IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024SMP0004

Publicized:2024/08/21

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Embedding Learning with Relational Heterogeneous Information in Social Network Posts to Detect Malicious Behavior

Ryo YOSHIDA[†a)], *Nonmember*, **Soh YOSHIDA**[†b)], *Member*, *and* **Mitsuji MUNEYASU**[†c)], *Fellow*

SUMMARY Although social networking services (SNS) have enabled the free exchange of opinions and feelings, the posting of malicious content has increasingly become a problem. To solve this problem, malicious behavior detection methods based on posting behavior are being developed. Existing methods focus on semantic analysis of posts using natural language processing, and one existing approach uses graph neural networks to consider context from various elements, such as users, posts, hashtags, and entities. However, this approach does not adequately capture the complex patterns and interactions of SNS networks. In particular, it is insufficient to fully capture the complexity of heterogeneity between nodes and edges in an SNS network. In this paper, we propose a method for extended heterogeneous graph construction and an architecture for heterogeneous graph embedding learning. The proposed method focuses on and exploits the diverse heterogeneity of social networks, optimally integrates heterogeneous information from SNS posts, and analyzes the relationships in the data to improve the performance of malicious behavior detection. The effectiveness of the proposed method is demonstrated by evaluation on a newly collected large dataset.

key words: *malicious behavior detection, heterogeneous information*

1. Introduction

With the development of social networking services (SNS), it has become easier for users to speak and discuss freely. However, the malicious posting of hate speech, offensive remarks, and misinformation have increasingly become a problem. For example, there have been criminal and civil proceedings arising from the use of SNS to defame and slander people in election campaigns and from false information about COVID-19 vaccines. These problems have forced social networking platforms to develop policies for removing malicious content.

Many researchers have conducted studies on malicious behavior detection (MBD)–in particular, to detect malicious posting behavior–from textual information using natural language processing (NLP) methods [1]. These methods extract textual features from posts using bidirectional encoder representations from transformers (BERT) [2] and other methods, and feed these features into a multi-layer perceptron (MLP) to detect malicious posts. However, this approach is not suitable for detecting all types of malicious posts. One problem is that it cannot determine the maliciousness of a post whose text is plausible and polite. Considering the context of the posts, which cannot be determined from the text alone, improves the accuracy of MBD.

GraphBERT [3] is an example of the above approach and is considered to represent the state of the art. A heterogeneous graph is constructed by taking the users, posts, hashtags, and entities contained in posts as nodes and the relationships between them as edges. In this heterogeneous graph, the initial features are constructed from the documents and words at each node using NLP, and the graph structure is learned using graph neural networks (GNNs). An architecture has been designed to generate embeddings that take into account both the semantic information of the posts and the contextual information.

Despite the progress made with heterogeneous graphs and GNNs, current models are still insufficient to comprehensively handle the heterogeneity inherent in graphs (*i.e.*, the presence of different types of nodes and edges, each with different properties and meanings). In particular, they do not sufficiently distinguish the unique relations of different types of nodes and edges. This oversight leads to underutilization of the wealth of relational data available in heterogeneous graphs.

This paper focuses on the use of heterogeneous graphs in MBD methods for SNS posts and proposes an approach that improves on existing methods. Its main novelty and contributions can be summarized by the following three points.

1. Design of an extended heterogeneous graph construction method (**Novelty 1**). We construct an extended heterogeneous graph to fully exploit the rich information contained in SNS posts for learning with GNNs. In contrast to existing methods that deal with specific related information, such as entities and hashtags, our method integrates and exploits all of this information in a comprehensive manner.
2. Design of an embedded learning architecture that accounts for heterogeneity (**Novelty 2**). Simply increasing the number of types of nodes in a heterogeneous graph is not sufficient to enable the understanding of the meaning of nodes and improve performance. Therefore, we developed an architecture to enable the understanding of, and exploit, the complex relationships between nodes of heterogeneous graphs. The architecture is constructed around a heterogeneous graph transformer (HGT) [4] that captures the properties of different types of nodes and edges and learns accurate contextualized embeddings based on their relationships. The pro-

[†]Graduate School of Science and Engineering, Kansai University Yamate-cho 3–3–35, Suita-shi, Osaka, 564-8680 Japan
a) E-mail: k924073@kansai-u.ac.jp
b) E-mail: sohy@kansai-u.ac.jp
c) E-mail: muneyasu@kansai-u.ac.jp

cess introduces algorithms and optimization techniques suited to heterogeneous graph structures and captures relationships that are often missed by traditional methods. The proposed method accurately captures heterogeneity by constructing an extended heterogeneous graph for SNS posts. The resulting embeddings can be connected to various deep learning architectures, and the proposed method is sufficiently simple that it can be easily extended to existing research and more advanced architectures in the future.

3. Validation on real-world data. The effectiveness of the proposed method has been validated using data posted on social networking sites in Japanese. Specifically, we selected topics related to COVID-19 on which real-world malicious posts have been made, and tested the ability of the method to detect malicious posts using a large novel dataset collected from X (formerly Twitter). In this experiment, we quantitatively evaluated the extent to which the proposed method improved accuracy and efficiency compared with previous benchmarks.

2. Related Work

Most research on MBD in SNS has focused on the detection of hate speech [5], [6]. Many existing methods [7]–[9] that have been shown to be effective use BERT, a language model based on bidirectional transformer encodings that learns bidirectional representations from a large corpus [10]. The accuracy of hate speech detection can be improved by replacing or integrating BERT embeddings with existing NLP-based modern features [7].

GraphBERT is a detection method that focuses on the comprehensive information available in SNS data. It considers the context of posts, using GNNs, in addition to the semantic information in the text. By extracting elements contained in posts and forming edges between them, a heterogeneous graph is constructed that expresses relational information. GraphBERT then applies a graph convolutional network (GCN) [11] to this heterogeneous graph to learn embeddings for detecting malicious posts. In addition, it uses a weight adaptation BERT module implemented between the transformer layers to use the relational information obtained by the GCN to improve the embedding of tweets. However, it does not take into account the heterogeneity of nodes and edges.

Finally, we consider GNNs, a deep learning framework for graph-structured data. The GNN aggregates the features of neighboring nodes and generates a new feature vector for each node by considering the relationships between nodes. GraphSAGE [12] aggregates the features of neighboring nodes by random sampling to enable scalable learning for large graphs. The graph attention network (GAT) [13] uses attention to focus on important nodes and selectively aggregate the features of neighboring nodes. Extensions of homogeneous graphs to heterogeneous graphs are popular for complementing complex patterns in real-world data. The heterogeneous attention network (HAN) [14] uses attention to capture relationships between different types of nodes in heterogeneous graphs. The HAN highlights important relationships between nodes along a particular metapath and represents the complex structure of heterogeneous graphs. Conversely, the HGT is based on Transformer [10] and takes into account the diversity of types of nodes and edges in heterogeneous graphs. The HGT is based on the attention mechanism, which is aggregated and combined from the relationships and structure between heterogeneous nodes in heterogeneous graphs, to mutually enhance heterogeneous elements and extract more detailed information from them.

3. Preliminaries

3.1 Problem Definition

An overview of the proposed heterogeneous graph is shown in Fig. 1. The heterogeneous graph is defined as $G = (V, E, X, \tau, \phi)$, where V is the set of nodes, E is the set of edges, and $X \in \mathbb{R}^{|V| \times d_x}$ is the feature matrix. $x_v \in \mathbb{R}^{d_x}$ denotes the feature vector of node $v \in V$. Each node $v \in V$ has type $\tau(v)$ and each edge $e \in E$ has type $\phi(e)$. Here, τ is a mapping function for node types and ϕ is a mapping function for edge types. The set of node types and the set of edge types are defined as $\mathcal{T}_V = \{\tau(v) : \forall v \in V\}$ and $\mathcal{T}_E = \{\phi(e) : \forall e \in E\}$, respectively. If $|\mathcal{T}_V| = |\mathcal{T}_E| = 1$, the graph is homogeneous. Specifically, the set of node types $\mathcal{T}_V$ includes V_u (users), V_p (posts), and V_o (elements of posts), with V_o being the interaction-based V_{me} (mentions) and V_{ht} (hashtags) and the attribute-based V_{em} (emoji), V_{tp} (topic), and V_{et} (entity). These are collectively referred to as the elements of posts. The set of edge types $\mathcal{T}_E$ includes verb form types such as E_p (posts), E^o_{in} (includes element o), and E_{sim} (is similar to). That is, τ and ϕ map nodes to these node types and edges to these edge types, respectively. To the best of our knowledge, no existing method uses all elements derived from SNS posts in such a manner.

Suppose that $T = \{t_0, t_1, \ldots, t_N\}$ is the set of posts from the SNS, where N is the total number of posts. Each post t_i is assigned a label $y_i \in \{0, 1\}$, where 1 and 0 indicate malicious and non-malicious posts, respectively. The set of labels is $Y_T = \{y_{t_0}, y_{t_1}, \ldots, y_{t_N}\}$. Our task is to construct a machine learning model $f_\theta : T \rightarrow Y_T$ that classifies the posts into two classes. On a heterogeneous graph, the set T represents the posts categorized under node type V_p, and we obtain the results of this task as a classification problem for nodes $v \in V_p$ using a heterogeneous graph G as input.

3.2 Data Collection and Data Annotation

In this study, we collected SNS data by web crawling, using the official API in accordance with the privacy policy defined by X Corp. Existing public datasets, such as hate speech18 [15] and MMHS (multimodal hate speech) 150K [6], are not suitable for contextual analysis (the focus of this paper) because they contain only the text of posts after the

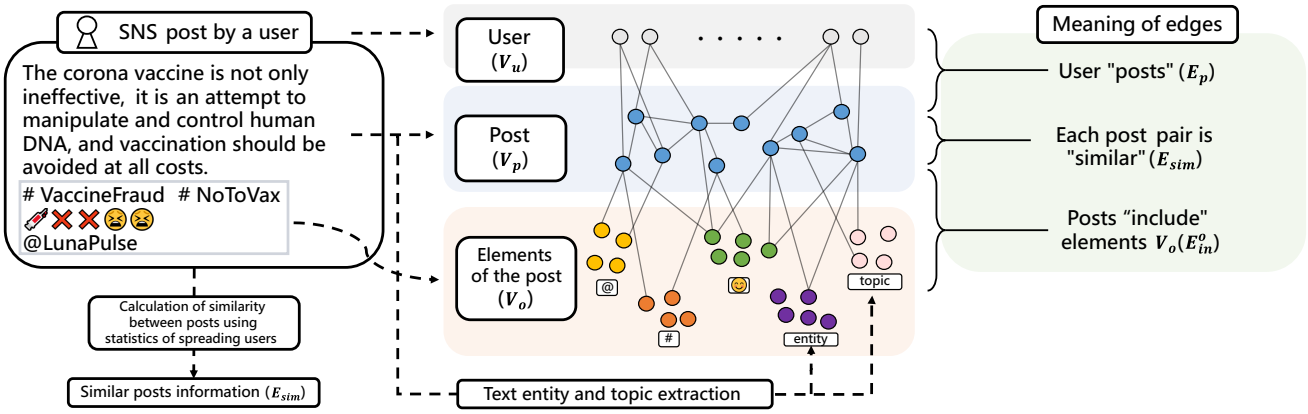


Fig. 1: Overview of the proposed extended heterogeneous graph.

user information has been removed. The subject of the study was posts about COVID-19 vaccination in Japan, which were particularly numerous during the vaccination launch period and the emergency declaration period, and some of which contained skeptical content about vaccination and offensive and malicious content directed at vaccinators and medical institutions. We used these posts to construct a dataset suitable for the task of identifying malicious posts. Specifically, posts containing the query "corona vaccine vaccination (コロナ ワクチン接種)" or "COVID vaccine vaccination (COVID ワクチン接種)" were collected for the period from January 2020 to April 2023, yielding approximately 3.4 million posts. To limit the collection of posts to those with high impact, we restricted the collection to the 134,950 posts that had at least 10 reposts. This amount of data satisfies the requirements of this study because it is larger than the 10,568 posts studied in [3] and the 91,500 posts studied in [15], and is comparable to the 149,823 posts studied in [6].

For annotation, we used the Perspective API [16] developed by Google Jigsaw, which evaluates the aggressiveness of a text as a score. The API is based on a Transformer-based model that was trained on various sources, including comments in online forms, such as Wikipedia and The New York Times. The API rates the offensiveness of a sentence by a probability score ranging from 0 to 1. Score attributes include TOXICITY, INSULT, PROFANITY, THREAT, and INFLAMMATORY; these allow users to select the score that best suits their purpose. In a preliminary experiment, we analyzed several maliciousness scores returned by the API and used them as annotation criteria. We visually inspected the scores of several posts that were randomly selected from the collected posts and obtained their scores. Posts that were judged to be malicious tended to have higher TOXICITY and INSULT scores. Thresholds of 0.35 for TOXICITY and 0.25 for INSULT were set, and posts whose scores exceeded either of these thresholds were flagged as malicious. As a result, of the 134,950 posts in the dataset, 125,438 posts were classified as normal and 9,512 as malicious.

The Perspective API has the limitation that it cannot fully capture contextual information because it only analyzes text. Therefore, using this labeled dataset as the ground truth in this study may compromise the accuracy of the evaluation. However, by setting low TOXICITY and INSULT thresholds, the proportion of posts that are considered hard samples, which are difficult to classify as malicious or not, is high. This intentional adjustment introduces a more challenging classification task, thereby providing a stringent test to assess the effectiveness of the proposed method through numerical comparisons with previous studies. Regarding the effect of mislabeling on classification losses such as cross-entropy, it has been reported that memorization [17] reduces performance in proportion to the noise rate [18], but this is not expected to affect relative numerical comparisons.

4. Proposed Method

In the proposed method, an extended heterogeneous graph G is first constructed from a set of SNS posts T. Second, using G as input, an embedding H is learned using an HGT-based architecture that takes into account the heterogeneity of the graph. The proposed architecture is designed to deal with imbalances in the dataset. Finally, the resulting embedding H is input to the MLP and MBD is performed according to both the semantic and relational information of the posts.

4.1 Heterogeneous Graph Construction

Our heterogeneous graphs allow us to go beyond mere textual analysis to gain a deeper understanding of user behavior and relationship patterns. For example, by analyzing trends in the use of specific hashtags and topics and interactions in a particular community through mentions, it is possible to identify accounts involved in malicious campaigns, slander, and spreading falsehoods and to reveal the network structure behind such campaigns. To achieve this, our **Novelty 1** is the construction of a comprehensive heterogeneous graph that makes maximum use of SNS posts. Specifically, the graph is constructed by the following steps.

Heterogeneous nodes. First, mentions, hashtags, and emojis are extracted from the posts. Entities are extracted using

the Python library SpaCy[†]. Here, four entity types were selected: place name, facility name, organization name, and name. These nodes (mentions, hashtags, emojis, and entities) are limited to those that appear three or more times in the collected posts, to ensure that only the more influential ones are extracted. Second, topic classification is performed and topics are assigned to posts. For topic classification, a cleaning process is performed by removing hashtags and emojis from each post and converting all numbers in the post to zeros. The resulting document set is then subjected to topic classification using latent Dirichlet allocation (LDA) [19]. LDA performs clustering of the latent semantic structure of all documents in the document set. The parameters of LDA are the number of topics to be generated and the number of words that make up a topic; the number of topics is set to 30 and the number of words in a topic to 10.

From each node v, a 768-dimensional (d_x=768) feature vector x_v is obtained using pretrained BERT [20]. For posts, hashtags, entities, and topics, BERT is applied directly to each document or word. For emoji, we use the Python library emoji[††] to convert emoji to words, and then apply BERT. For each user node, the average BERT feature vector of the last three posts of the user is used as the user feature.

Heterogeneous edges. Edges are defined between users and posts, between posts and elements, and between posts and their similar posts. The similarity between posts is defined according to statistics related to the number of spreading users of the posts in question. For our target SNS platform, spreading refers to reposting (formerly retweeting). Specifically, similarity edges measure the number of spreading users common to both posts in the pair using the Simpson coefficient, which is a method for computing the similarity between two sets. By using as the denominator the number of users that spread the post with the smaller number of spreading users, the similarity can be calculated in a manner that suppresses the effect of any difference in the number of spreading users. Let $\mathbf{r}_i$ be the set of users that spread t_i in the post set T. The similarity between two posts $\mathbf{r}_i, \mathbf{r}_j$ is then defined as follows.

$$Sim(\mathbf{r}_i, \mathbf{r}_j) = \frac{|\mathbf{r}_i \cdot \mathbf{r}_j|}{min(|\mathbf{r}_i|, |\mathbf{r}_j|)}, \tag{1}$$

where $|\mathbf{r}_i \cdot \mathbf{r}_j|$ is the number of common members of the two sets $\mathbf{r}_i$ and $\mathbf{r}_j$. For every pair of posts, their similarity $Sim(\mathbf{r}_i, \mathbf{r}_j)$ is calculated and they are connected with an edge if their similarity exceeds the threshold value of 0.8. Such an edge represents a relationship between two posts that are particularly similar to each other with respect to the users that spread them.

The graph constructed by the above procedure contains 172,110 nodes and 1,061,409 edges. The statistics of the dataset are shown in Tables 1 and 2.

[†]https://spacy.io/

[††]https://github.com/carpedm20/emoji

Table 1: Node statistics.

node_type		counts		
user		23,308		172,110
post		134,950		
other	hashtag	2,262	13,852	
	emoji	753		
	mention	750		
	topic	30		
	entity	10,057		

Table 2: Edge statistics.

edge_type		counts		
user-post		134,950	460,473	1,061,409
post-post		465,986		
other	post-hashtag	41,642		
	post-emoji	49,987		
	post-mention	10,173		
	post-topic	134,950		
	post-entity	223,721		

4.2 Overvierw

The HGT follows the basic principles of the GNN: the graph is considered to be a computational graph and the initial embedding assigned to a node is updated using the Aggregate function to generate an embedding that takes the graph structure into account. In heterogeneous graphs, there is a notion of a "metapath" at each target node. The metapath of our extended heterogeneous graph is defined as follows.

Definition of metapath. Metapaths are defined conceptually: $\langle V_u, E_p, V_p \rangle$ represents a user making a post, $\langle V_p, E^o_{in}, V_o \rangle$ represents a post containing an element, and $\langle V_p, E_{sim}, V_p \rangle$ represents similarity between posts. Further detailed analysis of metapaths in the graph reveals seven relationships between nodes connected by edges, which can be categorized into the following seven types of paths.

1. Between a user and a post: $\langle V_u, E_p, V_p \rangle$
2. Between a post and its mention: $\langle V_p, E^{me}_{in}, V_{me} \rangle$
3. Between a post and its hashtag: $\langle V_p, E^{ht}_{in}, V_{ht} \rangle$
4. Between a post and its emoji: $\langle V_p, E^{em}_{in}, V_{em} \rangle$
5. Between a post and its topic: $\langle V_p, E^{tp}_{in}, V_{tp} \rangle$
6. Between a post and its entity: $\langle V_p, E^{et}_{in}, V_{en} \rangle$
7. Between a post and its similar post: $\langle V_p, E_{sim}, V_p \rangle$

Applying the HGT to the graph allows for the preservation of unique representational spaces for each metapath. This means that it is possible to compute appropriate attention weights for each metapath, thereby obtaining embeddings derived from a representational space that reflects heterogeneity. This facilitates the discovery of hidden structures and patterns in the graph, enables precise analysis of malicious contexts, and improves detection accuracy. Therefore, our **Novelty 2** is an architecture that uses the HGT to employ metapath-dependent parameters characterizing heterogeneous attention on each type of edge in the extended heterogeneous graph. This maintains unique representations for different types of nodes and edges while performing classification.

4.3 Architecture Details

In an extended heterogeneous graph G, the meta-relationship of edge $e = (s, t)$ from source node $s \in V$ to target node $t \in V$ is represented as $\langle \tau(s), \phi(e), \tau(t) \rangle$. If t is the target node and s_1, s_2 are the source nodes obtained from its neighbors $N(t)$, the HGT is represented as edges $e_1 = (s_1, t), e_2 = (s_2, t)$ and the corresponding metapaths $\langle \tau(s_1), \phi(e_1), \tau(t) \rangle, \langle \tau(s_2), \phi(e_2), \tau(t) \rangle$ of G. The three operations $\mathbf{Attention}_{HGT}(\cdot)$, $\mathbf{Message}_{HGT}(\cdot)$, and $\mathbf{Aggregate}(\cdot)$ are then executed in order. The output of the l-th HGT layer is $H^{(l)}$, which is also the input of the $(l+1)$-th layer. Stacking L layers results in a contextualized representation $H^{(L)}$ that takes into account the relationships between nodes. This representation is generated by assigning elementary information from neighboring nodes to understand the meaning of the connections between nodes.

The GAT uses a single weight matrix to aggregate the information of neighboring nodes when applying the attention mechanism. This approach assumes that the source and target nodes have the same feature distribution, which is not guaranteed to be true for heterogeneous graphs. Mapping the target node t to the Query vector and the source node s to the Key vector, and using the product of these vectors, the mutual attention of the triplet $\langle \tau(s), \phi(e), \tau(t) \rangle$ is calculated. Specifically, the mutual attention for each edge $e = (s, t)$ is defined as follows.

$$\mathbf{Attention}_{HGT}(s, e, t) = \underset{\forall s \in N(t)}{\mathrm{Softmax}}\Big(\big\|_{i \in [1,h]} ATT\text{-}head^i(s, e, t) \Big), \tag{2}$$

$$ATT\text{-}head^i(s, e, t) = (K^i(s) W^{ATT}_{\phi(e)} Q^i(t)^T) \cdot \frac{\mu_{\langle \tau(s), \phi(e), \tau(t) \rangle}}{\sqrt{d}}, \tag{3}$$

$$K^i(s) = \text{K-Linear}^i_{\tau(s)}(H^{(l-1)}[s]), \tag{4}$$

$$Q^i(t) = \text{Q-Linear}^i_{\tau(t)}(H^{(l-1)}[t]). \tag{5}$$

Here, for the i-th attention head $ATT\text{-}head^i(s, e, t)$, the source node s of type $\tau(s)$ is linearly projected. $\text{K-Linear}^i_{\tau(s)} : \mathbb{R}^d \to \mathbb{R}^{d/h}$ is projected onto the i-th Key vector $K^i(s)$ using d. d is the hidden dimension of the HGT layer, h is the number of attention heads, and d/h is the vector dimension per head. The $\text{K-Linear}^i_{\tau(s)}$ is defined to hold a representation specific to the type $\tau(s)$ of the source node s. Similarly, the target node t is projected onto the i-th Query vector $Q^i(t)$ using the linear projection $\text{Q-Linear}^i_{\tau(t)}$. Furthermore, an edge-based matrix $W^{ATT}_{\phi(e)} \in \mathbb{R}^{d/h \times d/h}$ is prepared that is unique for each edge type $\phi(e)$. This allows for the computation of attention to be based on the unique representation distribution for each metapath. Furthermore, because all relations may not contribute equally to the target node, the tensor $\mu \in \mathbb{R}^{|\mathcal{T}_V| \times |\mathcal{T}_E| \times \mathcal{T}_V|}$ is added and adaptive scaling is performed for the mutual attention. Finally, h attention heads are concatenated to obtain $\mathbf{Attention}_{HGT}(s, e, t)$ for each pair of nodes.

In parallel with the computation of mutual attention, a message operation is performed to transfer information from the source node to the target node. This process also takes into account the meta-relationships and mitigates the differences in distribution between different types of nodes and edges, as in the case of attention computation. The message for the source node s of interest is defined as follows.

$$\mathbf{Message}_{HGT}(s, e, t) = \big\|_{i \in [1,h]} MSG\text{-}head^i(s, e, t), \tag{6}$$

$$MSG\text{-}head^i(s, e, t) = \text{M-Linear}^i_{\tau(s)}(H^{(l-1)}[s]) W^{MSG}_{\phi(e)}. \tag{7}$$

To obtain the i-th message head $MSG\text{-}head^i(s, e, t)$, the source node s of type $\tau(s)$ is projected onto a message vector using a linear projection $\text{M-Linear}^i_{\tau(s)} : \mathbb{R}^d \to \mathbb{R}^{d/h}$. Subsequently, a dedicated matrix for each edge type $W^{MSG}_{\phi(e)} \in \mathbb{R}^{d/h \times d/h}$ is employed to reflect the unique representational distribution for each metapath, concatenating h message heads to derive $\mathbf{Message}_{HGT}(s, e, t)$ for each pair of nodes.

Having obtained the message, $\mathbf{Attention}_{HGT}(s, e, t)$ and $\mathbf{Message}_{HGT}(s, e, t)$ are aggregated from the source nodes to the target node. In this process, the corresponding messages from the source nodes are averaged to yield the updated vector $\tilde{H}^{(l)}[t]$, as follows.

$$\tilde{H}^{(l)}[t] \leftarrow \underset{\forall s \in N(t)}{\mathbf{Aggregate}}(\mathbf{Attention}_{HGT}(s, e, t) \cdot \mathbf{Message}_{HGT}(s, e, t)), \tag{8}$$

$$\mathbf{Aggregate}(\cdot) = \sigma\left(\mathrm{Mean}(\cdot)\right), \tag{9}$$

where $\sigma(\cdot)$ denotes the activation function. Finally, to map the vector of the target node t to a distribution unique to its node type $\tau(t)$, the linear projection $\text{A-Linear}_{\tau(t)}$ is applied to the updated vector $\tilde{H}^{(l)}[t]$ and the following residual connection is used.

$$H^{(l)}[t] = \text{A-Linear}_{\tau(t)}(\sigma(\tilde{H}^{(l)}[t])) + H^{(l-1)}[t]. \tag{10}$$

This results in the output $H^{(l)}[t]$ of the l-th HGT layer for the target node t. The initial value $H^{(0)}[t]$ is the feature vector x_t of the target node t transformed using a linear transformation matrix with learnable parameters: $\mathbb{R}^{d_x} \to \mathbb{R}^d$. In addition, because of the small-world property of real-world graphs, by stacking fractional layers (L-layers, where L is a small value), each node can reach the majority of nodes that have different types and relationships to itself. The result is the output $H^{(L)}[t]$. The above allows us to design individual mutual attention weights that depend on the metapath $\langle \tau(s), \phi(e), \tau(t) \rangle$ and to learn in a manner that takes account of heterogeneity.

Loss function. The output obtained by Eq. (10) is input to the MLP, where it predicts whether a post t_i (corresponding to node $v \in V_p$ in G) is malicious. To maintain the textual information in the body of the post, the initial features x_{t_i} and $H^{(L)}[t_i]$ are skip connected. We refer to this method

as **Proposed method+**. In general, the number of malicious posts in the dataset is smaller than the number of normal posts (*i.e.*, data imbalance is likely). For this reason, the proposed method trains the L-layer HGT together with the MLP to minimize the focal loss (FL) [21], as follows.

$$\mathcal{L}_{\text{GNN-FL}} = -\sum_{v \in V_p} y_i (1 - p_v)^{\gamma} \log(p_v) + (1 - y_i) {p_v}^{\gamma} \log(1 - p_v), \quad (11)$$

$$p_v = \text{Softmax}\left(\sigma(H^{(L)}[v] \cdot W_1) \cdot W_2\right), \quad (12)$$

where γ represents the hyperparameters, W_1 and W_2 are the learnable parameter matrices of the MLP, and σ is the activation function. FL solves the problem of data imbalance by increasing the weights of misclassified training data. Predictions are made using p_v.

5. Experimental Results

The experiment was designed to answer the following research questions (RQs) and validate the proposed method.

- **RQ1**: How does the understanding of heterogeneous graphs enhance the performance of GNN-based MBD models?
- **RQ2**: To what extent do multiple node and edge types contribute to performance enhancement?
- **RQ3**: Can the proposed method achieve highly accurate contextual embeddings?

The objective of these RQs is to ascertain the benefit of analyzing the complex structure inherent in heterogeneous graphs. To answer RQ1, the proposed method was evaluated by comparing it with traditional GNN models. For RQ2, the contribution of individual elements of heterogeneous graphs to overall performance was investigated. For RQ3, the focus was on the ability of the proposed method to generate more effective embeddings by integrating heterogeneous information, in comparison with traditional GNNs such as the GCN. **Novelty 1** is evaluated by answering RQ2, and **Novelty 2** isevaluated by answering RQ1 and RQ3.

5.1 Experimental Setup

The proposed method was optimized using the adaptive moment estimation (Adam) optimizer with a learning rate of 5×10^{-4} and weight decay of 10^{-4}. The HGTs were configured with two layers ($= L$) and tested under two conditions with hidden dimensions of 64 and 128 ($= d$). The number of attention heads was set to 8 ($= h$), and the hyperparameter γ of FL was set to 1. Both the HGT and MLP used the rectified linear unit (ReLU) as the activation function.

For the comparison with existing methods, we selected both text-based baseline method using BERT and GNN-based embedding comparison methods (CMs), as described in [3]. These are listed below:

i). **BERT+MLP**: This method predicts the class of a post by training an MLP on the 768-dimensional BERT feature vectors extracted from the collection of SNS posts.
ii). **CM1 w/ GCN**: This method is a simplified version of GraphBERT, which combines a two-layer GCN with an MLP, using only graph representation learning.
iii). **CM2 w/ GAT**: This method combines a two-layer GAT with an MLP to assess the effect of introducing heterogeneous mutual attention.
iv). **CM3 w/ GraphSAGE**: Combining a two-layer GraphSAGE with an MLP, this method is another widely used GNN and is similar to a GCN.
v). **CM4 w/ HAN**: This method combines a two-layer HAN with an MLP to validate the effectiveness of considering heterogeneity using an HGT.

These methods comprise both approaches that analyze only text and those that analyze graph structures with GNNs. In particular, the purpose of the experiments was to ascertain how effectively the information in heterogeneous graphs could be used and to assess the impact of introducing the proposed HGT-based method on detection performance. CM1 to CM4, in common with the proposed method, take a graph as input. In addition, classification loss for both the baseline and comparison methods uses FL, adhering to the experimental conditions of the proposed method. The number of MLP parameters for all methods is also the same.

The F1 macro score was adopted as the evaluation metric in the experiments. This score quantifies the average accuracy of identifying both benign and malicious posts, to enable reliable evaluation on imbalanced datasets. The experiments divided the data into training data and test data in four ratios–20%, 40%, 50%, and 60%–to comprehensively examine the model's performance under varying degrees of data availability and a score was calculated for each. The percentage refers to the proportion of post nodes included in the training data, not the proportion of all nodes in the heterogeneous graph. The graph was split according to the specified proportion of post nodes. For example, for 60% training data, 60% of the total 134,950 posts were used as training post nodes, and all connected user nodes and post element nodes were included in the training graph. A similar process was applied to the test data. To ensure the reliability of the experimental results, five trials were conducted under the same conditions, and the average values of their F1 scores were reported.

5.2 Comparison Results for RQ1

Table 3 presents a numerical comparison of the proposed method with other methods. The highest F1 macro score is highlighted in bold and the second-highest F1 macro score is underlined. From the comparisons in Table 3, it is evident that the proposed method outperformed methods CM1 to CM4 under all conditions because of its understanding of heterogeneous graphs. Specifically, under Condition 1 (with a hidden layer dimension of 64), the proposed method achieved a 3% to 15% improvement in F1 macro score com-

Table 3: Comparison of proposed methods with other methods.

Methods	F1 macro score							
	Condition 1 with a hidden layer dimension of 64				Condition 2 with a hidden layer dimension of 128			
	20% Train	40% Train	50% Train	60% Train	20% Train	40% Train	50% Train	60% Train
BERT+MLP	68.37	68.30	70.61	69.74	**71.30**	**71.81**	71.41	71.44
CM1 w/ GCN	59.15	62.93	56.28	63.00	67.80	63.35	67.51	61.60
CN2 w/ GAT	63.70	64.17	63.32	64.24	63.84	64.69	64.93	64.47
CM3 w/ GraphSAGE	62.28	63.11	62.32	59.41	62.14	64.26	59.97	64.29
CM4 w/ HAN	65.76	65.87	65.34	65.45	67.66	67.16	66.82	67.30
Proposed method	68.91	70.12	70.95	70.58	68.44	69.86	70.56	70.63
Proposed method+	**70.51**	**71.37**	**72.03**	**72.17**	70.08	71.47	**71.74**	**71.98**

pared with CM1. Under Condition 2 (with a hidden layer dimension of 128), an improvement of 1% to 11% was observed. Moreover, the Proposed Method+ outperformed all other methods, including BERT+MLP, except at the lower training ratios (20% and 40%) under Condition 2. This indicates the superiority of the proposed method as an MBD approach. Of the GNN-based comparison methods, CM4, which is specialized for heterogeneous graphs, achieved the second-highest score, demonstrating the advantage of analyzing heterogeneous graphs. These results provide a positive answer to RQ1.

Although the Proposed Method+ fell short of the simpler BERT+MLP at lower training ratios (20% and 40%) under Condition 2, as a consequence of insufficient training, it demonstrated high accuracy (F1 macro score) compared with other GNN-based methods for imbalanced datasets. Specifically, it consistently outperformed other methods across various training data ratios, showing its ability to learn effectively from limited amounts of data and its resilience against overfitting with large datasets. The hidden layer dimension is a parameter that determines the complexity and number of features that a model can learn. The results suggest that a hidden layer dimension of 64 offers the optimal balance between performance and the increased computational cost and risk of overfitting associated with higher dimensionality.

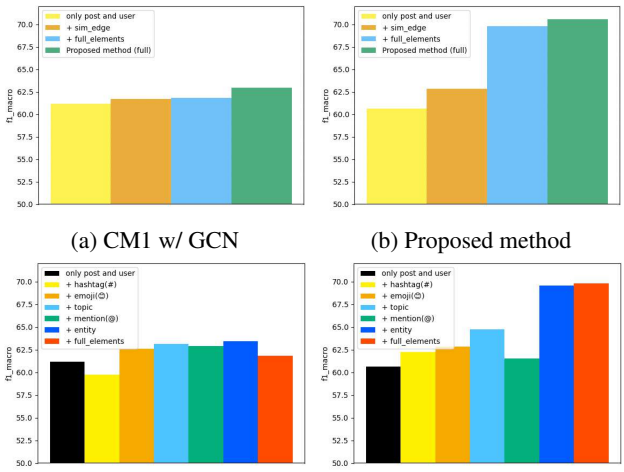
5.3 Ablation Studies for RQ2 and RQ3

We conducted ablation studies to understand the impact of integrating specific elements into heterogeneous graphs on learning outcomes, with the results presented in Table 4. This approach elucidates the contribution of node and edge types to performance. Table 4 shows the F1 macro scores for different graph configurations. "Only user and post" serves as the baseline, containing only these node types. "+ [element_name]" represents the baseline with a specific element added. "+ full_elements" includes all heterogeneous nodes from hashtag to entity. "Only user and post + sim_edge" represents the baseline with the addition of edges connecting similar posts. "Proposed method (full)" represents our full method with all heterogeneous nodes and edges. As indicated in Table 4, integrating each element into the basic heterogeneous graph structure, which comprises only user and post nodes, improved the F1 macro score. This finding suggests that the introduction of heterogeneous nodes and edges contributes to performance enhancement, providing a positive answer to RQ2. Entity and topic nodes significantly improved performance, suggesting that these attribute-based elements aid contextual understanding. Emoji nodes also improved performance, but to a lesser extent. In contrast, the performance gains from adding mentions and hashtags are limited. They are noisy on the SNS, suggesting the need for cleansing. The sim_edge improved accuracy by 2.16 points compared with the baseline, demonstrating the importance of post relationships. However, despite the large number of sim_edges compared with the number of other edge types (as shown in Table 2), their performance impact is relatively modest. This may be due to the current sim_edge formation method, which creates edges according to a threshold of user spread similarity, without considering the degree of similarity beyond this threshold. Despite this limitation, the current sim_edge formation method achieves an accuracy improvement of more than 2 points, indicating that edges representing post relationships are valuable in our method. Refining the sim_edge formation approach to capture detailed relationships between posts precisely could potentially lead to further performance gains.

Table 4: Contribution of each element to ablation studies.

Types	F1_macro_score
only user and post	60.64
+ hashtag	62.22
+ emoji	62.85
+ topic	64.73
+ mention	61.50
+ entity	69.57
+ full_elements	69.79
only user and post + sim_edge	62.80
Proposed method (full)	**70.58**

Tables 1 and 2 reveal quantitative differences between the components of heterogeneous graphs. In particular, the number of entity nodes is significantly higher than that of other element nodes; this difference may contribute to the substantial improvement in the F1 macro score. Conversely, although fewer in number, topic nodes improve performance significantly, reflecting their importance for highly specific data such as the targeted COVID-19 dataset. These observations suggest that the proposed method, which employs a Transformer-based HGT, effectively captures the complex structures and patterns of heterogeneous graphs, leading to high-precision contextualized embeddings. This provides



(a) CM1 w/ GCN (b) Proposed method

(c) CM1 w/ GCN (d) Proposed method

Fig. 2: Comparison of the performance of CM1 with that of the proposed method as heterogeneous nodes and edges are introduced individually.

a positive answer to RQ3, demonstrating the effectiveness of the proposed approach in exploiting the information in heterogeneous graphs.

Figure 2 compares the results of training CM1, using graphs to which multiple node and edge types were added individually as inputs, to those obtained from training the proposed method using a similar procedure. As shown in Fig. 2, CM1 achieved limited performance improvement with the addition of node and edge types. This outcome reinforces a positive answer to RQ3, demonstrating that the proposed method is well-suited to learning with extended heterogeneous graphs.

6. Conclusions

In this paper, we developed an extended heterogeneous graph construction method and a heterogeneous graph embedding learning architecture that enable the integration of heterogeneous elements and selection and mutual enhancement of heterogeneity. The effectiveness of these novelties in enhancing MBD was empirically validated through experiments conducted on a novel SNS dataset. These experiments demonstrated performance improvements in accurately identifying malicious content, underscoring the proposed method's capability in leveraging the complexity of heterogeneous information for more effective analysis.

Acknowledgment

This work was supported by JSPS (Japan Society for the Promotion of Science), KAKENHI Grant Number 22K18007, Japan, and the Kansai University Fund for Supporting Young Scholars, 2024. We thank Edanz (https://jp.edanz.com/ac) for editing a draft of this manuscript.

References

[1] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," ACM Computing Surveys, vol.51, pp.1–30, 2018.

[2] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.4171–4186, 2019.

[3] J. Wu, C. Zhang, Z. Liu, E. Zhang, S. Wilson, and C. Zhang, "Graph-bert: Bridging graph and text for malicious behavior detection on social media," Proceedings of the IEEE International Conference on Data Mining, pp.548–557, 2022.

[4] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," Proceedings of the International World Wide Web Conference, pp.2704–2710, 2020.

[5] S. MacAvaney, H.R. Yao, E. Yang, K. Russell, N. Goharian, and O. Frieder, "Hate speech detection: Challenges and solutions," PLOS ONE, vol.14, no.8, pp.1–16, 2019.

[6] R. Gomez, J. Gibert, L. Gómez, and D. Karatzas, "Exploring hate speech detection in multimodal publications," Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp.1459–1467, 2019.

[7] M. Mozafari, R. Farahbakhsh, and N. Crespi, "A bert-based transfer learning approach for hate speech detection in online social media," Proceedings of the International Conference on Complex Networks and Their Applications, pp.928–940, 2020.

[8] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Hate speech detection and racial bias mitigation in social media based on bert model," PLOS ONE, vol.15, no.8, pp.1–26, 2020.

[9] G. Rajput, N.S. Punn, S.K. Sonbhadra, and S. Agarwal, "Hate speech detection using static bert embeddings," Proceedings of the International Conference on Big Data Analytics, pp.67–77, 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L.u. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems, pp.6000–6010, 2017.

[11] T.N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," International Conference on Learning Representations, 2017.

[12] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," Advances in Neural Information Processing Systems, 2017.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," International Conference on Learning Representations, 2017.

[14] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P.S. Yu, "Heterogeneous graph attention network," Proceedings of the International World Wide Web Conference, pp.2022–2032, 2019.

[15] O. de Gibert, N. Perez, A. García-Pablos, and M. Cuadros, "Hate speech dataset from a white supremacy forum," Proceedings of the Workshop on Abusive Language Online, pp.11–20, 2018.

[16] A. Lees, V.Q. Tran, Y. Tay, J. Sorensen, J. Gupta, D. Metzler, and L. Vasserman, "A new generation of perspective api: Efficient multilingual character-level transformers," Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, p.3197–3207, 2022.

[17] D. Arpit, S.K. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M.S. Kanwal, T. Maharaj, A. Fischer, A.C. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," Int. Conf. Mach. Learn., pp.233–242, 2017.

[18] H. Song, M. Kim, D. Park, Y. Shin, and J.G. Lee, "Learning from noisy labels with deep neural networks: A survey," IEEE Transactions on Neural Networks and Learning Systems, vol.34, no.11, pp.8135–8153, 2023.

[19] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation,"

Journal of Machine Learning Research, vol.3, pp.993–1022, 2003.
[20] Tohoku NLP Group, "Bert japanese pretrained model." https://github.com/cl-tohoku/bert-japanese.
[21] T.Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.42, no.2, pp.318–327, 2020.

Ryo Yoshida received his B.S. degree in engineering science from Kansai University, Japan, in 2023. He is currently a master's degree student at Kansai University. His research interests are deep learning technologies and social media.

Soh Yoshida received his B.S., M.S., and Ph.D. degrees in engineering and information science from Hokkaido University, Japan, in 2012, 2014, and 2016, respectively. He joined the Faculty of Engineering, Kansai University, in 2016. He is currently an Associate Professor. His research interests are artificial intelligence technologies related to multimedia processing and information retrieval. He is a member of the ACM, IEEE, IEICE, and ITE.

Mitsuji Muneyasu received his B.S. and M.S. degrees in system engineering from Kobe University, Japan, in 1982 and 1984, respectively, and his Ph.D. degree in engineering from Hiroshima University, Japan, in 1993. He joined Oki Electric Industry Co., Ltd., Tokyo, Japan, in 1984. From 1990 to 1991, he was a Research Assistant at the Faculty of Engineering, Tottori University, Tottori, Japan. From 1991 to 2001, he was a Research Assistant and Associate Professor at the Faculty of Engineering, Hiroshima University, Higashi-Hiroshima, Japan. He joined the Faculty of Engineering, Kansai University, Osaka, Japan, in 2001. He is currently a Professor. His research interests are image processing theory and nonlinear digital signal processing. He is a fellow of the IEICE and a member of the IEEE and IPSJ.