Biomedical Document Classification with Literature Graph Representations of Bibliographies and Entities

Ryuki Ida, Makoto Miwa, and **Yutaka Sasaki**
Computational Intelligence Laboratory
Toyota Technological Institute
2-12-1 Hisakata, Tempaku-ku, Nagoya, Aichi, 468-8511, Japan
{sd22401, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

Abstract

This paper proposes a new document classification method that incorporates the representations of a literature graph created from bibliographic and entity information. Recently, document classification performance has been significantly improved with large pre-trained language models; however, there still remain documents that are difficult to classify. External information, such as bibliographic information, citation links, descriptions of entities, and medical taxonomies, has been considered one of the keys to dealing with such documents in document classification. Although several document classification methods using external information have been proposed, they only consider limited relationships, e.g., word co-occurrence and citation relationships. However, there are multiple types of external information. To overcome the limitation of the conventional use of external information, we propose a document classification model that simultaneously considers bibliographic and entity information to deeply model the relationships among documents using the representations of the literature graph. The experimental results show that our proposed method outperforms existing methods on two document classification datasets in the biomedical domain with the help of the literature graph. Our source code is publicly available at https://github.com/tticoin/BDCL-LitGraph.

1 Introduction

Document classification has improved significantly with large language models, such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). However, these methods use only text information and ignore a lot of information behind the text, such as bibliographic information (e.g., authors and publishing journals), citation information, and information about entities that appear in the target text, which can be considered as one of the keys to better classification.

With the help of neural models, several document classification methods have been proposed that use both text information and external information. Yao et al. (2019) classified documents using representations of a text graph. In the text graph, nodes correspond to papers and words, and edges are connected between word nodes and their paper nodes and between word nodes with high co-occurrence frequency. The text graph allows paper nodes to be connected through the common word nodes and makes document classification take into account the relationships among papers. BertGCN (Lin et al., 2021) further adds text information to the node representations of the text graph using BERT. Yasunaga et al. (2022) proposed a pre-trained language model LinkBERT that considers the relationship between documents. The inputs of the model are texts from documents in a citation relationship in pre-training. LinkBERT achieved higher performance than existing BERT models. Although these studies indicate the effectiveness of external information other than text information, they only consider limited information, such as the co-occurrence of words and citation relationships, and do not simultaneously consider multiple types of external information.

This paper proposes a document classification model that incorporates multiple types of external information into the target text information. Specifically, we first build the literature graph using *bibliographic information*, including authors and publishing journals, and *entity information*, including descriptions of entities and their taxonomic information. Then, we create representation vectors of the nodes that consider various relationships among different types of nodes in the literature graph so that the vectors can contain multiple types of external information. Finally, we build a document classification model that receives both the representation vectors and the target text information as input.

The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, pages 385–395

The contributions of this paper are as follows:

- Using representation vectors from the literature graph containing bibliographic and entity information, a novel document classification model is proposed that incorporates the information into the target text information.
- The proposed model with literature graph representation performs better than existing models on two document classification datasets in the biomedical domain: Ohsumed (Joachims, 1998) and Hallmarks of Cancer (HoC) (Baker et al., 2015).

2 Related work

2.1 Document classification

There are two types of document classification methods: methods that use only the target text information and methods that consider external information in addition to the target text information.

Yao et al. (2019) proposed a document classification model TextGCN using a text graph in which papers and words are nodes and edges weighted by TF-IDF values were connected between each paper node and its word nodes, whose words appear in the paper. In this text graph, edges were also connected between word nodes when considered highly relevant according to the PMI values. A Graph Convolutional Network (GCN) (Kipf and Welling, 2017), which aggregates information from the surrounding nodes to a node through the edges connected to the node, was used to update the node representation of the text graph with taking into account the graph structure, and the representation was used for document classification. Furthermore, BertGCN (Lin et al., 2021) achieved higher performance than TextGCN by initializing the representation vectors of the paper nodes in the text graph with BERT to incorporate text information about the papers into the graph.

Yasunaga et al. (2022) proposed LinkBERT, a pre-trained model considering the relationship between documents by using two linked documents as input to BERT simultaneously. LinkBERT was pre-trained with two tasks: masked language modeling, which was proposed in the original BERT model, and document relation prediction, which took two documents as input and classified whether they were in a citation relationship or not, aiming at modeling the information on the dependencies between documents and the information across documents. As a result, LinkBERT outperformed existing methods on the GLUE (Wang et al., 2018) and BLURB (Gu et al., 2021) tasks, which are the benchmarks in the general and biomedical domains, and achieved higher performance than existing BERT models in document classification.

2.2 Graph representation Learning

Graph representation learning has been actively studied to obtain representation vectors of nodes and links from a graph by taking into account the graph structure (Hamilton, 2020). In graph representation learning, it is common to represent the graph structure as a set of triples (h, r, t) using the head h, the tail t and the relation r of a directed edge, and methods such as TransE (Bordes et al., 2013) and DistMult (Yang et al., 2015) have been proposed to learn to represent the nodes and links in these triples. For instance, TransE uses the distance between $h + r$ and t as the score function. TransE is a simple and effective method, but it has several problems. For example, TransE cannot represent the relations that have multiple tail nodes for a head node. There are several models such as TransH (Wang et al., 2014) and TransR (Lin et al., 2015) that address these limitations. These models project the node representations into the relation-specific space using projections for each relation.

RotatE (Sun et al., 2019) is capable of modeling various relation patterns, including symmetry/antisymmetry, inversion, and composition. In RotatE, each relation is defined as a rotation from the head node to the tail node in the complex space.

3 Proposed method

This section proposes a novel document classification model that incorporates a representation of a literature graph with bibliographic and entity information. Figure 1 shows an overview of the proposed method. We first explain the definition and representation learning of the literature graph in Sections 3.1 and 3.2, respectively. We then introduce a document classification model that uses both vector representations of the literature graph containing bibliographic and entity information, and the target text information is described in Section 3.3.

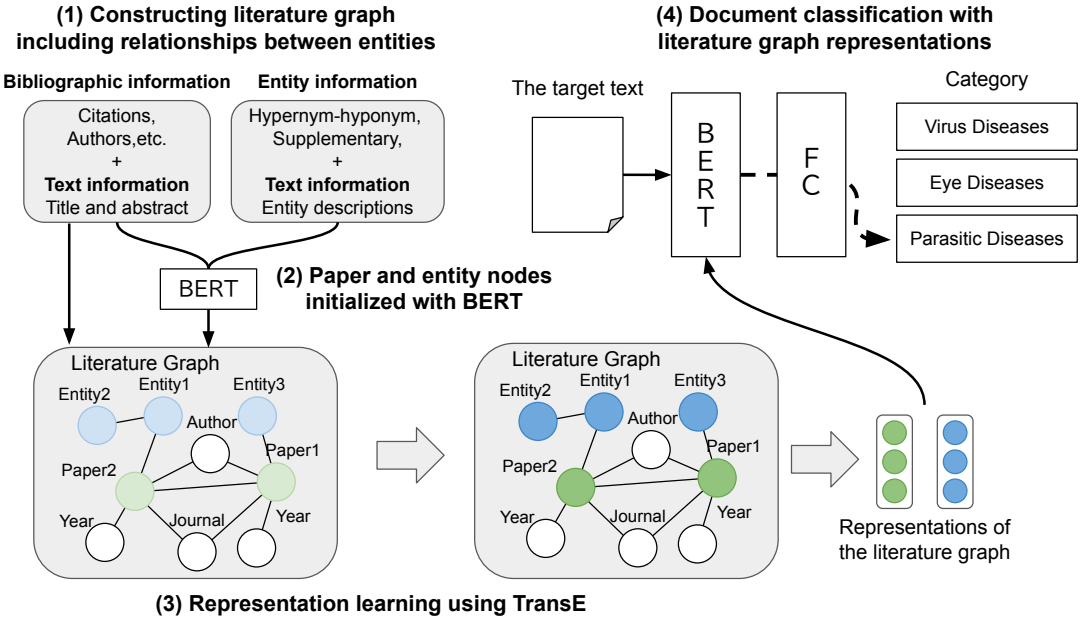

Figure 1: An overview of the proposed method. (1) The literature graph is constructed based on bibliographic and entity information. (2) The paper and entity nodes are initialized with BERT using their text information. (3) Representation learning is performed on the literature graph using TransE to obtain the vector representation of the literature graph. The node representations of the left graph are the initial representations. The node representations of the right graph are the learned representations. (4) The vector representations corresponding to the target paper in the literature graph are used as input of BERT. The representations corresponding to the target paper in the literature graph and the representations of the final layer of BERT that correspond to the paper are used for classification using a fully connected (FC) layer.

3.1 Definition of the literature graph

We define a literature graph based on bibliographic information and entity information. The literature graph has the papers, authors, publication years, publication venues, and entities as nodes. Each paper node is connected to other types of nodes based on bibliographic information about the paper. Each paper node is also connected to other paper nodes when they have a citation relationship. Furthermore, the entity nodes are connected if two entity nodes are in a hypernym-hyponym relationship or a supplementary concept relationship (Figure 1(1)).

The title and abstract are added to each paper node as text information, and the entity description providing the scope and content of the entity, which is given as "ScopeNote" in the entity database, is added to each entity node if it exists.

3.2 Representation learning on literature graph

First, the paper and entity nodes are initialized with BERT using their text information (Figure 1(2)). The representation of the [CLS] token in the BERT output is used as the initial representation of the corresponding node, since it is considered the representation of the whole sentence. If the text information is unavailable, the node is randomly initialized so that it follows a normal distribution of the mean and standard deviation of the node representations initialized by BERT. Other types of nodes are also randomly initialized.

Then, representation learning is performed on the literature graph using TransE (Bordes et al., 2013)[1] to obtain the vector representation of the literature graph (Figure 1(3)). The vector representation is expected to take into account various relationships between documents based on bibliographic and entity information, as well as the target text information.

3.3 Document classification with literature graph representation

To use the information from the literature graph in document classification, in addition to the target text information, the vector representations corre-

[1]TransE is employed because it is a simple model that deals with representations in the Euclidean space and the literature graph treated in this study is large and computationally expensive methods such as GCN (Kipf and Welling, 2017) cannot be easily applied. The application of other methods is left for future work.

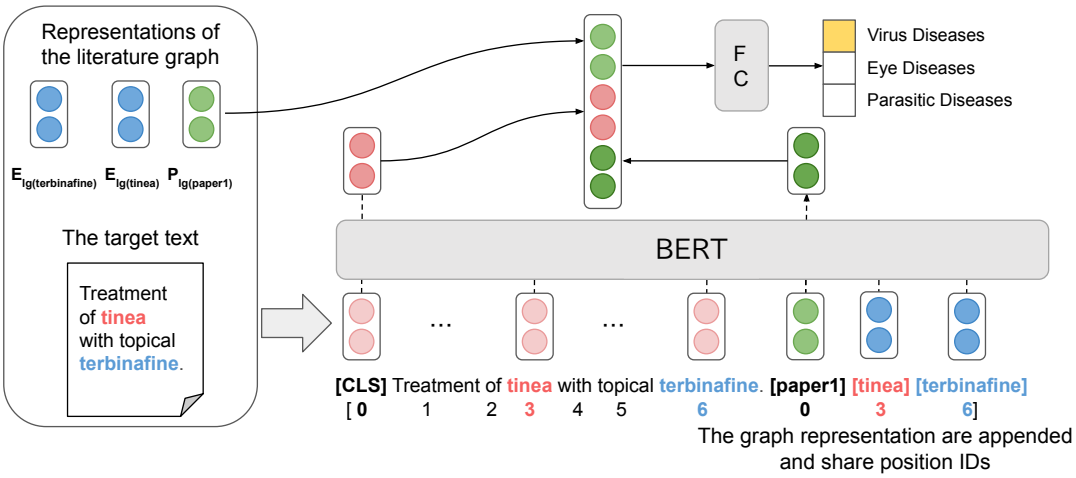

Figure 2: **Document classification model with literature graph representation.** This model receives the representations of the literature graph. When the target of the classification is paper1, $P_{lg(\text{paper1})}$ and $E_{lg(\text{entity name})}$ are the representation of the target paper and entities from the literature graph. Then, the representations of [CLS] and paper1 from the BERT final layer, as well as the representation from the literature graph $P_{lg(\text{paper1})}$ are concatenated and used for classification. Red circles are representations corresponding to words; green circles are representations corresponding to target paper; and blue circles are representations corresponding to target entities.

sponding to the target paper in the literature graph obtained in Section 3.2 are used (Figure 1(4)). For this purpose, the vector representations of the target paper node and entity nodes corresponding to the entities appearing in the paper denoted P_{lg} and E_{lg} are used.

The proposed model is shown in Figure 2. The entities in the target text are first extracted by case-insensitive string matching with the entities registered in the database. The paper node representation and the entity node representations, which correspond to the entities in the target text, are then appended to the target text information. The input of BERT when M entities are extracted from a paper i is as shown in the following Equation (1).

$$S = \{[\text{CLS}], w_1, \cdots, w_n, \\ [\text{P}_i], [\text{E}_1], \cdots, [\text{E}_M]\} \quad (1)$$

where w_i is a subword in the target text, P_i is the tokens representing the paper i's node of the literature graph, and E_i is a token representing the entity i's node of the literature graph. At this time, the tokens representing the nodes in the literature graph are mapped to the corresponding tokens in the text using the position IDs (Zhong and Chen, 2021); the same position ID is assigned to the [CLS] token and the token representing the paper node. Similarly, the same position ID is assigned to the first subword of the entity in the target text and the token representing the entity node. To allow appending as many paper and entity nodes as possible, the target text is truncated if it exceeds the BERT max length of 512 tokens. Also, if the beginning of an entity does not match the beginning of a subword, the representation of that entity is not used. The representations of [CLS] and w_i are assigned from the pre-trained BERT embedding table, the representations of P_i and E_i are assigned representation of the literature graph as follows:

$$\mathbf{W}^0 = \{\mathbf{w}_{[\text{CLS}]}, \mathbf{w}_{w_1}, \cdots, \mathbf{w}_{w_n}, \\ P_{lg(\text{P}_i)}, E_{lg(\text{E}_1)}, \cdots E_{lg(\text{E}_M)}\}, \quad (2)$$

where $\mathbf{w}_{[\text{CLS}]}$ and $\mathbf{w}_{w_i}$ are representations of [CLS] and w_i, respectively. $P_{lg(\text{P}_i)}$ and $E_{lg(\text{E}_i)}$ are representation of P_i and E_i of the literature graph. $\mathbf{W}^0$ is used as the input to BERT, and the representations of the final layer of BERT are obtained as in Equation 3.

$$\mathbf{W}^{l+1} = \text{Self-attention}^l(\mathbf{W}^l) \quad (3)$$

The representations of the final layer of BERT are represented as Equation 4. The representations among them corresponding to the [CLS] token and the paper node, as well as the paper representation of the literature graph, are concatenated to create $\mathbf{h}_{\text{paper}(i)}$ and used for classification using a fully connected (FC) layer. The output $\mathbf{z}_{\text{paper}(i)}$ from the FC layer is converted to probabilities by applying the softmax function for single-label data and the sigmoid function for multi-label data.

Relation Type	All	Train	Development	Test
Cites	246,136,539	241,213,809	2,461,365	2,461,365
Author	118,193,406	115,829,538	1,181,934	1,181,934
Year	33,405,863	32,737,747	334,058	334,058
Journal	33,405,863	32,737,747	334,058	334,058
MeSH	31,917,346	31,279,000	319,173	319,173
Hypernym	40,659	39,847	406	406
Supp	427,758	419,204	4,277	4,277
Total	463,527,434	454,256,892	4,635,271	4,635,271

Table 1: Triple statistics for the literature graph.

Node Type	
Paper	33,406,096
Author	4,932,150
Year	57
Journal	34,564
MeSH	348,081
Total	38,720,948

Table 2: Statistics on nodes in literature graphs.

$$\mathbf{W}^L = \{\mathbf{h}_{[\mathrm{CLS}]}, \mathbf{h}_{w_1}, \cdots, \mathbf{h}_{w_n}, P^L_{lg(\mathrm{P}_i)}, E^L_{lg(\mathrm{E}_1)}, \cdots, E^L_{lg(\mathrm{E}_M)}\} \tag{4}$$

$$\begin{aligned} \mathbf{h}_{\mathrm{paper}(i)} &= [\mathbf{h}_{[\mathrm{CLS}]}; P^L_{lg(\mathrm{P}_i)}; P_{lg(\mathrm{P}_i)}] \\ \mathbf{z}_{\mathrm{paper}(i)} &= \mathrm{FC}(\mathbf{h}_{\mathrm{paper}(i)}) \end{aligned} \tag{5}$$

4 Experimental Settings

The quality of the representation vectors obtained by representation learning of the literature graph is evaluated by link prediction, which predicts the target nodes that are related to the node. We also evaluated document classification using the representation of the literature graph to confirm the effectiveness of the literature graph in document classification. We used BioLinkBERT-base (Yasunaga et al., 2022) in both experiments.

4.1 Representation learning of the literature graph

The 2022 version of the medical literature database MEDLINE (National library of Medicine, 2020) was used to create the literature graph described in Section 3.1. The literature graph is huge because over 30 million articles are registered in MEDLINE, so author nodes that had few connections to other nodes with a degree of less than five were deleted. The 2021 version of MeSH (National Library of Medicine, 2020) was used as the entity database. Each paper was assigned with the MeSH as entities. We used only MeSH entities representing the most significant points labeled as "Major Topic" in MEDLINE for the MeSH relation in our literature graph. The literature graph has edges representing citation relations (Cites), between documents and their authors (Author), their years of publications (Year), their publication journals (Journal), and their entities (MeSH), hypernym-hyponym relations between entities (Hypernym), and supplementary concept relations between entities and supplementary MeSH entities (Supp). The statistics of nodes and edges of the literature graph are shown in Tables 1 and 2, respectively.

MAP@30 and Hit@N were employed as the evaluation metrics for link prediction on the literature graph. Data were split into a ratio of 98:1:1 for training, development, and test data sets, keeping the same ratio of relationship types. The development and test triples were chosen so that the nodes in the triples appear in the training data set. For the evaluation of link prediction, the target nodes included in the training data sets were removed from the prediction candidates. Only nodes with a target node type determined from the head and a relation type were used for prediction; for example, in predicting the Author relation from a paper node, the Author nodes were used for the prediction candidates. Section 4.3 shows the libraries and training setups used for the experiments.

For comparison, we build an entity graph that is a subgraph of the literature graph and has only entity nodes. The entity graph comprises hypernym-hyponym relationships and supplementary concept relationships among entities, so the entity node representations from the entity graph reflect the re-

Relation Type	MAP@30	Hit@1	Hit@3	Hit@10
Cites	0.0046	0.0005	0.0079	0.0397
Author	0.0283	0.0156	0.0335	0.1219
Year	0.3261	0.1789	0.3807	0.9712
Journal	0.1658	0.0973	0.1950	0.5237
MeSH	0.0870	0.0483	0.1017	0.3304
Hypernym	0.0851	0.0	0.1358	0.4444
Supp	0.0	0.0	0.0	0.0
macro average	0.0996	0.0487	0.1221	0.3473

Table 3: Results of link prediction on the literature graph

	Literature graph	Entity graph
Hypernym		
MAP@30	**0.0851**	0.0341
Hit@1	0.0	0.0
Hit@3	**0.1358**	0.0346
Hit@10	**0.4444**	0.3235
Supp		
MAP@30	0.0	**0.0428**
Hit@1	0.0	**0.0226**
Hit@3	0.0	**0.0525**
Hit@10	0.0	**0.1781**

Table 4: Comparison of link prediction on the literature graph and entity graph

lationships between entities. Entity graph representation is learned in the same setting as the literature graph, and the obtained entity node representations E_{eg} are used for document classification.

4.2 Document classification using the representation of the literature graph

For the evaluation, we used Ohsumed (Joachims, 1998) and Hallmarks of Cancer (HoC) (Baker et al., 2015). Ohsumed is a document classification dataset composed of abstracts in the biomedical domain, in which documents are assigned one or more of 23 different cardiovascular disease categories. Since Ohsumed is built using MeSH, the relationships between the papers and MeSH in Ohsumed were excluded from the literature graph, and this literature graph was used for both datasets. As in existing studies (Yao et al., 2019), documents with multiple categories were excluded. This resulted in 3,357 and 4,043 documents in the training and test data sets, respectively. The training data set was divided into 7:3 to create a development data set. Each document in the HoC was assigned multiple categories chosen from 10 different cancer features. HoC was split according to existing studies (Gu et al., 2021), and we used 1,295, 186, and 371 documents for training, development, and test data sets, respectively. Both datasets are suitable for evaluating our proposed method as they are assigned categories that are semantically close and difficult to classify with textual information. For both data, the entities in the papers were extracted by string matching with the entities in the database, resulting in 20.49 entities per paper for Ohsumed and 26.94 entities per paper for HoC on average.

In the evaluation, three models with different seeds of random numbers were trained for the Ohsumed, and five models for the HoC, and the average of their evaluation scores are reported as the final predicted result; accuracy was used for the evaluation of Ohsumed and the F1-measure for the evaluation of HoC. As a baseline model, we modified the document classification model in Section 3.3 to use only textual information as input. We compared the settings with or without paper and entity representations from the literature graph. We also compare the entity information from the literature and the entity graphs. When both entity information is used, each representation is added to the target text, and the same position ID is assigned to each representation and the first subword of the entity in the target text.

We chose the best setting for the final test on each corpus.

4.3 Experimental environments

Python 3.7.11 was used for implementation, DGL-KE 0.1.2 (Zheng et al., 2020) for TransE, Transformers 4.19.4 (Wolf et al., 2020) for using the pre-training model, and Pytorch 1.10.0 (Paszke et al., 2019) for model creation. The link prediction evaluation was approximated using the neighborhood search library NGT (Iwasaki and Miyazaki, 2018).

Method	Ohsumed	HoC
BertGCN (Lin et al., 2021)	72.8	–
PubMedBERT (Gu et al., 2021)	–	82.32
BioLinkBERT (Yasunaga et al., 2022)	77.30	84.35
Ours	**78.08**	**84.50**

Table 5: Comparison of document classification results on the test set [%].

TransE was trained 50 epochs for both the literature graph and entity graph. As the representations of paper and entity nodes in the literature graph are initialized with BERT, the representation is 768-dimensional, so the representation of nodes to be randomly initialized was also initialized with 768 dimensions. Negative sampling, which randomly replaces h or t of the triple (h, r, t) on the graph, is used in TransE training. Only nodes with a target node type determined from the head node type and the relation type were used for negative sampling. A TransE training was conducted on an AMD Ryzen Threadripper 3990X 64-Core Processor as CPU and a GeForce RTX 3090 as GPU.

In the document classification model, the representation of the [CLS] token in BERT is classified with a linear layer. A dropout (Srivastava et al., 2014) was added before the linear layer to prevent overfitting on the training data. The document classification was performed on Intel(R) Xeon(R) CPU E5-2698 v4 and Intel(R) Xeon(R) W-3225 CPU as CPUs and Tesla V100-DGXS-32GB and A100. AdamW (Loshchilov and Hutter, 2019) was used as the optimizer for the document classification model.

5 Results

5.1 Representation learning

The results of link prediction on the literature graph are summarized in Table 3. Since the literature graph used in this study had a large number of nodes, both MAP@30 and Hit@N were generally low. The low performance of the relation between papers (Cites) and the relation between MeSHs (Hypernym, Supp) in particular may be due to the fact that TransE cannot represent them because there can be multiple tail nodes for a head node. Although the performance is low for link prediction, since the node representation is prepared for document classification, not link prediction, it would not be a critical problem if the representation cannot represent the multiple tail nodes.

The comparison of the link prediction results of the literature graph and entity graph is shown in Table 4. For the hypernym-hyponym relations (Hypernym), the performance was higher on the literature graph than on the entity graph. On the other hand, for the supplementary concept relations (Supp), the relations could be predicted only on the entity graph. MeSH consists of two types of concept records: descriptors and supplementary concept records. The descriptors play a central role in MeSH and have a tree structure. In addition, each paper from Medline is assigned only the descriptors as the entity representing the paper. On the other hand, supplementary records do not have a tree structure and are linked to descriptors. Therefore, in the literature graph, descriptors have more relationships than supplementary records. Thus the Hypernym relations, which are relationships between descriptors, may be taken into account more in the literature graph.

P_{lg}	E_{lg}	E_{eg}	Ohsumed	HoC
			76.79	83.57
✓			**77.45**	84.14
	✓		77.22	**84.58**
		✓	76.59	84.53
✓	✓		76.82	84.39
✓		✓	77.22	84.57
	✓	✓	75.79	83.49

Table 6: Document classification results on the development data set [%]. P_{lg} is the paper representation of the literature graph, E_{lg} is the entity representation from the literature graph, and E_{eg} is the entity representation from the entity graph.

5.2 Document classification

The comparison of our model with the existing models on the test set is summarized in Table 5. The scores are taken from the original papers, with the exception of the result of BioLinkBERT on Ohsumed and our results. For our model, we show the results with the best setting on the development set, i.e., adding P_{lg} for Ohsumed and E_{lg} for HoC. The tuning results on the development data sets

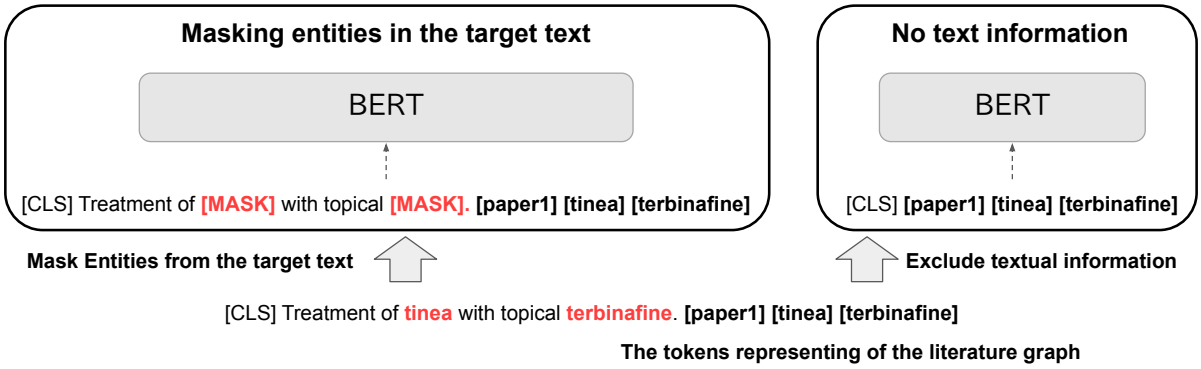

Figure 3: Two analysis settings. **Masking entities in the target text**: Evaluate the effectiveness of E_{lg} and E_{eg} by comparing settings with and without E_{lg} and E_{eg}. **No text information**: Exclude the target text from the BERT input and evaluate the performance only with the representations obtained from the graphs.

are shown in Table 6. The classification scores are higher than the previously reported scores, especially for Ohsumed.

As in Table 6, when we added each type of node representation from the literature graph separately, the performance improved over the baseline for both datasets. As for the entity representations, the entity representations from the literature graph are more effective than those from the entity graph. Especially on the Ohsumed dataset, adding the entity representations from the entity graph shows a negative effect on the performance. When we use multiple types of representations, the performance is not better than that of a single type of representations. The results show that the simultaneous use of multiple types of representations is not simple.

6 Analysis

To check the effects of the representation vectors of the literature graph and the entity graph alone, we conducted two experiments: document classification with masking the entities in the target text, which eliminates the impact of entity information in the text, and document classification with no text information, which eliminates the impact of paper and entity information in the text. The overview of these experiments is shown in Figure 3. The results of each experiment are shown in Table 7. Note that in these experiments, only the representation of the final layer of BERT corresponding to the [CLS] is used for classification.

Masking entities in the target text. As shown in Table 7, when the entities in the target text are masked, the performance is degraded for all settings. This may be because the entity information in the language model is not available. We can see the use of E_{lg} is effective by comparing the settings with or without E_{lg}. From this result, we can say that E_{lg} compensates for the missing entity information caused by masking entities. The performance with E_{eg} is lower than one with E_{lg}, which is consistent with the results in Table 6, suggesting that the E_{eg} representation is less suitable for document classification compared to E_{lg}. This may partly be because that E_{lg} takes into account the relationship of entities with documents instead of independently representing entities as for E_{eg}.

P_{lg}	E_{lg}	E_{eg}	Mask entities	No text
			79.91 ± 1.21	–
✓			79.22 ± 0.88	58.67 ± 0.62
	✓		**80.51 ± 1.06**	62.57 ± 2.13
		✓	79.10 ± 0.29	62.59 ± 2.36
✓	✓		80.32 ± 1.81	67.72 ± 1.36
✓		✓	79.41 ± 0.95	65.36 ± 1.70
	✓	✓	79.35 ± 1.21	61.43 ± 2.87
✓	✓	✓	80.29 ± 1.33	**67.97 ± 2.48**

Table 7: The results of document classification [%] on the development data set of HoC with masking entities (Mask entities) and without textual information (No text)

.

No text information. To evaluate the performance only with the representations obtained from the graphs, we excluded the target text from the BERT input, i.e., only paper and entity representations were used as input to BERT. We found that even the paper and entity representations from graphs alone could classify documents to some extent despite the lack of text information. The performance of using both the paper and entity representations was higher than that of using each representation. These

results show that the paper and entity information from literature and entity graphs are all helpful in classifying documents, although combining them with text information is not always helpful.

7 Conclusions

In this study, for the purpose of document classification that can use multiple types of information at the same time, we proposed a document classification model that creates a representation vector from a literature graph that contains a lot of information, such as bibliographic and entity information, considering various relationships between documents, and incorporates the representation vector and textual information about the documents. Experimental results on two datasets, Ohsumed and HoC, show that both models improve performance with the information from the literature graph, and the models show state-of-the-art performance on both datasets. We also found that the performance degraded when we simultaneously used multiple information of paper and entity, showing that the incorporation of different types of information is not simple.

In the future, we will investigate representation learning of large-scale literature graphs using methods such as GCN to obtain better representations. We will also explore methods for simultaneously learning representations of literature graphs and document classification.

Limitations

Our proposed model has three limitations. Firstly, because of the large size of the literature graph in this study, representation learning is performed on the literature graph using TransE, but in fact, there are relations in the literature graph that cannot be represented by TransE as shown in Section 5.1. To overcome this issue, more expressive methods such as RotatE (Sun et al., 2019) and GCN (Kipf and Welling, 2017) could be investigated. These methods are expected to be able to represent complex relationships associated with multiple types of external information. Secondly, our model adds a representation of the literature graph to the target text, so longer sentences require truncation of textual information. Thirdly, we have not analyzed the results in detail and how the proposed method positively and negatively impacted document classification. We leave these limitations for future work.

Acknowledgements

This work was supported by JSPS Grant-in-Aid for Scientific Research JP20K11962.

References

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2015. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

William L Hamilton. 2020. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159.

Masajiro Iwasaki and Daisuke Miyazaki. 2018. Optimization of indexing based on k-nearest neighbor graph for proximity search in high-dimensional data. *CoRR*, abs/1810.07355.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML'98, page 137–142, Berlin, Heidelberg. Springer-Verlag.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

National Library of Medicine. 2020. Medical subject headings - home page. [Online; accessed 2023-04-11].

National library of Medicine. 2020. Pubmed. [Online; accessed 2023-04-11].

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. 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, Online. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. Proceedings of the AAAI conference on artificial intelligence.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

Da Zheng, Xiang Song, Chao Ma, Zeyuan Tan, Zihao Ye, Jin Dong, Hao Xiong, Zheng Zhang, and George Karypis. 2020. Dgl-ke: Training knowledge graph embeddings at scale. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 739–748, New York, NY, USA. Association for Computing Machinery.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Tuning results on the development data set

The tuning results on document classification on the development data set are shown in Table 8. The representations used for the inputs were compared as in Table 6, and tuning was performed to determine which combination of the representations of the final layer of BERT corresponding to the [CLS] token and the paper node, as well as the paper representation of the literature graph is used for classification using the FC layer. We also tuned whether the representations used for classification should be concatenated (concat) or pooled with max pooling.

Input			Output					
P_{lg}	E_{lg}	E_{eg}	$\mathbf{h}_{[\text{CLS}]}$	P_{lg}^{L}	P_{lg}		HoC	Ohsumed
			✓			–	83.43 ± 0.70	75.89 ± 0.88
			✓		✓	concat	**83.57 ± 0.58**	**76.79 ± 0.36**
			✓		✓	max pooling	83.06 ± 0.45	75.74 ± 0.21
✓			✓			–	83.83 ± 0.58	76.72 ± 0.29
✓				✓		–	**84.14 ± 0.91**	76.29 ± 1.11
✓			✓	✓		concat	83.67 ± 0.93	77.28 ± 0.77
✓			✓	✓		max pooling	83.37 ± 0.44	76.79 ± 1.31
✓			✓		✓	concat	83.70 ± 0.49	77.02 ± 0.30
✓			✓		✓	max pooling	83.54 ± 0.44	75.79 ± 0.53
✓				✓	✓	concat	83.64 ± 0.69	77.25 ± 0.80
✓				✓	✓	max pooling	83.41 ± 0.67	76.46 ± 1.20
✓			✓	✓	✓	concat	84.13 ± 0.95	**77.45 ± 0.55**
✓			✓	✓	✓	max pooling	83.09 ± 1.15	76.06 ± 0.15
	✓		✓			–	**84.58 ± 0.84**	75.86 ± 0.70
	✓		✓		✓	concat	83.29 ± 1.03	**77.22 ± 0.80**
	✓		✓		✓	max pooling	83.81 ± 0.59	75.56 ± 1.45
		✓	✓			–	**84.53 ± 0.48**	75.83 ± 1.09
		✓	✓		✓	concat	83.63 ± 1.01	**76.59 ± 0.91**
		✓	✓		✓	max pooling	83.54 ± 0.42	75.93 ± 1.20
✓	✓		✓			–	84.23 ± 1.28	76.29 ± 0.95
✓	✓			✓		–	83.80 ± 1.16	**76.82 ± 0.94**
✓	✓		✓	✓		concat	84.20 ± 0.71	75.83 ± 1.18
✓	✓		✓	✓		max pooling	83.56 ± 0.59	75.10 ± 1.89
✓	✓		✓		✓	concat	84.01 ± 0.35	76.09 ± 0.55
✓	✓		✓		✓	max pooling	82.94 ± 0.78	75.79 ± 0.95
✓	✓			✓	✓	concat	83.91 ± 0.79	75.69 ± 0.69
✓	✓			✓	✓	max pooling	83.95 ± 1.17	74.60 ± 3.12
✓	✓		✓	✓	✓	concat	**84.39 ± 1.03**	75.20 ± 2.21
✓	✓		✓	✓	✓	max pooling	83.78 ± 0.91	73.48 ± 1.89
✓		✓	✓			–	83.92 ± 1.29	76.03 ± 0.06
✓		✓		✓		–	**84.57 ± 0.86**	75.76 ± 0.66
✓		✓	✓	✓		concat	83.67 ± 0.83	73.74 ± 2.61
✓		✓	✓	✓		max pooling	83.68 ± 0.42	75.23 ± 0.97
✓		✓	✓		✓	concat	83.74 ± 0.45	77.12 ± 0.77
✓		✓	✓		✓	max pooling	82.98 ± 0.53	76.79 ± 0.60
✓		✓		✓	✓	concat	84.20 ± 0.85	**77.22 ± 0.29**
✓		✓		✓	✓	max pooling	82.51 ± 1.15	75.83 ± 0.50
✓		✓	✓	✓	✓	concat	83.67 ± 1.16	76.49 ± 1.05
✓		✓	✓	✓	✓	max pooling	81.90 ± 0.46	73.81 ± 1.54
	✓	✓	✓			–	83.42 ± 1.09	75.36 ± 0.55
	✓	✓	✓		✓	concat	**83.49 ± 1.11**	**75.79 ± 0.65**
	✓	✓	✓		✓	max pooling	82.11 ± 1.22	74.80 ± 1.33
✓	✓	✓	✓			–	83.31 ± 1.23	75.30 ± 0.55
✓	✓	✓		✓		–	83.07 ± 2.72	75.17 ± 1.56
✓	✓	✓	✓	✓		concat	**83.54 ± 0.38**	74.54 ± 0.86
✓	✓	✓	✓	✓		max pooling	83.52 ± 1.61	74.27 ± 1.13
✓	✓	✓	✓		✓	concat	83.27 ± 0.50	75.93 ± 0.40
✓	✓	✓	✓		✓	max pooling	82.83 ± 1.22	73.54 ± 1.35
✓	✓	✓		✓	✓	concat	83.24 ± 1.48	75.86 ± 1.04
✓	✓	✓		✓	✓	max pooling	82.18 ± 1.22	73.81 ± 1.86
✓	✓	✓	✓	✓	✓	concat	83.48 ± 1.35	**75.96 ± 0.80**
✓	✓	✓	✓	✓	✓	max pooling	83.29 ± 1.62	73.02 ± 0.26

Table 8: Tuning results on the development data set [%]. P_{lg} adds the paper representation of the literature graph, E_{lg} adds the entity representation of the literature graph, and E_{eg} adds the entity representation of the entity graph. $\mathbf{h}_{[\text{CLS}]}$ and P_{lg}^{L} are the representations of the final layer of BERT that correspond to the [CLS] token and the token representing the paper of the literature graph.