Gated Multi-channel Network Embedding for Large-scale Mobile App Clustering

Yeo-Chan Yoon[1] and Soo Kyun Kim[2*]
[1] Department of Artificial Intelligence, Jeju National University
Jeju-si, Jeju-do, Republic of Korea
[e-mail: ycyoon@jejunu.ac.kr]
[2]Dept. of Computer Engineering, Jeju National University
Jeju-si, Jeju-do, Republic of Korea
[e-mail: kimsk@jejunu.ac.kr]
Corresponding author: Soo Kyun Kim

Abstract

This paper studies the task of embedding nodes with multiple graphs representing multiple information channels, which is useful in a large volume of network clustering tasks. By learning a node using multiple graphs, various characteristics of the node can be represented and embedded stably. Existing studies using multi-channel networks have been conducted by integrating heterogeneous graphs or limiting common nodes appearing in multiple graphs to have similar embeddings. Although these methods effectively represent nodes, it also has limitations by assuming that all networks provide the same amount of information. This paper proposes a method to overcome these limitations; The proposed method gives different weights according to the source graph when embedding nodes; the characteristics of the graph with more important information can be reflected more in the node. To this end, a novel method incorporating a multi-channel gate layer is proposed to weigh more important channels and ignore unnecessary data to embed a node with multiple graphs. Empirical experiments demonstrate the effectiveness of the proposed multi-channel-based embedding methods.

1. Introduction

Network embedding or graph embedding refers to the task of embedding a node in a network into a low-dimensional vector space. All nodes from a network such as a social network [1], mobile application (app) network [2, 3], or scientific paper network [4, 5] can be embedded through explicit or indirect connections. Embedding nodes enables similarities between nodes to be calculated and can be used for a variety of tasks such as node labeling, clustering, link prediction, visualization, and graph analysis [1]. Clustering is a representative task that can be used for the embedding task. Clustering groups similar nodes or documents and successfully extracting insights by classifying a large number of documents without an expensive prebuilt training set. Recently, clustering techniques have become frequently used for big data analysis.

Network-based clustering is one of the methods that groups nodes quickly and effectively by analyzing the importance of nodes and the strength of relationships between them based on graphs. Network-based clustering methods generally show higher performance because nodes are treated as connected objects rather than independently [3].

In recent years, research that applies the word2vec algorithm [6], which is widely used for word embedding, to network embedding has become mainstream. The word2vec-based method embeds a node by predicting its neighbor nodes; a randomized algorithm such as a random walk is used to sample neighbor nodes. DeepWalk [7] assumes that nodes are closer if they generate a higher sampling frequency during a random walk when analyzing the connection relationships in a graph. Node2vec [8] improves performance by considering both a breadth-first search and depth-first search strategy.
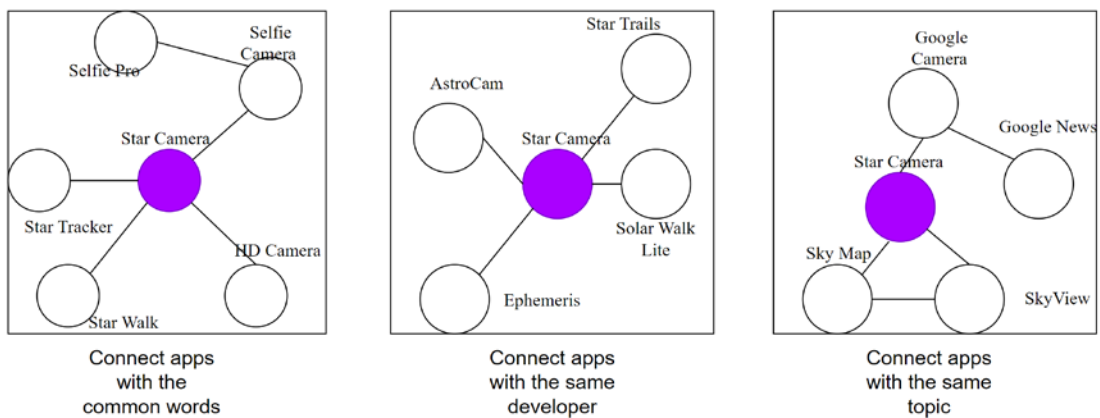

Fig. 1. Example of graphs built using different information.

Although the existing method builds the graph through explicit connections, the connection between nodes can be determined in various ways. **Fig. 1** shows graphs constructed differently according to various information about a mobile app. The graph on the left forms links between apps that share the same word in the title, and the middle graph connects applications created by the same developer. The third graph forms links between apps according to two topics: "con-stellation" and "camera," All three graphs were built with different views for one app, reflecting three different characteristics. In this paper, we propose a multi-channel embedding method that considers various types of graphs simultaneously. When embedding a node, the proposed method learns various latent characteristics of the nodes. When using multi-channel

information, the importance of each channel should be considered. It is intuitive to give more weight to channels with more important information and not to use channels that are of little help. To this end, we propose a method of embedding a node by adding a gate layer including a reset gate to the neural network.

The contributions of this paper are as follows. First, we propose a novel method that uses different types of interconnections to embed a certain node simultaneously using gated multi-channel embedding methods; This not only enables the method to learn several characteristics of the node but also enforces node embeddings to focus on more important information channels. it leads to better performance on the benchmark test set. Second, we propose an indirect interconnection method that can be applied to all domains. Using topic analysis results derived from the other method, we were able to successfully create a graph suitable for clustering tasks. Lastly, We prove that the proposed method is effective for big data analysis by applying it to a large data set.

2. Literature Review

2.1 Neural network-based embedding

The word2vec algorithm, which is based on a neural network, converts a high-dimensional integer vector represented by one-hot encoding into a low-dimensional real vector. CBOW and skip-gram are two methods that embed words with the word2vec algorithm [6]. CBOW learns to predict adjacent words from a series of input words, whereas skip-gram learns to predict a sequence of words from an input word. The learned projection layer is used to encode the one-hot encoded-word vector into a low-dimensional vector. As extensions of word-based representations, low-dimensional vector representations have been studied in terms of sentences, documents, or semantic units [6, 9, 10, 11, 12, 13, 14]. An embed-ding method called doc2vec, which is similar to word2vec, has been proposed to embed documents [9] and outperforms word2vec when comparing the similarity of documents [15]. Other studies have used knowledge bases to improve the performance of embeddings. Some studies have improved performance using the WordNet Knowledge Base [13, 16], whereas others have improved performance in a semi-supervised way [2]. In this paper, we improve the embedding performance using meta information and the topic analysis results of the latent Dirichlet allocation (LDA) model [17].

2.2 Graph Embedding

Some studies have investigated embedding nodes in graphs by extending text embedding methods [7, 8, 18]. Various types of graphs for social networks [1], scientific papers [4, 5], protein interactions [8], drug-target interactions [19] and mobile apps [2, 3] have been embedded to predict, cluster, or visualize nodes. Many previous approaches focus on the sampling strategy that is used to select target nodes for prediction. The DeepWalk [7] and node2vec [8] approaches used randomized sampling methods to select target nodes. The LINE [18] approach is an edge-sampling approach for embedding a large network, taking into account both first- and second-order proximities to avoid underfitting. In this paper, we simply use first-order proximity to reduce complexity. To avoid underfitting, we use multi-channel information instead. Text information is one of the primary sources for node representations. A graph can be built by interconnecting the shared words in the text of nodes [4]. PTE [20] is a semi-supervised embedding method that uses text networks instead of an unsupervised text embedding method such as skip grams. Yoon et al. [3] also used a text network to embed a

large number of mobile apps with a description set. In this study, we also use a text network as one of the basic channels for embedding nodes.
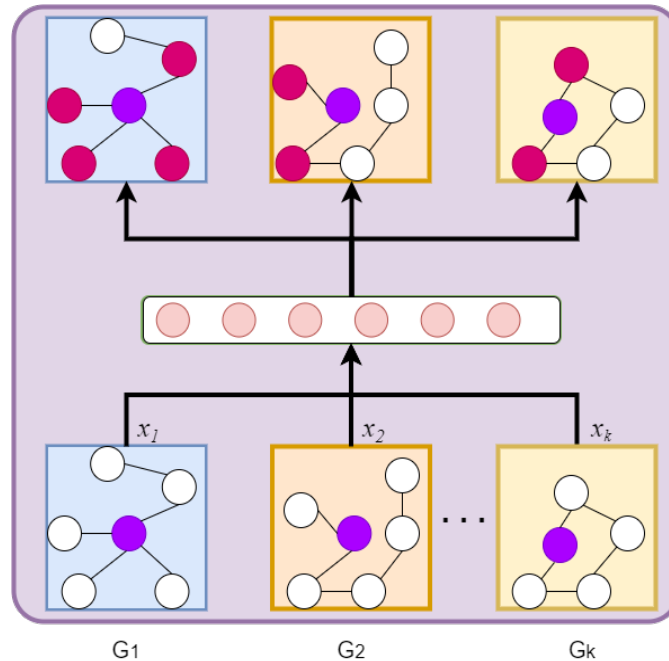

Fig. 2. DeepWalk with multi-channel information.

Recently, heterogeneous network embedding methods are being studied to utilize multiple information from multiple sources. NEDTP [19] builds a similar network of nodes based on 15 heterogeneous information networks to predict drug-target interactions. Duong et al. [21] introduce a heterogeneous network embedding method using message passing [22] for an information retrieval system. MixSp [23] studies multiple networks and enforces node embedding to be similar across multiple networks with cross-view co-regularization. However, in these methods, all networks are treated with equal weight and do not differentiate between sensitive and neglected information. In this paper, we propose a method of assigning high weight to important information and giving low weight to unnecessary information.

2.3 Network-based Clustering

Network clustering is an unsupervised approach to clustering network objects with affinity [4, 24, 25, 26, 27]; document networks or social networks are often clustered to analyze datasets without expending much human effort. Network-based clustering is different from content-based clustering in that it clusters documents based on connectivity. The network-based approach has performed better than content-based clustering in certain tasks [20, 3] because it uses human-labeled information such as citations and tagging information. Academic papers with citations [4, 5], web documents with hypertext [25, 28], social networks [29], and news [30] with text networks have been clustered using the network-based approach. In this paper, we used network-based clustering on a large volume of mobile apps. To this end, we use three channels: text links, developer links, and topic links.

3. Multichannel Graph Embedding

In this paper, multi-channel graph embedding means trans-forming $V \rightarrow R+$ when there are Ek $\in E$ and $Gk \in G$ for node-set V, edge set E, edge subset Ek, graph set G, and graph subset Gk. We propose a method for multi-channel embedding. The DeepWalk algorithm is a method of embedding a network based on negative sampling, and as shown in **Fig. 2**, it can learn to predict a neighboring node for every channel, so that it can be easily extended to multi-channel. However, when embedding a node, a difference in the amount of information included in each channel is not considered. The proposed method adds a gate layer to the negative sampling algorithm and considers the importance of each channel.

3.1 DeepWalk with Multiple Channels

The DeepWalk algorithm with multiple channels learns the node embedding function $V \rightarrow R+$ when graph $G_k = (V, E_k)$ and $G_k \in G$ are given for node V. Function f is shared across all graphs, as shown in **Fig. 2**. For a given graph $G_k = (V, E_k)$, $G_k \in G$, this algorithm minimizes the negative log-likelihood of the following summation and learns k embedding functions $f_k \in f$, $f : V \rightarrow R+$ as follows:

$$\underset{f}{\text{argmin}} \sum_{f_i \in f, N_i^S \in N^S} \sum_{v \in V} -logP\left(N_i^s(v)\middle|f_i(v)\right)$$

Here, $N^s(v)$ denotes the set of neighbor nodes of v and N_i^s denotes the set of neighbor nodes of v in the i-the graph. In this method, a neighbor node is sampled in a graph using a sequence-based method [7]. The neighbor node $N_s(v)$ is selected from a window ω in a sequence. Because v and $f(v)$ have a one-to-one mapping relationship, the probability of predicting the neighbor vector set $Ns(v)$ of embedding v can be defined as follows:

$$P\left(N^s(v)\middle|f(v)\right) = \sum_{n_i \in N^s(v)} logP(n_i|v)$$

In the embedding space, the symmetry between node v and neighbor nodes must be satisfied. This is because the similarity between nodes v_1 and v_2 should be constant. The softmax function satisfies this property.

$$logP(n|v) = \frac{ex\,p\left(f(n)f(v)\right)}{\sum_{u \in V} ex\,p\left(f(u),\ f(v)\right)}$$

Using the above formulas, the summation to be optimized can be calculated.

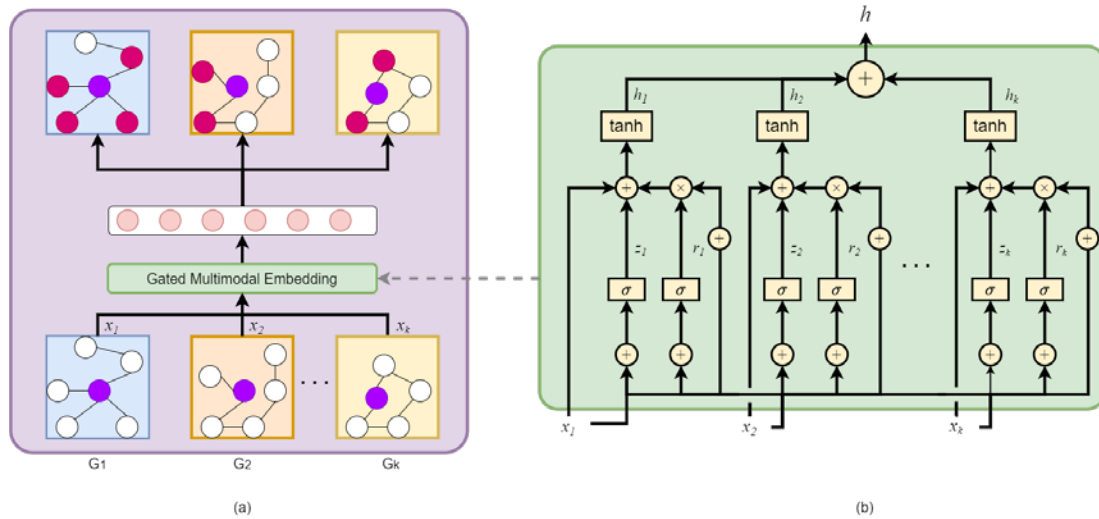## 3.2 Gated Multi-channel Network Embedding

Fig. 3. Network Embedding with a multichannel gate layer

John et al. [31] proposed a multimodal gate unit to classify movie genres using both image and text modalities. Inspired by this study, the proposed method learns by selecting channels when embedding nodes by adding a multi-channel gate layer to the negative sampling algorithm for network embedding. Learning from multiple sources is closely related to information fusion. Information fusion finds the optimal way to select the necessary information from various sources of information to predict the correct answer. The simplest and most effective method of information fusion is an ensemble method using a weighted sum or voting on the prediction results of a model trained using each source. However, to obtain the result, several models must be run simultaneously. Therefore, a method called late fusion, which feeds multi-source information into a single model and performs fusion at the final decision stage has been recently studied. Another method is to fuse information from each source at the feature level. It is a method called early fusion, where information selected from each source is projected onto a shared latent vector space and used for prediction. Fusion-based methods are commonly used for classification tasks. It uses classification-based neural networks such as CNNs and RNNs with an additional layer to handle information fusion. In this paper, an novel information fusion method for node embedding is introduced. It embeds network nodes by selecting important information from multiple sources by adding a gate layer to the negative sampling algorithm. That is, the gate layer used in GRU or LSTM is added to the multi-channel DeepWalk algorithm introduced in 3.1. **Fig. 3** a shows the negative sampling algorithm using the proposed gate layer. In the DeepWalk algorithm, a node with multiple graphs is simply projected onto a shared latent semantic space using a negative sampling algorithm. It assumes that all graphs have the same weights and does not consider node representations between graphs. To solve this problem, a gate layer as shown in **Fig. 3.b** is proposed. Each x_i represents a node x represented in graph i. The update gate adjusts how much x_i contributes to the output vector, and the reset gate determines how much information to ignore when x_i contains unnecessary information. The equations governing the multi-channel gate layer are as follows:

$$z_i = \sigma\left(\sum_{i=1}^{k} W_i^{(z)} \cdot x_i\right)$$

$$r_i = \sigma\left(\sum_{i=1}^{k} W_i^{(r)} \cdot x_i\right)$$

$$h_i = tanh\left(x_i + z_i + r_i \otimes \sum_{j \neq i}^{k} x_j\right)$$

$$h = \sum_{k}^{i} W_i^{(h)} \cdot h_i$$

$$\Theta = \{W_{i \in \{1,\dots,k\}}^{(z)}, W_{i \in \{1,\dots,k\}}^{(r)}, W_{i \in \{1,\dots,k\}}^{(h)}\}$$

z_i and r_i indicates results of the update gate and the reset gate. Θ indicates the parameters to be learned. With the gate layer, the probability of predicting the neighbor v can be defined as follow:

$$logP(n|v) = \frac{exp\left(f\big(MG(n)\big)f\big(MG(v)\big)\right)}{\sum_{u \in v} exp\left(f\big(MG(u)\big), f\big(MG(v)\big)\right)}$$

MG indicates the multi-channel gate layer.

4. Evaluations

Table 1. Test dataset. Five categories were selected in which various types of apps were registered.

		Life	Edu.	Travel	Tool	Ent.
Test dataset	# Apps in set 1	802	806	890	840	760
	# Apps in set 2	1000	1000	1000	1000	1000
	#Classes	67	35	19	52	30

4.1 Dataset and Experimental Setup

To evaluate the effectiveness of the proposed multi-channel embedding methods, we applied the methods to a fine-grained clustering task. For the graph set, we used a mobile app dataset collected by web crawling the Google Play Store.

For the test dataset, we used a fine-grained clustering dataset[3]. The Google Play Store has fewer than 60 default categories that developers can select when registering their app. The following categories were selected for the test dataset: life, education, travel, tools, and entertainment. The details of the datasets are given in **Table 1**. The test set was classified into two sets to evaluate the single-labeling and multi-labeling clustering performance of the multi-channel embedding model. For the first set, each app is assigned to a single cluster, and for the second set, each app can be assigned to multiple clusters. The mobile app datasets are formally defined as follows:

- MobileApp Graphs: $G_k \in G$ and $G_k = <V, E_k>$. Each G_k has its own rule for building edges E_k between nodes, which are as follows:

    - E_1: Connect apps sharing some title words;

    - E_2: Connect apps sharing a topic;

    - E_3: Connect apps built by the same developer.

For our experiments, we used a system equipped with an Intel i9 CPU clocked at 4.6 GHz and 128 GiB of main memory.

4.2 Clustering Performance

To evaluate the mobile app clustering performance of the multi-channel network-embedding algorithms GNME, we compared them with several existing embedding algorithms. We included not only neural-network-based embedding algorithms but also traditional frequency-based algorithms. All methods compared in the evaluation are listed below:

- TF: Nodes are embedded based on the term frequency in a description. The dimension of the embedding is decided by the total number of words N.

- LDA: To embed nodes as vectors, the topic-score vector is calculated from the description of the mobile app based on LDA [16]. The topic number was determined to be 124 using parameter tuning.

- Doc2vec: The doc2vec algorithm [13] is used for the node embedding. For this evaluation, we set the dimension size to 300, five negative examples were sampled, and the window was set to 8.

- DeepWalk: Network embedding with the word2vec algorithm. This algorithm selects neighbors with a random walk strategy. Nodes sharing title words are connected as neighbors.

- MDeepWalk: In **Fig. 2**, we applied the DeepWalk algorithm for multichannel embedding using the three channels introduced in Section 4.1. The algorithm predicts neighboring nodes from each of the three channels and reflects all their characteristics in the final node embeddings.

- GMNE: **Fig. 3** depicts the proposed multichannel embedding algorithm with a gate layer. This layer assigns higher weights to more important nodes and channels, addressing the issues of the MDeepWalk algorithm, which treats all nodes equally.

To evaluate the effectiveness of the embedding algorithms for graph clustering, each of the embedding algorithms was applied to the two following clustering algorithms:

-

Table 2. K-means clustering performance of each embedding method.

Measure	Embedding method	Life	Edu.	Travel	Tool	Ent.	Avg.
Purity	TF	0.39	0.39	0.7	0.28	0.32	0.42
	LDA	0.47	0.49	0.76	0.39	0.38	0.5
	Doc2vec	0.44	0.51	0.76	0.39	0.44	0.51
	DeepWalk	0.51	0.5	0.8	0.37	0.45	0.53
	MDeepWalk	0.53	0.55	0.82	0.45	0.42	0.55
	GMNE	**0.58**	**0.61**	**0.88**	**0.51**	**0.49**	**0.61**
Entropy	TF	3.25	2.67	1.26	3.85	3.12	2.83
	LDA	2.52	2.37	1.05	2.9	2.74	2.32
	Doc2vec	2.93	2.29	1.1	3.25	2.63	2.44
	DeepWalk	2.23	2.12	0.97	2.01	2.43	1.95
	MDeepWalk	2.31	2.09	1.04	1.99	2.31	1.95
	GMNE	**2.07**	**1.96**	**0.92**	**1.94**	**2.08**	**1.8**

- K-means clustering: This clustering minimizes the variance between the clusters. Each cluster has one centroid, and each object is assigned to the nearest center; objects assigned to the same center gather to form a cluster. To fairly compare algorithms, the gold-standard number of clusters k is given. For the development set, the lifestyle category was used and the hyperparameters were determined using the irace package [32].

- Keyword-based clustering: This clustering algorithm [3] clusters nodes by extracting representative words from a document. This algorithm calculates the probability that each app will be assigned to a specific cluster and assigns the app to the cluster when the probability exceeds a threshold. The advantage of this method is that it can be used for multiple labels.

We used purity and entropy as the evaluation measures to compare the clustering performances. Purity is an external evaluation index for measuring the quality of the clusters and is determined as follows:

$$Purity = \frac{1}{N} \sum_i^k max_j \left| c_i \cap t_j \right|$$

where N denotes the total number of documents, c_i denotes the i-th cluster predicted by the system, and t_j denotes the cluster tag that appears the most often in the documents that belong to c_i. Purity has a value between 0 and 1, and a higher value indicates better performance. Entropy is an indicator that measures the degree of unexpectedness and is determined as follows:

Table 3. Keyword-based clustering performance of each embedding method

Measure	Embedding method	Life	Edu.	Travel	Tool	Ent.	Avg.
Purity	**TF**	0.47	0.52	0.85	0.51	0.55	0.58
	LDA	0.55	0.68	**0.**89	0.63	0.64	0.68
	Doc2vec	0.64	0.61	0.81	0.69	0.62	0.67
	DeepWalk	0.65	0.6	0.8	0.7	0.63	0.68
	MDeepWalk	0.72	0.63	0.84	0.71	0.65	0.71
	GMNE	**0.84**	**0.78**	**0.92**	**0.8**	**0.74**	**0.82**
Entropy	**TF**	1.97	1.91	0.47	2.2	**1.19**	1.55
	LDA	1.5	1.29	0.45	1.25	1.48	1.19
	Doc2vec	1.25	1.32	0.43	1.14	1.35	1.1
	DeepWalk	1.27	1.26	0.51	1.23	1.3	1.11
	MDeepWalk	1.07	1.13	0.47	1.12	1.28	1.01
	GMNE	**0.97**	**0.95**	**0.41**	**0.97**	**1.12**	**0.88**

$$Entropy = \sum_{i}^{k} \sum_{j} -\frac{N_i}{N} \times p(t_j) log p(t_j)$$

where N_i is the number of documents that belong to c_i, and $p(t_j)$ is the probability that a document that belongs to c_i is classified as t_j.

Table 2 shows the performances of the embedding methods. The simple document frequency-based embedding method obtains average purity and entropy values of 0.42 and 2.83, respectively, and yields the lowest values in all categories. The doc2vec and LDA methods have similar average values for purity; the LDA method performs better in the lifestyle category and the doc2vec method performs better in the entertainment category. The proposed GMNE methods perform better concerning both purity and entropy compared to the DeepWalk method. In particular, the GMNE method obtains results that are better than those of DeepWalk by 8% concerning purity and 0.15 concerning entropy, which indicates that a model considering various graphs with a gate layer is more effective than a model that considers only one graph. Furthermore, the proposed GMNE methods exhibit better performance in all categories, indicating that the consideration of multiple graphs and the use of the gate layer can improve the quality of the embeddings. Overall, our experimental results demonstrate that the proposed GMNE methods can effectively capture the multi-channel characteristics of social networks and outperform state-of-the-art methods in terms of both
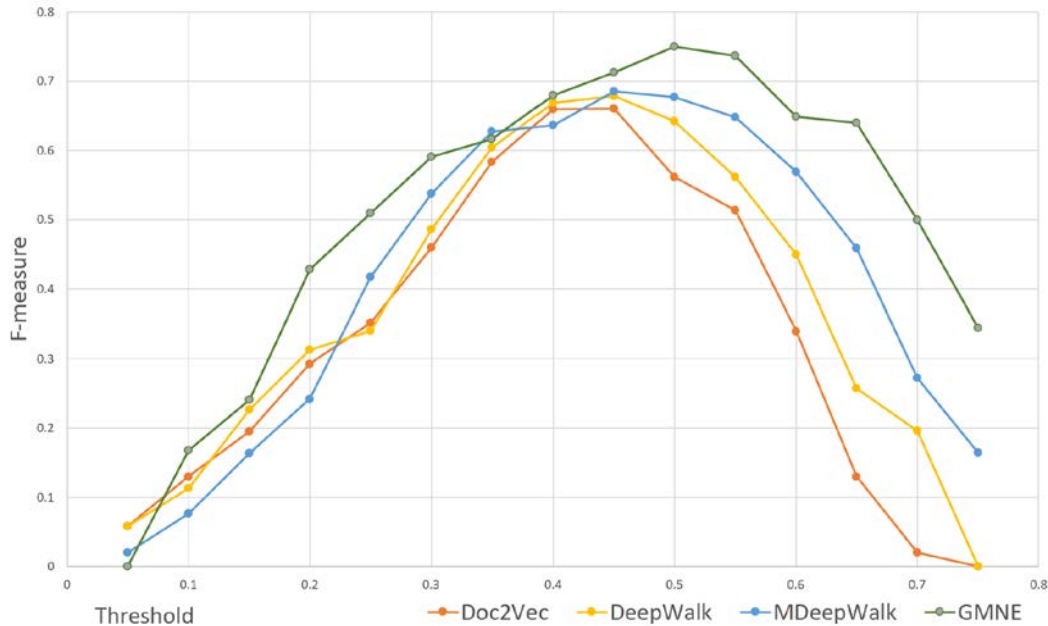
Fig. 4. Multi-labeling performance.

purity and entropy. **Table 3** shows the keyword-based clustering performance. The MDeepWalk method also performs better than the existing methods in the keyword-based clustering experiments. In MDeepWalk, the results of topic allocation obtained by LDA were used as the input of the topic channel. This indirectly has the same effect as an ensemble model and improves performance when compared with using the LDA or DeepWalk method alone. When using multiple channels, GMNE is more effective than MDeepWalk for both the k-means algorithm and the keyword-based algorithm. The keyword-based clustering performance evaluation in **Table 3** further supports the superiority of the GMNE methods over other embedding techniques. Specifically, the results show that the MDeepWalk method outperforms existing methods, including LDA and DeepWalk, in terms of purity and entropy metrics. This is attributed to the use of topic allocation results from LDA as an additional channel to complement the existing graph-based channel in MDeepWalk. However, when multiple channels are considered, the proposed GMNE methods are shown to be more effective than MDeepWalk in both the k-means algorithm and the keyword-based algorithm. This suggests that GMNE is better able to capture the diverse sources of information present in social networks and use them to generate high-quality embeddings for clustering tasks.

4.3 Multi-Labeling Performance

An app can be assigned to multiple clusters depending on its nature. For example, the "star camera" app can be assigned to a constellation topic cluster or a camera topic cluster. Set 2 is a dataset that includes apps with multiple labels. We hence used this dataset to evaluate the performance of the proposed method in the multi-labeling task. The F-measure is used as the metric for the evaluation. The F-measure of cluster C $F(C)$ is defined as follows:

$$F(C) = \sum_{x_j \in X} \frac{|x_j|}{|D|} \max_{c_i \in C}\{F\}, where\ F = \frac{2PR}{P+R}, P = \frac{|c_i \cap x_j|}{|c_i|}\ and\ R = \frac{|c_i \cap x_j|}{|x_j|}$$

In general, a high $F(C)$ value indicates that the cluster is of good quality. In this study, document x was allocated to every cluster C_i, where $p(C_i|x) > threshold$ using the proposed clustering method.

Fig. 4 exhibits the average F-measure scores of the embedding algorithms. The GMNE algorithm has an F-measure value of 0.75 for a threshold of 0.5, which is higher than those of doc2vec (0.66 at 0.45), DeepWalk (0.67 at 0.4), and MDeepWalk (0.68 at 0.45). The precision and recall for the highest F-measure value are respectively 0.77 and 0.73 for GMNE; 0.72 and 0.61 for doc2vec; 0.7 and 0.64 for DeepWalk; and 0.68 and 0.69 for MDeepWalk. Hence, both the precision and recall of the GMNE method are higher than those of the other methods. Overall, the results demonstrate that the proposed GMNE method is effective in multi-labeling tasks and outperforms the other embedding algorithms. This is attributed to the ability of GMNE to capture the multi-channel characteristics of networks, which enables it to assign apps to multiple clusters accurately. The findings suggest that the proposed method can be useful in various applications, such as personalized recommendation systems and content classification.

4.4 Ablation Test

We conducted ablation studies to evaluate the influence of each channel in our model. MDeepWalk and GMNE obtain the best performance when used with all channels, as expected. In k-means clustering, the performance drops most drastically without the text network, while in keyword-based clustering, the performance falls most significantly without the topic network. This reflects the nature of the keyword-based clustering algorithm, which uses textual information. This property can retain some of the properties of a text network even without an explicit text network.

These results demonstrate that all channels play a crucial role in achieving high performance in the proposed model. Overall, the ablation studies support the effectiveness of the proposed model in app clustering and suggest that all channels should be considered in app clustering tasks.

Table 4. Keyword-based clustering performance of each embedding method

Method	Channel	K-means		Keyword-based clustering	
		Purity	**Entropy**	**Purity**	**Entropy**
MDeepWalk	with all channels	0.554	1.948	0.71	1.014
	w/o topic network	0.523	1.957	0.681	1.121
	w/o text network	0.507	2.284	0.685	1.082
	w/o developer network	0.551	1.952	0.693	1.032
GMNE	with all channels	0.614	1.794	0.816	0.884
	w/o topic network	0.593	1.867	0.692	1.012
	w/o text network	0.542	1.989	0.759	0.911
	w/o developer network	0.597	1.852	0.791	0.909

5. Conclusion

In this study, we proposed two methods for embedding a certain node using multiple graphs: the GMNE with a multi-channel gate layer, which obtained better results than the base methods. The contributions of this study are as follows: First, unlike the previous approaches, we used different types of interconnections to embed a certain node using multichannel embedding methods; this enables the embedding vector to learn several characteristics about the node. Second, we proposed an indirect interconnection method that can be applied in all domains. By using topic analysis results derived from another method, we were able to successfully create a graph suitable for clustering tasks. Lastly, we proposed a method with a multi-channel gate layer. The gate layer addresses the mismatch from different graphs; it obtains better results than the methods that do not use the gate layer. We believe this study provides interesting directions for research in embedding networks. There remains considerable room for improving the proposed methods by discovering new types of graphs.

Acknowledgment

References

[1] Goyal, P., Ferrara, E, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, 151, 78–94, 2018. Article (CrossRef Link)

[2] Yoon, Y.C., Lee, J., Park, S.Y., Lee, C, "Fine-grained mobile application clustering model using retrofitted document embedding," *ETRI Journal*, 39, 443–454, 2017. Article (CrossRef Link)

[3] Yoon, Y.C., Gee, H.K., Lim, H, "Network-Based Document Clustering Using External Ranking Loss for Network Embedding," *IEEE Access*, 7, 155412–155423, 2019. Article (CrossRef Link)

[4] Sun, Y., Yu, Y., Han, J, "Ranking-Based Clustering of Heterogeneous Information Networks with Star Network Schema," in *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 797–806, 2019. Article (CrossRef Link)

[5] Zhang, S., Xu, Y., Zhang, W, "Clustering scientific document based on an extended citation model," *IEEE Access*, 7, 57037–57046, 2019. Article (CrossRef Link)

[6] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J, "Distributed representations of words and phrases and their compositionality," in *Proc. of the 26th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3111–3119, 2013. Article (CrossRef Link)

[7] Perozzi, B., Al-Rfou, R., Skiena, S, "Deepwalk: Online learning of social representations," in *Proc. of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014. Article (CrossRef Link)

[8] Grover, A., Leskovec, J, "node2vec: Scalable feature learning for networks," in *Proc. of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016. Article (CrossRef Link)

[9] Le, Q., Mikolov, T, "Distributed representations of sentences and documents," in *Proc. of International conference on machine learning*, pp. 1188–1196, 2014. Article (CrossRef Link)

[10] Dai, A.M., Olah, C., Le, Q.V, "Document embedding with paragraph vectors," *arXiv preprint arXiv:1507.07998*, 2015. Article (CrossRef Link)

[11] Kiros, R., Zhu, Y., Salakhutdinov, R.R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S, "Skip-thought vectors," in *Proc. of the 28th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3294–3302, 2015. Article (CrossRef Link)

[12] Wang, S., Tang, J., Aggarwal, C., Liu, H, "Linked document embedding for classification," in *Proc. of the 25th ACM international on conference on information and knowledge management*, pp. 115–124, 2016. Article (CrossRef Link)

[13] Johansson, R., Pina, L.N, "Embedding a semantic network in a word space," in *Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1428–1433, 2015. Article (CrossRef Link)

[14] Rothe, S., Schütze, H. Autoextend: "Extending word embeddings to embeddings for synsets and lexemes," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1793-1803, 2015. Article (CrossRef Link)

[15] Lau, J.H., Baldwin, T, "An empirical evaluation of doc2vec with practical insights into document embedding generation," *arXiv preprint arXiv:1607.05368*, 2016. Article (CrossRef Link)

[16] Dai, Y., Wang, S., Xiong, N.N., Guo, W, "A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks," *Electronics*, 9, 750, 2020. Article (CrossRef Link)

[17] Blei, D.M., Ng, A.Y., Jordan, M.I, "Latent dirichlet allocation," *Journal of machine Learning research*, 3, 993–1022, 2003. Article (CrossRef Link)

[18] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q. Line, "Large-scale information network embedding," in *Proc. of the 24th international conference on world wide web*, pp. 1067–1077, 2015. Article (CrossRef Link)

[19] Qi An and Liang Yu, "A heterogeneous network embedding frame-work for predicting similarity-based drug-target interactions," *Brief-ings in Bioinformatics*, 22.6, bbab275, 2021. Article (CrossRef Link)

[20] Qiu, J., Dong, Y., Ma, H., Li, J., Wang, K., Tang, J, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," in *Proc. of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 459–467, 2018. Article (CrossRef Link)

[21] Duong, Chi Thang, et al, "Efficient and effective multi-modal queries through heterogeneous network embedding," *IEEE Transactions on Knowledge and Data Engineering*, 34.11, 5307-5320, 2022. Article (CrossRef Link)

[22] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl, "Neural message passing for quantum chemistry," in *Proc. of International conference on machine learning, PMLR*, pp. 1263–1272, 2017. Article (CrossRef Link)

[23] Xu, L., Wang, J., He, L., Cao, J., Wei, X., Philip, S. Y., & Yamanishi, K., "MixSp: a framework for embedding heterogeneous information networks with arbitrary number of node and edge types," *IEEE Transactions on Knowledge and Data Engineering*, 33.6, 2627-2639, 2021. Article (CrossRef Link)

[24] Sutanto, T. E., & Nayak, R, "Ranking based clustering for social event detection," in *Proc. of Working Notes in Proc. of the MediaEval 2014 Workshop [CEUR Workshop Proceedings, Volume, 1263]*, Sun SITE Central Europe, pp. 1-2, 2014. Article (CrossRef Link)

[25] Jon M Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM (JACM)*, 46.5, 604–632, 1999. Article (CrossRef Link)

[26] Yang Liu and Songhua Xu, "A local context-aware lda model for topic modeling in a document network," *Journal of the Association for Information Science and Technology*, 68.6, 1429–1448, 2017. Article (CrossRef Link)

[27] Pei, Xiaobing, Chuanbo Chen, and Weihua Gong, "Concept factorization with adaptive neighbors for document clustering," *IEEE Transactions on Neural Networks and Learning Systems*, 29.2, 343-352, 2018. Article (CrossRef Link)

[28] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The pagerank citation ranking: Bringing order to the web," *Tech. report, Stanford InfoLab*, vol. 98, pp 161-172, 1999 Article (CrossRef Link)

[29] Rozemberczki, B., Davies, R., Sarkar, R., & Sutton, C., "Gemsec: Graph embedding with self clustering," in *Proc. of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*, pp. 65–72, 2019. Article (CrossRef Link)

[30] Xu, Z., Wei, X., Luo, X., Liu, Y., Mei, L., Hu, C., & Chen, L., "Knowle: a semantic link network based system for organizing large scale online news events," *Future Generation Computer Systems*, 43-44, 40-50, 2015. Article (CrossRef Link)

[31] Arevalo, J., Solorio, T., Montes-y-Gómez, M., & González, F. A, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017. Article (CrossRef Link)

[32] López-Ibáñez, Manuel, et al., "The irace package: Iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, 3, 43-58, 2016. Article (CrossRef Link)

Yeo-Chan Yoon received BS, MS and Ph.D degrees in computer science and engineering from Korea University, Seoul, Rep. of Korea, in 2004,2007 and 2020 respectively. Currently, he is an assistant professor in the Department of Artificial Intelligence at Jeju National University, Jeju-si, Rep. of Korea. His research interests include Deep Learning, Vision, Natural Language Processing, and machine learning

Soo Kyun Kim received Ph.D. in Computer Science & Engineering Department of Korea University, Seoul, Korea, in 2006. He joined Telecommunication R&D center at Samsung Electronics Co., Ltd., from 2006 and 2008. He worked a professor at Department of Game Engineering at Pai Chai University from 2008 to 2020. He is now a professor at Department of Computer Engineering at Jeju National University, Korea. Dr. Kim has published many research papers in international journals and conferences. Dr. Kim has been served as Chairs, program committee or organizing committee chair for many international conferences and workshops. His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics. He is a member of ACM, IEEE, IEEE CS, KACE, KMMS, KKITS and KIIT.