal{R}$ that map nodes and edges to node types $\mathcal{A}$ and edge types $\mathcal{R}$, respectively.

2.1.2 Graph Neural Network

Graph Neural Network (GNN) is a class of neural networks that directly operates on the graph structure. Generally, a GNN inputs a graph and updates node features via message passing [9]. In this procedure, neighborhood information is aggregated to the target node and forms the contextual node representation.

Definition 2.1.2 (Graph Neural Network)**.** Suppose node v has node representation $[\mathbf{h}_v]^l$ at the (l)-th GNN layer. It is updated from the (l-1)-th layer as shown below:

$$[\mathbf{h}_v]^l = \text{Update}\left([\mathbf{h}_v]^{l-1},\ \underset{\forall s \in \mathcal{N}_v}{\text{Aggregate}}\left([\mathbf{h}_s]^{l-1}\right)\right)$$

where $\mathcal{N}_v$ denotes the local neighborhood of node v. Aggregate$(\cdot)$ gathers all source nodes' information via an operator, such as $mean$, sum, max, or customized pooling function. Update$(\cdot)$ combines neighborhood messages with target node itself.

2.2 Related Work

2.2.1 Graph Neural Networks

Multiple homogeneous GNNs have been designed with the Definition 2.1.2. Graph Convolutional Network (GCN) [12] is the first proposed method that generalizes convolutional filters on graph structure to gather information from the first-order neighbors. To improve the efficiency, GraphSAGE [10] samples local neighborhoods and aggregate their features via various operators, such as mean, LSTM, or one-layer neural network followed by max pooling. Later on, Graph Attention Network (GAT) [21] adopts the self-attention

doi:10.6342/NTU202200334

mechanism [20] to aggregate nearby nodes with different weights.

2.2.2 Heterogeneous GNNs

As the above GNNs are designed for homogeneous graphs, they do not consider the rich and complex interactions among nodes. Therefore, new approaches tailored to heterogeneous graphs are then proposed. Relational Graph Convolutional Network (R-GCN) [19] extends GCN by learning a distinct weight matrix for each edge type. Heterogeneous Graph Attention Network (HAN) [23] applies attention mechanism on the heterogeneous graph by maintaining importance for each meta-path, and Metapath Aggregated Graph Neural Network (MAGNN) [8] further incorporates all intermediate node features along the meta-path. Heterogeneous Graph Transformer (HGT) [11] adopts Transformer-like attention architecture with relative temporal encoding and mini-batch graph sampling techniques.

2.2.3 Fake News Detection Methods

The automatic fact-checking methods can be classified into two categories: text-based and graph-based.

2.2.3.1 Text-based Approach

These methods detect fake news based on features obtained from news, such as body text, news title, and writing styles [17, 16]. Multi-modal approaches [24, 28] further incorporate both textual and visual features for news verification. For the task of detecting misinformation about COVID-19, Mazzeo et al. [13] input news contents and URLs to a

doi:10.6342/NTU202200334

machine learning classifier, and Wang et al. [25] fine-tuned BERT with BiLSTM or CNN layers.

2.2.3.2 Graph-based Approach

The graph-based approach employs users' information obtained from social media (e.g. user profiles, user posts and responses, and network structure) to determine the veracity of news articles. FANG [15] models news-source and user homogeneous subgraphs separately to maximize the similarity of connected nodes, and considers user stance (sentiment) to an article. SAFER [4] generates the embeddings of news and users independently. Each news node's textual feature is then concatenated with the average representation of its interacting users and passed to a logistic regression classifier. Hetero-SCAN [6] utilizes meta-path aggregation techniques to capture important user engagements for news pieces.

2.2.3.3 Summary

Graph-based approaches consider user engagements for each news story, so they are supposed to have better performance than text-based approaches on fake news detection. However, the existing graph-based methods do not learn meaningful representation for news articles because they ignore the heterogeneity of the multiple types of nodes and connections, treat a node neighborhood with equal influence, or do not fit into a graph with millions of edges. Our approach is able to learn the important components in a complex social context and save memory space as well as human efforts.

doi:10.6342/NTU202200334

Chapter 3 Dataset and Feature Engineering

3.1 Dataset

3.1.1 Overview of Dataset

Since we aim to detect fake health news with the help of social networks, FakeHealth [7] data repository is chosen as the target of this work. This repository includes news content, reviews with ratings, news dissemination on Twitter, and user networks. News is collected from *HealthNewsReview*, a fact-checking website for healthcare news, where experts evaluate the authenticity of news articles based on ten criteria and give each a rating from 0 to 5. Similar to the authors of the data repository, a news piece is considered fake for a score less than 3 and real otherwise.

Though news and review contents are publicly available, social engagements and user information are not released due to Twitter privacy policies. Instead, the authors provide unique identifiers of Tweet objects and Twitter APIs, allowing us to retrieve the content of Tweets and user information by ourselves.

The repository is split into two datasets based on news sources. HealthStory contains articles published by news media, and HealthRelease contains news articles released from various institutes such as universities and research centers. For the detailed statistics of two datasets, HealthStory includes 1,690 news articles, 384k tweets, 120k retweets, 27k replies, and 241k users, while HealthRelease gathers 606 news pieces, 47k tweets, 16k retweets, 1.5k replies, and 30k users. Since HealthStory contains much more data than HealthRelease, it is the dataset we used in this study.

3.1.2 Dataset Refinement

Though FakeHealth authors provide news content as well as rich social engagements for fake health news detection, their data collection is not perfect and leaves room for improvement.

3.1.2.1 News

Published time is a basic component for a news piece. Nevertheless, approximately 35 percent of the published time is empty. To handle this issue, we first search for any time-related keyword in the news files. If a key is matched, we would obtain a date, date and time, or timestamp. Otherwise, we use regular expressions to extract a date from the news URL, top image's URL, or news content. If time information is not included in the dataset, we manually check the release time of news from its website. After each published time is obtained, we convert multiple formats of time into a uniform format.

It is observed that some news articles have less than 100 words, which seems to be too short to form a news story. After examining each short content, we found that it is either a news title, a crawl error message, or a snippet of news. As a result, we re-scraped

doi:10.6342/NTU202200334

news pieces with less than 100 words using the Beautiful Soup library to update news content.

Last, we exclude news articles that are not disseminated on Twitter, not applicable to any review criterion, or have become unavailable over time. In total, there are 1,548 news pieces (28% fake and 72% real articles) in the refined dataset.

3.1.2.2 Tweets

HealthStory authors choose the Twython library as the wrapper for Twitter APIs, but it is limited to scraping at most 140 characters of each Tweet. For those truncated Tweets, we use another parser library called Tweepy to obtain their full texts. The incomplete Tweets are thus updated. We further remove retweets and replies whose original tweets are not in the dataset. The statistics of the improved dataset are shown in Table 3.1.

Table 3.1: The statistics of the refined HealthStory dataset.

Nodes		Edges		Edges per node	
News	1,548	Citation	304,090	Tweet citations / news	196.44
Fake news	437	Post	304,090	Tweet posts / user	1.76
Real news	1,111	Retweet	83,204	Retweets / user	0.48
Tweet	303,330	Reply	5,941	Replies / user	0.03
User	172,317	Following	11,125,651	Followers / user	64.57

3.2 Feature Engineering

Before utilizing graph structure, it is necessary to initialize node features for each node type. Therefore, we process the possible node features and then exclude features that are likely unrelated to fake health news detection to reduce feature dimensions.

3.2.1 Feature Processing

First, we select the following features of each node type and process them.

- **News**: content, news published time, news media, news title length, content length, and the count of numeric values in a news piece;
- **Tweet**: content, tweet posting time, devices that post tweets, the number of times this tweet has been retweeted, the number of likes the tweet has received;
- **User**: user profile, account creation time, the number of followers, the number of following people, the number of tweets the user has posted, the number of likes the user has given.

3.2.1.1 Text

To vectorize sentences, we apply three pre-trained Transformer-based language models according to the types of texts. Specifically, news contents are encoded by BigBird [27] which is capable to handle long text, tweets are passed to BERTweet [14] which was trained with 850M English tweets, and user profiles are vectorized by Sentence-BERT [18] that is designed to create general sentence embeddings. The output of Transformer is then combined with a mean pooling operation to form a fixed-sized text representation.

3.2.1.2 Date and time

News published time Since we want to build a model capable of verifying the authenticity of future news, we do not consider news release time as a feature but rather a criterion to split our dataset into training, validation, and testing sets.

doi:10.6342/NTU202200334

Tweet posting time, and user account creation time We convert the posting time of tweets into the year, month, and day, while the creation time of user accounts is converted to the number of years or months since Twitter was founded (March 2006).

3.2.1.3 Category

News media We merge news media with less than ten instances into the same category, leaving 25 kinds of news media, such as HealthDay, Reuters, WebMD, and The New York Times. Categories are then encoded to numbers with one-hot encoding.

Twitter devices We extract devices used to post Tweets with regular expression. Since there are thousands of Twitter device types, we classify them into 9 main categories: iPhone, Android, other mobile operating systems, tablet, Mac, Windows, bot, website, and others. The main Twitter devices are also handled by one-hot encoding.

3.2.1.4 Number

News title length, content length, and the count of numeric values Review criteria suggest that the real health news would analyze the benefits and harms of the new treatment, include enough details about the experiment, and so on. Based on this scoring guide, we hypothesize that real news would contain more words and numbers than fake news. Therefore, we calculate the number of words and numeric values in a news article and consider headline length as well. Their values are normalized into a range between 0 to 1.

The number of retweets/likes the tweet has received, and the number of followers/following people/tweets/likes for a user The numeric features of tweets and users

doi:10.6342/NTU202200334

have very wide and extreme distributions: more than half of values are zeros, but the maximum can reach thousands or millions. Hence, we calculate the logarithm of them and compare their magnitudes only.

3.2.2 Feature Selection

In addition to text embedding, we want to include non-textual features that are useful for machines to classify news articles. As a result, we visualize the relationship between each feature and the class labels. However, since tweets and users are not labeled, we calculate their ratings by averaging the ratings of news stories that a tweet has cited or a user has engaged. Same as the classification standard applied to news, tweet/user with average ratings less than 3 is classified as misleading tweet/rumor spreader, while the other class is truthful tweet/regular user.

3.2.2.1 Date and time

From Figure 3.1, we find that there is no significant difference between the proportions of truthful and misleading tweets in terms of years, months, or days ($p > 0.001$, chi-square test). Also, the elapsed years and months of user accounts are not related to user credibility ($p > 0.001$, chi-square test) in Figure 3.2. As a result, we exclude the temporal information of tweets and users from node features.

3.2.2.2 Category

The distributions of the average ratings among different news media and Twitter devices are illustrated in Figure 3.3. It is observed that the average ratings of news media

doi:10.6342/NTU202200334

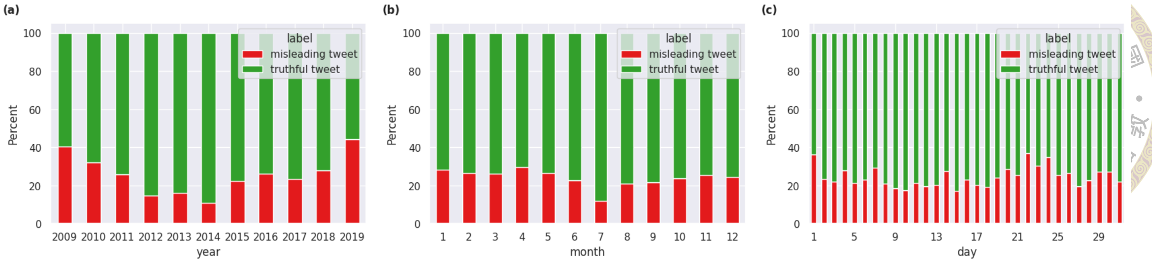

Figure 3.1: The years (a), months (b), and days (c) when misleading tweets and truthful tweets are posted.

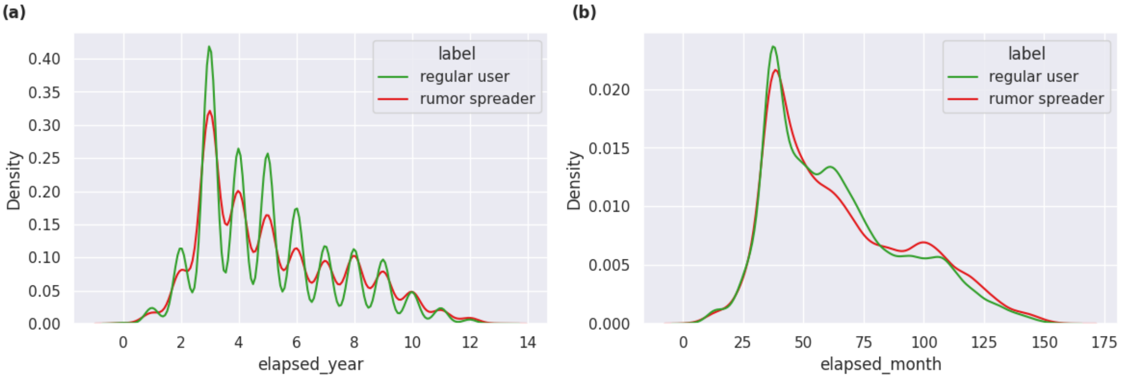

Figure 3.2: The elapsed years (a) and elapsed months (b) starting from the foundation of Twitter to the account creation for rumor spreaders and regular users.

differ a lot, which is shown in Figure 3.3 (a). Also, the 95% confidence intervals of the average ratings for news media do not have many overlaps. These observations indicate that news media can be a useful feature for detecting fake news. However, in Figure 3.3 (b), the average ratings of main Twitter devices range from 3.2 to 3.5 and most lower limits of the intervals are above 3, guiding models to assign all the tweets to the truthful class. Therefore, Twitter devices will not be chosen as a feature for the downstream task.

3.2.2.3 Number

Now we compare the distributions of numeric features for fake news and real news. From Figure 3.4 (a), we find that two classes of news articles have similar title lengths ($p > 0.001$, Kolmogorov-Smirnov test). On the contrary, the number of words or numeric values in a piece of real news is significantly higher than the fake ($p < 0.001$,

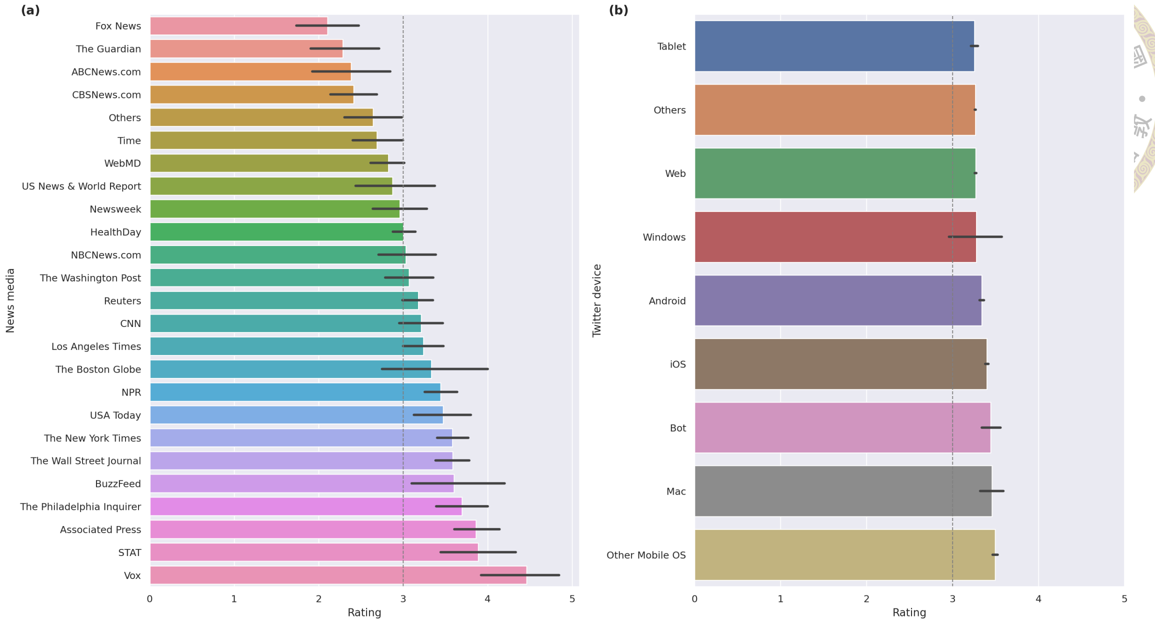

Figure 3.3: The average ratings of news media (a) and Twitter devices (b). Error bars represent the 95% confidence interval of the means.

Kolmogorov-Smirnov test). Their distributions are shown in Figure 3.4 (b) and 3.4 (c). Thus, news with few words or numeric values can be an indicator of fake news.

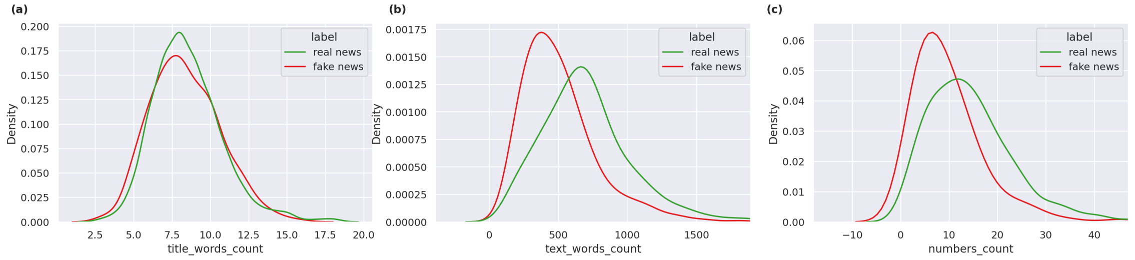

Figure 3.4: The count of title words (a), content words (b), and numeric values (c) of fake news and real news.

As for the numeric features of tweets, we observe that the distributions of retweets log-count and likes log-count almost overlap in two classes ($p > 0.001$, Kolmogorov-Smirnov test), which is displayed in Figure 3.5. It indicates that it is impossible to detect misleading tweets based on the number of times the tweet has been retweeted or liked.

Last, we investigate whether any user numeric features can help machines debunk misleading news. However, there is no significant difference in every log-count distribu-

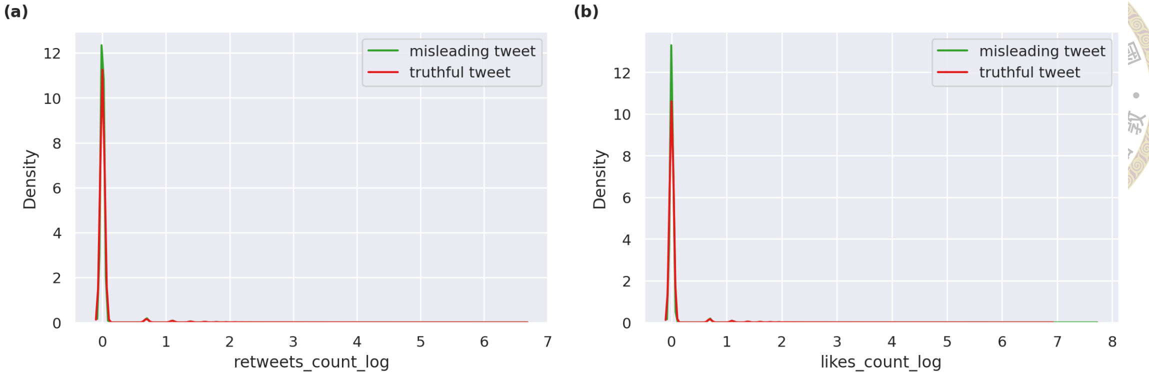

Figure 3.5: The log-count of retweets (a) and likes (b) given to misleading tweets and truthful tweets.

tion ($p > 0.001$, Kolmogorov-Smirnov test) in Figure 3.6. In other words, rumor spreaders and regular users have similar distributions in the numbers of followers, following people, tweet posts, and likes given to others. As a result, the numeric features of users are all excluded.

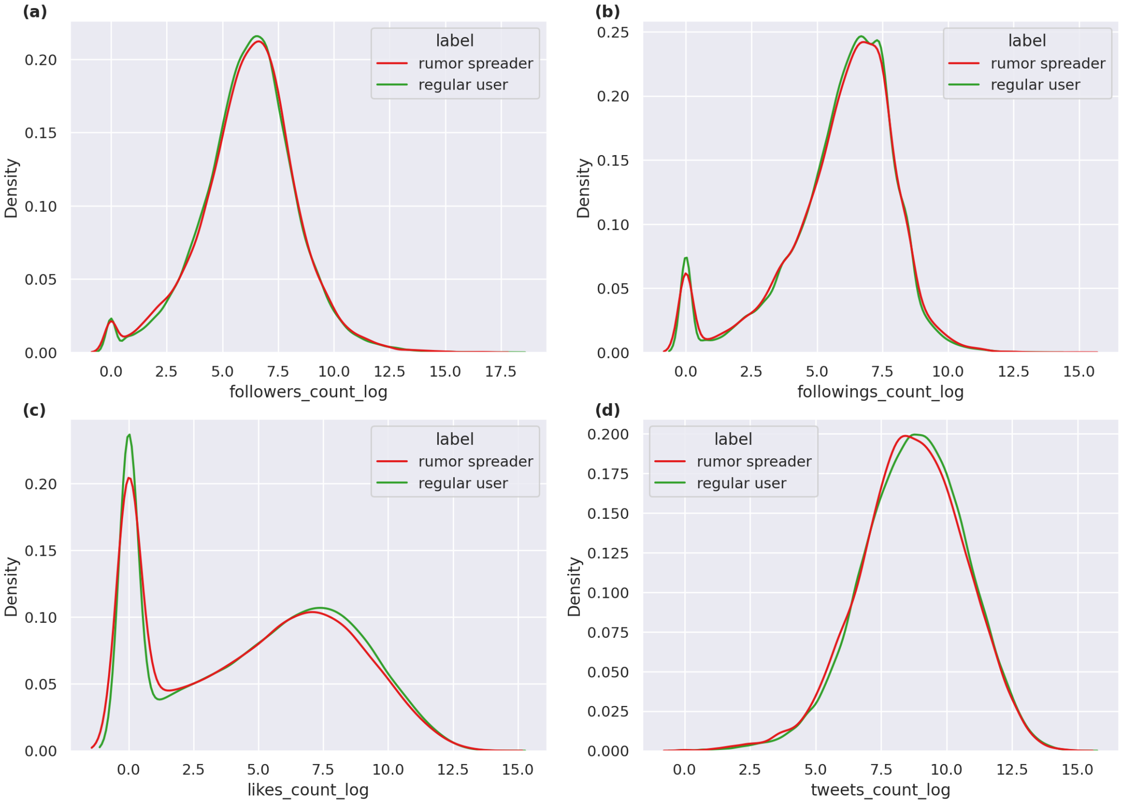

Figure 3.6: The log-count of followers (a), following people (b), tweets (c), and likes (d) of rumor spreaders and regular users.

doi:10.6342/NTU202200334

3.2.2.4 Summary

To summarize, we select news media as well as the counts of news words and numeric values as the non-textual node features. They will be concatenated with text embedding and input to models for fake news detection.

doi:10.6342/NTU202200334

Chapter 4 Methodology

In this chapter, we construct our graph and define the problem of fake health news detection. After that, we describe our learning framework and training technique.

4.1 Graph Construction

To gather the social context of health news, we construct a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $\mathcal{V}$ is a set of the nodes in graph and $\mathcal{E}$ is a set of edges. $\mathcal{A}$ contains node types in $\{news, tweet, user\}$ and $\mathcal{R}$ includes edge types in $\{citation, post, retweet,$ $reply, following,$ and their reversed edges$\}$. We display the graph of social context in Figure 4.1 and report the statistics of graph components in Table 3.1. In total, there are 477,195 nodes and 23,645,952 edges.

Compared with other studies using the same dataset, the graph in SAFER consists of news and user nodes, and the graph in Hetero-SCAN contains three node types (news source, news, and user). Besides, these two graphs contain much fewer user nodes than our graph. Specifically, SAFER authors select 20k most active users and exclude users who have engaged in both real and fake news, Hetero-SCAN authors randomly sample 63k users, and we include 172k users.

4.2 Problem Definition

After the graph is created, we can now define our task.

Definition 4.2.1 (Graph-based Fake Health News Detection)**.** Given a graph $\mathcal{G}$ of social context, fake health news detection is a binary classification problem that predicts whether health news x is fake or real. In other words, there exists a prediction function $F : x \rightarrow \{0, 1\}$ such that

$$F(x) = \begin{cases} 1, & \text{if } x \text{ is a piece of fake health news} \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

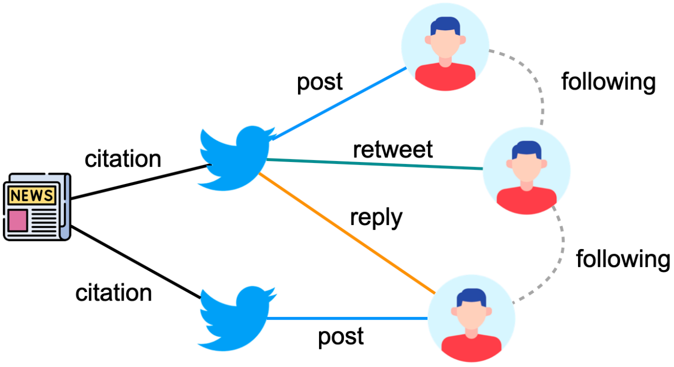

Figure 4.1: An example of the social context grpah.

4.3 Model Architecture

Hierarchical Graph Attention Network (HiGAT) takes a social-context graph with the selected node features as input and outputs the probabilities of two news classes. Figure 4.2 presents the framework of HiGAT. For example, target node v has five neighboring

doi:10.6342/NTU202200334

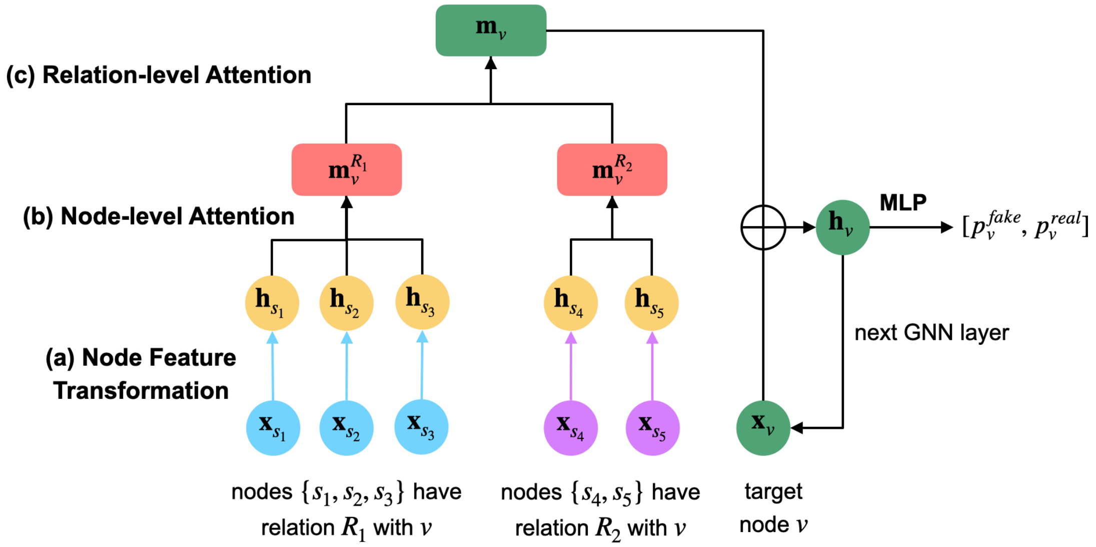
(c) Relation-level Attention
$\mathbf{m}_v$
$\mathbf{m}_v^{R_1}$
$\mathbf{m}_v^{R_2}$
(b) Node-level Attention
$\mathbf{h}_{s_1}$ $\mathbf{h}_{s_2}$ $\mathbf{h}_{s_3}$ $\mathbf{h}_{s_4}$ $\mathbf{h}_{s_5}$
(a) Node Feature Transformation
$\mathbf{x}_{s_1}$ $\mathbf{x}_{s_2}$ $\mathbf{x}_{s_3}$ $\mathbf{x}_{s_4}$ $\mathbf{x}_{s_5}$
$\mathbf{h}_v$
MLP
$[p_v^{fake}, p_v^{real}]$
next GNN layer
$\mathbf{x}_v$
nodes $\{s_1, s_2, s_3\}$ have relation R_1 with v
nodes $\{s_4, s_5\}$ have relation R_2 with v
target node v

Figure 4.2: The overall architecture of HiGAT.

nodes $\{s_1, s_2, s_3, s_4, s_5\}$ which have 2 types of relationships (colored in blue and magenta) with node v. The neighborhood messages are aggregated in three steps: node feature transformation, node-level attention, and relation-level attention. We then concatenate node v's features with the message from its neighborhood and pass the updated embedding to multi-layer perceptron (MLP).

4.3.1 Node Feature Transformation

In a heterogeneous graph, initial features of different node types may distribute in different vector spaces since multiple feature engineering techniques are used. To project them in the same latent vector space, we apply a linear transformation for each type of nodes followed by an activation function. Given a node v of node type $A \in \mathcal{A}$, we conduct:

$$\mathbf{h}_v = \sigma\left(\mathbf{W}_A \cdot \mathbf{x}_v + \mathbf{b}_A\right) \tag{4.2}$$

doi:10.6342/NTU202200334

where $\mathbf{x}_v \in \mathbb{R}^{d_A}$ is the initial feature vector of node v, and $\mathbf{h}_v \in \mathbb{R}^{d_t}$ denotes its transformed feature. $\mathbf{W}_A \in \mathbb{R}^{d_t \times d_A}$ is the projection matrix for type A's nodes, $\mathbf{b}_A \in \mathbb{R}^{d_t}$ is the corresponding bias vector, and $\sigma(\cdot)$ is a non-linear activation function.

After this operation, feature vectors of different node types lie in the same latent space, facilitating the aggregation process in a heterogeneous graph.

4.3.2 Node-level Attention

Before aggregating any information from neighboring nodes, we should notice that nodes in the same relation also show different importance to their connected nodes. For instance, a health news tweet shared by a chief medical advisor has much more influence than being shared by an ordinary person. Therefore, we adopt self-attention [20] to learn the weights of nodes in the same type. Given a source node s connecting to a target node v via relation $R \in \mathcal{R}$, the importance of node s for node v is formulated as follows:

$$w_{vs}^{R} = \text{LeakyReLU}\left(\mathbf{a}_R^{\top} \cdot [\mathbf{W}\mathbf{h}_v \| \mathbf{W}\mathbf{h}_s]\right) \tag{4.3}$$

Here $\mathbf{W} \in \mathbb{R}^{d \times d_t}$ is a learnable matrix that transform input features into hidden dimension d, $\mathbf{a}_R \in \mathbb{R}^{2d}$ is the node-level attention vector for relation R, and $\|$ represents the operation for vector concatenation. Finally, LeakyReLU (with a negative slope of 0.2) is applied to output importance w_{vs}^{R}.

The above Eq. (4.3) shows that given a relation, a neighbor's importance depends on the features of two connected nodes. Also, the attention weight is asymmetric ($w_{vs}^{R} \neq w_{sv}^{R}$), indicating that the importance of node s to node v is irrelevant to the importance of node v to node s.

Now we consider graph structure in the mechanism by only computing w_{vs}^R for nodes $s \in \mathcal{N}_v^R$, where $\mathcal{N}_v^R$ is a set of neighbors of v (not including itself) that connects to v by relation R. Once the importance of every $s \in \mathcal{N}_v^R$ is obtained, we normalize them across all choices of neighbors $\mathcal{N}_v^R$ using softmax function:

$$\alpha_{vs}^R = \text{softmax}_s\left(w_{vs}^R\right) = \frac{\exp\left(w_{vs}^R\right)}{\sum_{u \in \mathcal{N}_v^R} \exp\left(w_{vu}^R\right)} \tag{4.4}$$

Then, the normalized attention weights and their corresponding features are used to form a linear combination. Last, we add an activation function $\sigma(\cdot)$ to output the relation-specific embedding $\mathbf{m}_v^R \in \mathbb{R}^d$:

$$\mathbf{m}_v^R = \sigma\left(\sum_{s \in \mathcal{N}_v^R} \alpha_{vs}^R \cdot \mathbf{W}\mathbf{h}_s\right) \tag{4.5}$$

To stabilize the learning process, we extend the above mechanism to multiple heads. That is, we execute K times of node-level attention mechanisms independently and concatenate the outputs as the relation-specific embedding $\mathbf{m}_v^R \in \mathbb{R}^{d \cdot K}$:

$$\mathbf{m}_v^R = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{s \in \mathcal{N}_v^R} [\alpha_{vs}^R]^k \cdot \mathbf{W}^k\mathbf{h}_s\right) \tag{4.6}$$

where $\Vert$ represents concatenation, while $[\alpha_{vs}^R]^k$ and $\mathbf{W}^k$are the normalize importance and weight matrix at the k-th node-level attention.

Given a set of edge types $\{R_1, ..., R_n\}$ incoming to node v, after feeding neighborhood information into their corresponding node-level attention layers, we obtain n relation-specific node embeddings, denoted as $\{\mathbf{m}_v^{R_1}, ..., \mathbf{m}_v^{R_n}\}$.

4.3.3 Relation-level Attention

Generally, there are multiple interaction types among social entities in a heterogeneous graph, and relation-specific node embedding can only represent one aspect. To obtain a more comprehensive node representation, we need to combine information from all relations. The simplest way is averaging all relation-specific embeddings. Here, we extend the mean operation by learning the relative importance of each relation.

Given an edge type R_i incoming to node $v \in \mathcal{V}_A$, we project the relation-specific node embedding $\mathbf{m}_v^{R_i}$ with a nonlinear transformation. The importance of relation R_i for node v, denoted as $s_v^{R_i}$, is learned by the similarity between the transformed embedding and a relation-level attention vector $\mathbf{q}_A \in \mathbb{R}^q$. Then, a set of importance values $\{s_v^{R_i},\ \forall v \in \mathcal{V}_A\}$ is averaged to form the importance of relation R_i for type A's nodes. The procedure is shown as follows:

$$\begin{aligned} s_v^{R_i} &= \mathbf{q}_A^\top \cdot \tanh\left(\mathbf{W}_A \cdot \mathbf{m}_v^{R_i} + \mathbf{b}_A\right) \\ w_{R_i} &= \frac{1}{|\mathcal{V}_A|} \sum_{v \in \mathcal{V}_A} \left(s_v^{R_i}\right) \end{aligned} \tag{4.7}$$

where $\mathbf{W}_A \in \mathbb{R}^{q \times dK}$ is a learnable parameter matrix, $\mathbf{b}_A \in \mathbb{R}^q$ is an additive bias, and $\mathbf{q}_A \in \mathbb{R}^q$ is the relation-level attention for nodes of type A. Note that the above parameters are shared for all nodes in the same type (e.g. all type A's nodes). After obtaining the importance of each relation, we compute the normalized attention weight β_r via softmax function:

$$\beta_{R_i} = \frac{\exp\left(w_{R_i}\right)}{\sum_{R_i \in \mathcal{R}_A} \exp\left(w_{R_i}\right)} \tag{4.8}$$

With the relative importance of each relation, we weighted sum relation-specific node embeddings $\{\mathbf{m}_v^{R_i}, \ \forall R_i \in \mathcal{R}_A\}$ to obtain the final neighborhood message for node v:

$$\mathbf{m}_v = \left(\sum_{R_i \in \mathcal{R}_A} \beta_{R_i} \cdot \mathbf{m}_v^{R_i} \right) \tag{4.9}$$

Neighbors' message $\mathbf{m}_v$ is then concatenated with the features $\mathbf{x}_v$ of target node v itself to update node embedding:

$$\mathbf{h}_v = \mathbf{m}_v || \mathbf{x}_v \tag{4.10}$$

For fake health news detection task, node embedding $\mathbf{h}_v$ for $v \in \mathcal{V}_{news}$ is passed through a two-layer perceptron followed by a softmax function to predict the fake/real class probabilities:

$$\begin{gathered} \mathbf{z}_v = \text{MLP}(\mathbf{h}_v) \quad , \forall v \in \mathcal{V}_{news} \\ [p_v^{fake}, \ p_v^{real}] = softmax(\mathbf{z}_v) \end{gathered} \tag{4.11}$$

where p_v^C is the probability that node v is classified to class $C \in \{fake, real\}$.

4.4 Two-stage Training

4.4.1 Motivation

In each step of message passing, a node aggregates information from its directly connected neighbors. To receive messages from other nodes at n hops away, a graph neural network needs to have a least n layers. Therefore, we can simply add more GNN layers. However, training a deep graph neural network may lead to vanishing gradient, overfitting, or GNN-specific problems [3]. SAFER authors [4] have pointed out this phenomenon and thus stacked two GNN layers for fake news detection. We also observed that a baseline graph neural network (RGCN) in our study drops 24% of performance when the third GNN layer is added.

Since news nodes will not receive information from users' social networks within two hops, SAFER removes tweet nodes and connects a news node with a user node if the user has shared the news in a tweet or retweet. In this setting, news can directly aggregate user information but the graph will lose tweet features as well as the edges of user retweeting and replying to a tweet. To send user network information to news nodes while maintaining the complete graph structure, we design a two-stage training for graph neural networks.

4.4.2 Process of Two-stage Training

The whole training process is divided into two steps: user representation learning and fake health news detection. We illustrate the overall process in Figure 4.3

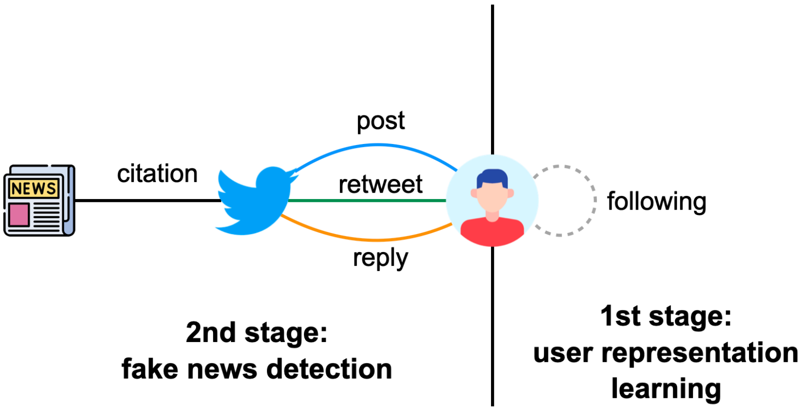

Figure 4.3: Two-stage training for graph neural networks.

At the first stage, we label social media users as fake/real news spreaders by their average ratings, and then construct a user subgraph consisting of user nodes and two interactions between them ($following$ and $followed\ by$). GNN takes the user subgraph as

doi:10.6342/NTU202200334

input and targets to classify social users. The best model, with the highest AUC score, is used to generate the contextual user representations. In the second stage, we replace user profiles with the pre-trained embedding as the initial user features. The two-layer GNN is input with the updated graph and trained to detect fake health news.

4.5 Loss Function

For the semi-supervised learning framework targeting to detect health misinformation, we calculate the loss from the 0.32% nodes (only news nodes are labeled) and optimize model parameters by minimizing the loss through gradient descent and backpropagation. Furthermore, since the dataset is imbalanced (28% fake news and 72% real news), we assign a larger penalty to the class with few samples, forcing models to pay attention to the minority class. The class weight w_c is calculated as:

$$w_C = \frac{n\_samples}{n\_classes \cdot n\_samples[C]} \quad , \forall C \in \{fake, real\} \tag{4.12}$$

where w_C is the weight of class C, $n\_classes$ is the number of unique target classes, $n\_samples$ is the total number of targets, and $n\_samples[C]$ is the number of targets in class C.

The cross entropy loss with class weight is then formulated as:

$$\mathcal{L}_{news} = -\sum_{v \in \mathcal{V}_{\text{news}}} \left(w_{fake} \cdot y_v \cdot \log\left(p_v^{fake}\right) + w_{real} \cdot (1 - y_v) \cdot \log\left(p_v^{real}\right) \right) \tag{4.13}$$

where $y_v \in \{0, 1\}$ is the label of node v.

doi:10.6342/NTU202200334

Chapter 5 Evaluation

5.1 Experimental Setup

5.1.1 Baselines

To demonstrate that the proposed method is superior in classifying health news, we compare HiGAT with current text-based and graph-based approaches. The NLP model is built by using the Transformers library [26], and all graph neural networks are implemented with Deep Graph Library (DGL) package [22]. The baseline models are listed as follows.

- **BigBird** is a Transformer-based language model. It is fine-tuned to detect fake news with news content only.
- **Support Vector Machine (SVM)** [5] is a machine learning classifier that makes predictions based on news features, such as news texts and the non-textual features of articles.
- **Graph Convolutional Network (GCN)** is a homogeneous GNN that aggregates nearby nodes' embeddings by a linear projection.
- **GraphSAGE** is also a homogeneous GNN that averages messages from the sam-

doi:10.6342/NTU202200334

pled neighborhood.

- **Heteogeneous Graph Attention Network (HAN)** is a heterogeneous GNN which learns metapath-specific node embeddings and combines them with different weights. HAN is fed with the graph structure, labeled node features, and user-defined metapaths.

- **Relational Graph Convolutional Network (RGCN)** is a heterogeneous GNN that aggregates the representations of neighbors in the same relation and learns a weight matrix for each relation.

- **Heterogeneous Graph Transformer (HGT)** designs the Transformer-like attention architecture to learn the importance of different meta relations in a heterogeneous graph. Note that relative temporal encoding and the sampling algorithm are not included in the DGL-version HGT.

We input graph structure and the selected node features to graph-based approaches. To handle features of different dimensions, we add node feature transformation to RGCN and HGT, and use zero padding for homogeneous GNNs as they view all nodes in the same type.

5.1.2 Dataset Splits and Evaluation Metrics

For the task of fake health news detection, we use news articles published before the year 2016 as the training set (69%), news in 2016 for validation (12%), and news between 2017 and 2018 as testing (19%). In each set, there are about 28% fake and 72% real news stories. Due to class imbalance, we use AUC (area under the ROC curve) as the primary

doi:10.6342/NTU202200334

metric to measure model performance. Nevertheless, we provide precision, recall, and F1 score for readers to compare classification algorithms from different perspectives. Each experiment is run with five random seeds and the average result is reported.

5.1.3 Implementation Details

For all graph-based methods, we keep two GNN layers and set the hidden dimension to 64 for fake news detection and 256 for user representation learning. For GNNs that apply node feature transformation (RGCN, HGT, HiGAT), the projected dimension is set to 512. For GNNs with the multi-head attention mechanism (HAN, HGT, HiGAT), we use 4 heads to stabilize the training. We set the learning rate to 0.001 and use Adam optimizer with weight decay of 10^{-4} for all graph neural networks. We train GNNs for 200 epochs and adopt early stopping with patience set to 20 epochs to avoid overfitting.

5.2 Results

5.2.1 Fake Health News Detection

The experimental results of all methods in end-to-end learning are summarized in Table 5.1. As shown in the table, the proposed HiGAT outperforms the existing text-based and graph-based algorithms on fake health news detection in both AUC and F1-score.

Support Vector Machine (SVM) is fed with news content only or all news features, including news texts, news media, news length, and the count of numeric values. The result shows that SVM with non-textual features ($\text{SVM}_{\text{with non}-\text{text}}$) gets better performance than SVM with only news texts. It is consistent with the analysis in the Section of Feature

doi:10.6342/NTU202200334

Engineering.

Homogeneous GNNs have worst prediction performance compared to the text-based approaches and heterogeneous GNNs. It suggests that aggregating neighborhood information without noticing the diverse relations in a graph will minimize the differences between two news classes, thus undermining models' abilities to detect misinformation.

We also observe that HAN predicts poorly among four heterogeneous GNNs in Table 5.1. The possible reason is that HAN requires meta-paths specified by human beings and humans may omit some meta-paths that machines may find useful during iterative training. Besides, though HAN utilizes graph structure to aggregate news nodes in different semantics, it does not consider tweet contents or user profiles. These two limitations cause the loss of important information for identifying fake news.

With taking features of all nodes as input, RGCN, HGT, and HiGAT reach higher scores than HAN. Interestingly, though hetero-GNN baselines can spot significantly more fake health news than BigBird, they do not defeat the performance of $\mathrm{SVM}_{\text{with non-text}}$. The result indicates that news text and other news features are effective for fake news detection. It also shows that RGCN and HGT may need extra graph information, such as user interaction on Twitter, to better classify health news stories. Therefore, we apply the proposed training technique to them as well as our model to see whether including user networks can improve fake news detection.

5.2.2 Two-stage Training

We report the results of two-stage training in Table 5.2. It is observed that HiGAT's performance on news classification is enhanced with two-stage training, thus it achieves

doi:10.6342/NTU202200334

Table 5.1: Results of training models in the end-to-end manner for fake health news detection. Numbers highlighted in **bold** denote the best score of that evaluation metric.

Category	Model	Precision	Recall	F1 Score	AUC
Text-based	BigBird	0.5209	0.6950	0.5891	0.7189
	SVM	0.4921	0.7750	0.6019	0.7322
	SVM$_{\text{with non-text}}$	**0.5463**	0.7375	0.6277	0.7498
Homo-GNN	GCN	0.3357	0.6800	0.4490	0.5788
	GraphSAGE	0.4624	0.6925	0.5544	0.6899
Hetero-GNN	HAN	0.4930	0.7875	0.6058	0.7360
	RGCN	0.5090	0.7825	0.6158	0.7442
	HGT	0.5008	0.8050	0.6174	0.7467
	HiGAT	0.5165	**0.8250**	**0.6347**	**0.7620**

the highest score among the two training techniques. HGT is also improved and now more powerful than SVM fed with all news features. The results of HiGAT and HGT indicate that attention mechanisms allow models to capture the important followers and friends from the complicated user social networks, thus learning meaningful user representations. In the second stage, news nodes can access the pre-trained user embeddings in two times of message passing.

On the other hand, though RGCN obtains higher precision with two-stage training, other evaluation scores decrease a little. This is because RGCN summarizes the information of all followers or all following people with equal importance, causing the trained user representations not that effective.

5.2.3 Ablation Study

Node-level attention and relation-level attention are the core components of HiGAT. To systematically analyze their effects, we conduct an ablation study by removing each

doi:10.6342/NTU202200334

Table 5.2: Comparison results between end-to-end training and two-stage training for GNNs in the task of fake health news detection. **Bold** numbers denote the best value in each evaluation metric. Symbol * indicates GNN has higher score in two-stage training than in end-to-end learning.

Category	**Model**	**Precision**	**Recall**	**F1 Score**	**AUC**
End to End	RGCN	0.5090	0.7825	0.6158	0.7442
	HGT	0.5008	0.8050	0.6174	0.7467
	HiGAT	0.5165	**0.8250**	0.6347	0.7620
Two Stages	RGCN	0.5136*	0.7675	0.6153	0.7425
	HGT	0.5183*	0.7925	0.6237*	0.7511*
	HiGAT	**0.5310***	0.8050	**0.6396***	**0.7642***

Table 5.3: Results for ablation study. All HiGAT variants are trained in the end-to-end manner.

Model	**Precision**	**Recall**	**F1 Score**	**AUC**
HiGAT$_{-\text{node}}$	0.5122	0.8125	0.6281	0.7558
HiGAT$_{-\text{relation}}$	0.4874	0.8475	0.6183	0.7500
HiGAT	**0.5165**	**0.8250**	**0.6347**	**0.7620**

of them. In detail, HiGAT$_{-\text{node}}$ is a model that aggregates neighboring nodes in the same relation with mean operation, and HiGAT$_{-\text{relation}}$ assigns same weights for all relations. Except for the differences mentioned above, all other settings are the same as HiGAT. The results are shown in Table 5.3. As can be seen, simply averaging neighborhood information in HiGAT$_{-\text{node}}$ causes the main evaluation scores to drop, which reveals the necessity of extracting messages from the important nodes. Besides, by removing relation-level attention, HiGAT$_{-\text{relation}}$ drops more obviously, suggesting that different relations in a heterogeneous graph do have unequal importance. The ablation study thus validates the effectiveness of two major components in the proposed model. Nonetheless, two HiGAT variants still outperform all the baselines on the AUC metric.

doi:10.6342/NTU202200334

Chapter 6 Conclusion and Future Work

6.1 Conclusion

In this thesis, we first refine the HealthStory dataset and perform feature analysis to screen the useful non-textual features for fake health news detection. We then propose a graph learning framework named Hierarchical Graph Attention Network (HiGAT). Our method is capable of capturing the important nodes and interactions in a heterogeneous graph by using node-level attention and relation-level attention in a hierarchical manner. Meanwhile, since GNN is often built with a limited number of layers, we further propose two-stage training to embed user network information into news articles.

We conduct experiments for the task of fake health news detection. The results show that our architecture is superior to all text-based and graph-based baselines, and two-stage training can improve the classification performance of attention-based GNNs. Last, we demonstrate the efficacy of the major components of HiGAT in an ablation study.

6.2 Future Work

Graph neural networks can be trained in both transductive and inductive settings. We perform transductive learning to identify misleading health news in this work, and we plan to generalize the predictions on completely unseen graphs in the future. To run experiments in an inductive setup, we will split the original graph into three disconnected subgraphs. GNNs are fed with the first two subgraphs for training and validation, and make predictions on the third subgraph for testing.

doi:10.6342/NTU202200334

References

[1] Hundreds dead because of covid-19 misinformation, 2020. URL https://www.bbc.com/news/world-53755067.

[2] Robocalls, rumors and emails: Last-minute election disinformation floods voters, 2020. URL https://n.pr/3fcMsyi.

[3] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In International Conference on Learning Representations, 2021.

[4] Shantanu Chandra, Pushkar Mishra, Helen Yannakoudakis, Madhav Nimishakavi, Marzieh Saeidi, and Ekaterina Shutova. Graph-based modeling of online communities for fake news detection. arXiv preprint arXiv:2008.06274, 2020.

[5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):1–27, 2011.

[6] Jian Cui, Kwanwoo Kim, Seung Ho Na, and Seungwon Shin. Hetero-scan: Towards social context aware fake news detection via heterogeneous graph neural network. arXiv preprint arXiv:2109.08022, 2021.

[7] Enyan Dai, Yiwei Sun, and Suhang Wang. Ginger cannot cure cancer: Battling fake health news with a comprehensive data repository. In Proceedings of the

doi:10.6342/NTU202200334

International AAAI Conference on Web and Social Media, volume 14, pages 853–862, 2020.

[8] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In Proceedings of The Web Conference 2020, pages 2331–2341, 2020.

[9] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In International Conference on Machine Learning, pages 1263–1272, 2017.

[10] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, pages 1025–1035, 2017.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In Proceedings of The Web Conference 2020, pages 2704–2710, 2020.

[12] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In 5th International Conference on Learning Representations, 2017.

[13] Valeria Mazzeo, Andrea Rapisarda, and Giovanni Giuffrida. Detection of fake news on covid-19 on web search engines. Frontiers in Physics, pages 14–14, 2021.

[14] Dat Quoc Nguyen, Thanh Vu, and Anh-Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 9–14, 2020.

[15] Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. Fang: Leveraging social context for fake news detection using graph representation. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pages 1165–1174, 2020.

[16] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. Automatic detection of fake news. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3391–3401, 2018.

[17] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. A stylometric inquiry into hyperpartisan and fake news. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 231–240, 2018.

[18] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, pages 671–688, 2019.

[19] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In European Semantic Web Conference, pages 593–607, 2018.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 5998–6008, 2017.

[21] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In International Conference on Learning Representations, 2018.

doi:10.6342/NTU202200334

[22] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315, 2019.

[23] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In The World Wide Web Conference, pages 2022–2032, 2019.

[24] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 849–857, 2018.

[25] Yuxiang Wang, Yongheng Zhang, Xuebo Li, and Xinyao Yu. Covid-19 fake news detection using bidirectional encoder representations from transformers based models. arXiv preprint arXiv:2109.14816, 2021.

[26] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, 2020.

[27] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. Advances in Neural Information Processing Systems, 33, 2020.

[28] Xinyi Zhou, Jindi Wu, and Reza Zafarani. Safe: Similarity-aware multi-modal fake news detection. In 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2020, pages 354–367, 2020.

doi:10.6342/NTU202200334