HGCN4MeSH: Hybrid Graph Convolution Network for MeSH Indexing

Miaomiao Yu **Chenhui Li** **Yujiu Yang** *
Tsinghua Shenzhen International Graduate School, Tsinghua University, China
{yumm18,lch17}@mails.tsinghua.edu.cn
yang.yujiu@sz.tsinghua.edu.cn

Abstract

Recently, deep learning has been used in *Medical Subject Headings* (MeSH) indexing to reduce the labor costs associated with manual annotation, including DeepMeSH, TextCNN, etc. However, these models fail to capture the complex correlations between MeSH terms. To this end, we use a Graph Convolution Network (GCN) to learn the relationship between these terms and present a novel Hybrid Graph Convolution Net for MeSH index (HGCN4MeSH). We utilize two bidirectional GRUs to learn the embedding representation of the abstract and the title of the MeSH index text respectively. We construct the adjacency matrix of MeSH terms, based on the co-occurence relationships in corpus, and use the matrix to learn representations using the GCN. On the basis of learning the joint representation, the prediction problem of the MeSH index keywords is an extreme multi-label classification problem after the attention layer operation. Experimental results on two datasets show that HGCN4MeSH is competitive with the state-of-the-art methods.

1 Introduction

MEDLINE[1] is an important database for publications of biomedical and life science containing more than 24 million journal citations. To facilitate information storage and retrieval, the National Library of Medicine (NLM) created *Medical Subject Headings* (MeSH)[2] to index articles in MEDLINE. MeSH is an annually-updated hierarchical glossary. There are 29368 concepts[3] of MeSH in 2019, covering various area from biomedicine to information technology. Currently, the articles in MEDLINE are indexed primarily by NLM human experts. It is estimated that it costs millions of dollars each year to index new articles (Mork et al., 2013). Therefore, it is necessary to build an efficient and accurate model for indexing documents — MeSH index.

Example1: [Animals, Blotting Western, Body, Weight, Heme, Oxygenase1, Male, Mice ,Mice Obese, Motor, Activity, Oxygen, Consumption, Protoporphyrins, Receptor Melanocortin Type 4, Thermogenesis, Weight]
Example2: [Animals, Blotting Western, Cell Hypoxia, Cell Line, Cell Survival, Cells Cultured, E2F1 Transcription Factor, Hepatocytes, Hypoxia-Inducible Factor 1 alpha Subunit, Membrane Proteins, Mice, Mice Inbred C57BL, Mitochondrial Proteins, RNA Small Interfering]
Example3: [Animals, Appetite Regulation, Energy Metabolism, Fats, Feedback Physiological, Glucose, Humans, Intestine Small, Signal Transduction]

Table 1: Examples of tags from article 26815432, 27391842, 26736497 in MEDLINE. It can be seen that when the tag 'Mice' appears, tag 'Animals' is likely to appear. However, when tag 'Animals' appears, the tag 'Mice' does not necessarily appear.

Xun et al. (2019) demonstrated that the MeSH indexing problem can be cast as an extreme multi-label classification task. Each MeSH term can be regarded as a tag, with a total of 29368 tags, and each article has an average of 13 tags. Recently, there are some deep learning models applied to MeSH terms indexes successfully, such as AttentionMeSH (Jin et al., 2018), MeSHProbeNet (Xun et al., 2019), etc. However, these models do not considered the correlation and the co-occurrence relationship between MeSH terms. By ignoring the complexity between objects, these methods are inherently limited. Table 1 is a real example of article tags from the data.

In this paper, we propose a novel GCN (Kipf and Welling, 2016)-based MeSH term index model, HGCN4MeSH, which learns the co-occurrence representation of tags via a GCN-based mapping function. Specifically, we design a novel data-driven adjacency matrix to guide the information propagation between nodes. To solve the problem of too many tags in extreme multi-label classification

*The corresponding author.
[1]https://www.nlm.nih.gov/bsd/medline.html
[2]https://www.nlm.nih.gov/mesh/meshhome.html
[3]https://www.nlm.nih.gov/databases/download/mesh.html

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 20–26

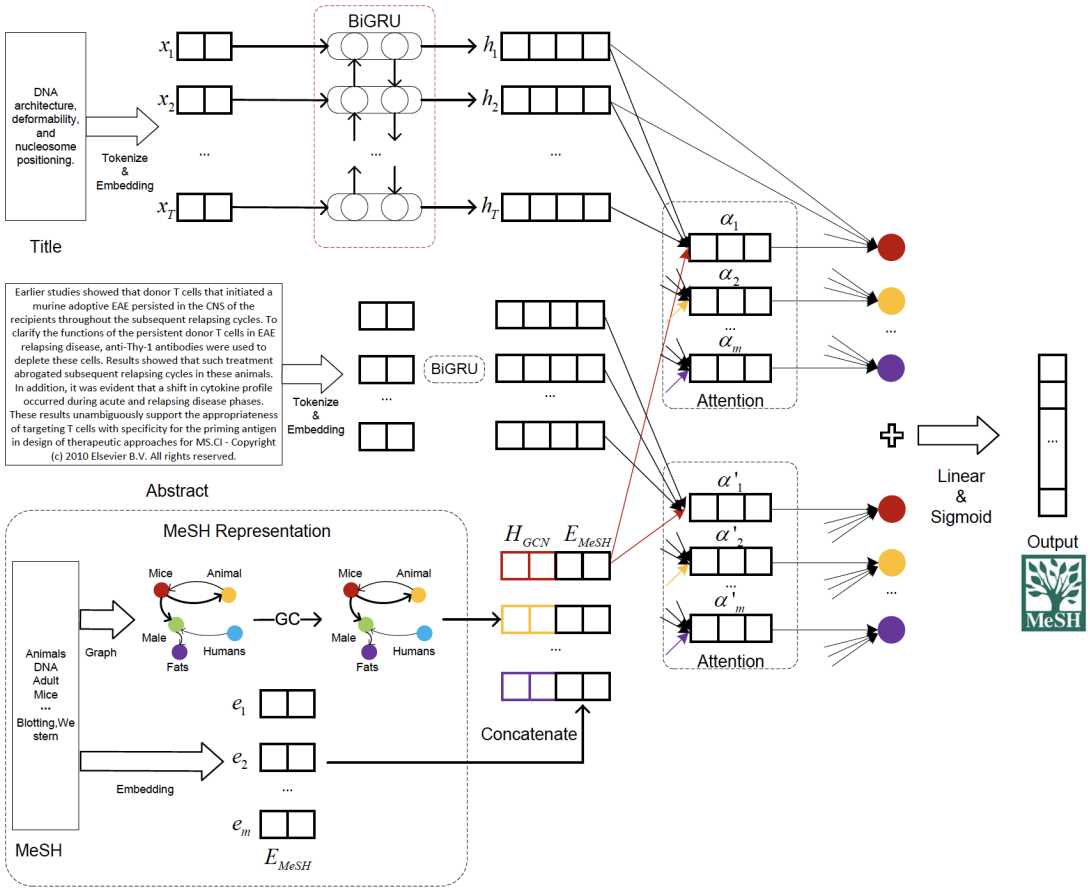

Figure 1: The proposed model framework. Balls of various sizes and colors represent different representations of MeSH terms, BiGRU is the bidirectional gated recurrent unit. First, A hybrid graph is constructed for MeSH terms, where each node represents a MeSH term. The abstract and title are input into GRU for feature extraction respectively and GCN updates the representation of MeSH terms by learning co-occurrences of MeSH terms during training. The final representation of MeSH terms consists of two parts, one is the representation generated by GCN, the other is the semantic representation of MeSH terms. Then we can calculate the attention weight between MeSH terms and title; abstract, output the final score via a linear layer and a sigmoid activation function.

cases, we propose a hybrid adjacency matrix, that is, constructing a bidirectional GCN between high-frequency tags and a unidirectional GCN between high-frequency and low-frequency tags to reduce the computation. The major contribution are:

- We propose a novel end-to-end extreme multi-label classification framework (Figure 1), which employs a GCN to learn tags representation.
- We utilize a partial block adjacency matrix to reduce calculation and noise for extreme multi-label classification. The experimental results show that our method is competitive with the state-of-the-art method.

2 Related Work

Aronson et al. (2004) introduced the Medical Text Index (MTI) to help experts find suitable MeSH terms for articles quickly and accurately. Peng et al. (2016) proposed DeepMeSH, which achieved the best results in the 2017 BioASQ challenge task A. BioASQ is a challenge funded by the European Union; the task A of BioASQ requires participants to use only the abstracts and titles to predict corresponding MeSH terms. DeepMeSH utilized TF-IDF (Jones, 1972) and document to vector (D2V) (Le and Mikolov, 2014) to represent each abstract and They used k-nearest-neighbor (KNN) (Altman, 1992) classifiers to generate candidate MeSH terms. AttentionMeSH (Jin et al., 2018) was also divided into two parts. The first part used KNN to generate candidate MeSH terms, and the second used bidirectional Recurrent Gated Unit (BiGRU) (Cho et al., 2014) architecture to capture context features. Xun et al. (2019) used

the representation learned from the name of journal combine with the information from the abstract and a multi-view neural classifier to get results. Wang and Mercer (2019) provided a useable data set, including the title, abstract, paragraphs associated with the figures, and tables of each text, and used multi-channel TextCNN (Kim, 2014) to solve the problem.

MeSH terms were modelled independently in those methods, which ignored the relationships between MeSH terms. In this paper, we use a GCN to capture the more complex topological relationships.

3 HGCN4MeSH Model

3.1 Graph Convolutional Network and Correlation Matrix

We use Graph Convolutional Network (GCN) to model the relationship between MeSH terms. Kipf and Welling (2016) proposed GCN which induces embedding vectors of the nodes according to the properties of their neighbor nodes. Given a graph G = (V,E) where V and E denote the set of nodes and edges respectively. The GCN is a multi-layer neural network. With convolutional operations, the propagation of every layer can be written as

$$H^{l+1} = h(\tilde{A} \cdot H^{l} \cdot W^{l}). \tag{1}$$

Here, $H^l \in \mathbb{R}^{n \times d}$ and $H^{l+1} \in \mathbb{R}^{n \times d'}$ indicate the nodes representation of the l^{th} and $(l+1)^{th}$ hidden layer respectively (where n is the number of nodes and d, d' are the dimensions of the node representations), $\tilde{A} \in \mathbb{R}^{n \times n}$ represents the normalized version of the correlation matrix $A \in \mathbb{R}^{n \times n}$, $h(\cdot)$ means a non-linear operation such as ReLU, $\cdot$ means the matrix product operation, $W^l \in \mathbb{R}^{d \times d'}$ is a layer-specific trainable transformation matrix.

GCN updates the node features by propagating the information between neighbor nodes, based on the corresponding correlation matrix. Hence, the crucial thing is how to build the adjacency matrix. In most applications, the adjacency matrix is predefined. However, there is no corresponding adjacency matrix already defined in the area of extreme multi-label text classification. Facing this problem, we propose the hybrid adjacency matrix construction method. We construct the adjacency matrix between tag frequencies and the co-occurrence relationships between tags.

In extreme multi-label text scenarios, the number of tags is often in the tens of thousands. If we consider the relationship between all the tags, the adjacency matrix would be huge and consume considerable memory and time during the computation. Considering that in the extreme multi-label classification task, the distribution of tags is long-tailed, which means that there are some tags appear rarely, hence $\tilde{A}$ is a sparse matrix.

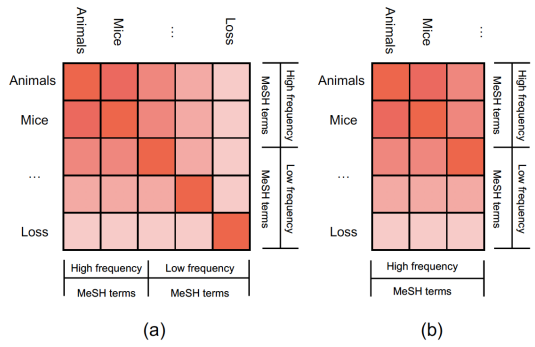

Figure 2: The construction of adjacency matrix. (a) the adjacency matrix of original GCN ($m \times m$) (b) the hybrid adjacency matrix of our model ($m \times n$)

Hence, we set a threshold frequency to divide tags into low-frequency and high-frequency groups. We find that the number of low-frequency tags co-occurring with high-frequency tags is larger than the number of low-frequency tags co-occurring with low-frequency tags through empirically. Thus, we build an adjacency matrix $\tilde{A} \in \mathbb{R}^{m \times n}$, where m is the number of the high-frequency tags and n denotes the total number of tags. It means that we utilize the information between high-frequency tags and low-frequency tags, so it is called hybrid adjacency matrix. Figure 2 shows the example of adjacency matrix. We use the empirical conditional probability to model the directed relationship between tags:

$$p(L_j|L_i) = \frac{M_{ij}}{N_i} \tag{2}$$

which means the occurrence probability of tag L_j when tag L_i appears, where N_i denotes the occurrences times of the tag L_i, and M_{ij} denotes the concurring times of tag L_i and tag L_j.

$$P_{ij} = p(L_j|L_i) \tag{3}$$

However, due to a large number of tags, these co-occurrencesmay be noisy estimate for some tags with low co-occurrence frequency, so we set a threshold τ as follows:

$$A_{ij} = \begin{cases} P_{ij} & P_{ij} > \tau \\ 0 & P_{ij} \leq \tau \end{cases} \tag{4}$$

3.2 Document Representation

The core challenging in MeSH idnexing is to learn representations for the title and abstract. After tokenizing the titles and abstracts, we derive the context-aware title representation via a bidirectional Gated Recurrent Unit (BiGRU) (Cho et al., 2014):

$$\begin{aligned} H_{title} &= \text{BiGRU}(X_{title}) \in \mathbb{R}^{L \times 2d_h} \\ H_{abstract} &= \text{BiGRU}(X_{abstract}) \in \mathbb{R}^{L' \times 2d_h} \end{aligned} \tag{5}$$

where H_{title}, $H_{abstract}$ mean the hidden state of title, abstract respectively. $X_{title} \in \mathbb{R}^{L \times d_e}$, $X_{abstract} \in \mathbb{R}^{L' \times d_e}$ denote the feature of title, abstract respectively (d_e means the embedding dimension of word), L is the length of title, L' is the length of title, d_h is the hidden layer dimension. In this work, the title and the abstract share the same process.

3.3 MeSH Representation

First, we use the corresponding word embedding of all MeSH terms as the initial input (H_0) to GCN. In section 3.1, we introduced a novel adjacency matrix A, we can get the new representation of MeSH terms with co-occurrence information after multi-layers of GCN.

$$H_{GCN} = \sigma(\tilde{A} \cdot H^l \cdot W^l) \tag{6}$$

where $H^l \in \mathbb{R}^{m \times d_l}$ is the high-frequency MeSH terms representation of l^{th} layer, $\tilde{A}$ is the normalized version of adjacency matrix and W^l is a layer-specific trainable transformation matrix. In other words, only the representations of high-frequency MeSH terms are propagated at each layer in GCN. After getting the representation of MeSH terms interrelation by GCN, we also use the embedding of MeSH terms to retain the semantic information.

$$H_{MeSH} = [H_{GCN} : e_{MeSH}] \tag{7}$$

where the symbol : means the concatenated operation; e_{MeSH} is the word embedding of MeSH terms.

Now we can utilize MeSH representations to select the most relevant text representation features for classification by attention mechanism (Bahdanau et al., 2014). We calculate the similarity between MeSH terms and text by dot products and use Softmax to normalize the word axis:

$$\begin{aligned} Sim &= H_{title} \cdot H_{MeSH} \\ A_{attn} &= softmax(Sim) \end{aligned} \tag{8}$$

Ultimately, we can get the representation of MeSH terms by words representation:

$$H'_{MeSH} = A_{attn} H_{title} + A'_{attn} H_{abstract} \tag{9}$$

where A'_{attn} is the attention score between abstract and MeSH terms, and $H_{abstract}$ is the hidden state of abstract. Then we can gain the score of MeSH terms:

$$\hat{y} = \sigma(W H'_{MeSH} + b) \tag{10}$$

here, $\sigma(\cdot)$ is the sigmoid function, W is the trainable weight matrix and b is the bias. The binary cross-entropy loss function is applied in the model:

$$L_j = -(y_j log(\hat{y}_j) + (1 - y_j) log(1 - \hat{y}_j)) \tag{11}$$

where y_j is the ground truth, $\hat{y}_j \in [0, 1]$. The total loss is:

$$\mathcal{L} = \frac{1}{K} \sum_{j=1}^{K} L_j \tag{12}$$

Here, K is the total number of training data.

Finally, the MeSH multi-label classifier outputs the MeSH index that we want.

4 Experiments

4.1 Dataset

PMC Collection contains 257590 manually annotated biomedical articles and covers 22881 MeSH terms in total. Each documents contains 13.34 MeSH terms on average.

SETC2015 contains 14828 annotated articles created by Demner-Fushman and Mork (2015). Wang and Mercer (2019) used this dataset to create a new dataset, which covers 14365 MeSH terms and contains 13.15 MeSH terms per document.

4.2 Implementation Details

In the processing, non-English characters are removed. The embedding dimensions of title and abstract are both 200, GRU layer number is set to 2, and the hidden dimension is 200. In the part of GCN, we use a layer of GCN with both input and output dimensions of 200. LeakyReLU (Maas et al., 2013) with a negative slope of 0.2 is used as the non-linear activation function. For the division of word frequency, we choose the high-frequency MeSH terms with more than 1000 occurrences, the low-frequency MeSH terms with less than 1000 of the PMC Collection dataset. For SETC2015 dataset, the threshold is 500. We set τ in Eq.(4)

	$p@k$					$nDCG@k$		
	$p@1$	$p@3$	$p@5$	$p@10$	$p@15$	$nDCG@1$	$nDCG@3$	$nDCG@5$
PMC Collection								
multichannel TextCNN	0.8791	0.7214	0.6148	0.5179	0.4801	0.8791	0.7574	0.6752
HGCN4MeSH-1	0.9145	0.8250	0.7417	0.5773	0.4618	0.9145	0.8463	0.7832
HGCN4MeSH	**0.9267**	**0.8495**	**0.7707**	**0.6124**	**0.4953**	**0.9267**	**0.8677**	**0.8086**
SETC2015								
multichannel TextCNN	0.8051	0.6298	0.5206	0.4196	0.3959	0.8051	0.6698	0.5841
HGCN4MeSH-1	0.9054	0.7841	0.6921	0.5415	0.4450	0.9054	0.8124	0.7411
HGCN4MeSH	**0.9185**	**0.7930**	**0.7078**	**0.5581**	**0.4563**	**0.9185**	**0.8221**	**0.7555**

Table 2: Results for our Model in p@k and nDCG, HGCN4MeSH-1 is the model using the embedding of MeSH terms merely without GCN, HGCN4MeSH is the model with GCN

to be 0.1. Dropout (Srivastava et al., 2014) is 0.2, and learning rate 0.0005. Besides, we apply the Adam optimizer (Kingma and Ba, 2014) and early stopping strategies (Yao et al., 2007). The model is implemented with PyTorch (Paszke et al., 2017).

4.3 Evaluation Metrics

Due to the large space of the tags, only a few tags can match the text. Hence, the major metrics for performance evaluation are ranking-based methods.

Precision at k (p@k) and normalized discounted cumulative gain (nDCG) are ranking-based evaluation methods. In this paper, we also utilize these two authoritative metrics.

4.4 Experiments Results

Table 2 shows the rank-based matric result. Although there are some strong baselines of bioASQ challenge, the code is available to test on the two dataset. We compare with the state-of-art method, multichannel TextCNN (Wang and Mercer, 2019). For the proposed model, we report the results of the model with GCN or not. It is obvious that our model without GCN outperforms baseline, and the performance of the model with GCN is the best result, which may due to the fact that the model with GCN pays more attention to the co-occurrence relationships between the tags.

In addition, the score of the PMC Collection dataset increases by about 2-4 points after introducing GCN. However, the score of SETC2015 only increases by 1-2 points. The reason is that there are only 14000 samples of SETC2015. Thus the data-driven adjacency matrix is biased. Nevertheless, since the PMC Collection dataset contains about 250000 data, the adjacency matrix based on the dataset should be closer to the true co-occurrence relationship between the MeSH terms, and results to better performance.

Model		**p@k**			
l	f	$p@1$	$p@3$	$p@5$	$p@10$
1	0.5k	0.9116	0.8345	0.7597	0.6029
1	1k	**0.9267**	**0.8495**	**0.7707**	**0.6124**
1	1.5k	0.9185	0.8409	0.7518	0.6103
4	2k	0.9174	0.8359	0.7618	0.6046

Table 3: The result of MeSH terms on testing set for different frequency threshold. l is the GCN layer, f is the frequency threshold, f=1k means MeSH terms with less than 1000 occurrences is low-frequency tag, and those with more than 1000 occurrences are high-frequency tags.

Model		**p@k**			
l	f	$p@1$	$p@3$	$p@5$	$p@10$
1	1k	**0.9267**	**0.8495**	**0.7707**	**0.6124**
2	1k	0.9094	0.8323	0.7577	0.6008
3	1k	0.9170	0.8285	0.7494	0.5945

Table 4: The result of MeSH terms for different GCN layers. l=1 means the GCN layer is 1.

4.5 Ablation Studies

In the Table 3, we can observe effects of thresholds that define low-frequency MeSH terms and high-frequency MeSH terms. If the threshold is too high, it may cause fewer high-frequency MeSH terms, which causes the representation between different MeSH terms to be too smooth. However, when the frequency threshold is too low, there are many high-frequency words, and some co-occurrence of many words may become noise.

Table 4 shows that with the number of GCN layers increasing, the results decrease. As the number of GCN layers increasesthe information transmission between nodes may accumulate, resulting in excessive smoothness of the final representation.

Model	$p@1$	$p@3$	$p@5$	$p@10$
w/o atten	0.8897	0.7978	0.7235	0.5531
w/o GCN	0.9145	0.8250	0.7417	0.5773
w/o title	0.9094	0.8351	0.7589	0.5984
w/o abs	0.8763	0.7857	0.7050	0.5569
title&abs	0.9082	0.8361	0.7621	0.6058
ours	**0.9267**	**0.8495**	**0.7707**	**0.6124**

Table 5: The result of ablation studies. w/o: without; atten: attention; abs: abstract; ours:HGCN4MeSH; title&abs: title and abstract are concatenated as the input of GRU.

The results of the ablation experiment are shown in Table 5. Title contains a lot of useful information, the effect of extracting information from title and abstract separately is slightly better than directly concatenating both.

5 Conclusion

Modelling the relationship between MeSH terms is a key issue in MeSH indexing. This paper proposes a model for constructing specifying the relationship between MeSH terms based on GCN and a new end-to-end model for MeSH indexing.

In the field of biomedicine, the co-occurrence relationship of tags is very common and useful. We use the co-occurrence relationship between tags to design the adjacency matrix by the GCN using the data-driven method, which can also be extended to other extreme multi-label classification fields.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No. 2018YFB1601102), and Shenzhen special fund for the strategic development of emerging industries (No. JCYJ20170412170118573). In addition, we would like to thank Dr. Roy Schwartz; Dr. Rishi Bommasani and the anonymous reviewers for thoughtful feedback and constructive suggestions.

References

Naomi S Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Alan R Aronson, James G Mork, Clifford W Gay, Susanne M Humphrey, and Willie J Rogers. 2004. The nlm indexing initiative's medical text indexer. In *Medinfo*, pages 268–272.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Dina Demner-Fushman and James G Mork. 2015. Extracting characteristics of the study subjects from full-text articles. In *AMIA Annual Symposium Proceedings*, volume 2015, page 484. American Medical Informatics Association.

Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. 2018. Attentionmesh: Simple, effective and interpretable automatic mesh indexer. In *Proceedings of the 6th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering*, pages 47–56.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

James G Mork, Antonio Jimeno-Yepes, and Alan R Aronson. 2013. The nlm medical text indexer system for indexing biomedical literature. In *BioASQ@ CLEF*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Shengwen Peng, Ronghui You, Hongning Wang, Chengxiang Zhai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2016. Deepmesh: deep semantic representation for improving large-scale mesh indexing. *Bioinformatics*, 32(12):i70–i79.

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.

Xindi Wang and Robert E Mercer. 2019. Incorporating figure captions and descriptive text in mesh term indexing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 165–175.

Guangxu Xun, Kishlay Jha, Ye Yuan, Yaqing Wang, and Aidong Zhang. 2019. Meshprobenet: a self-attentive probe net for mesh indexing. *Bioinformatics*.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.