NIP-GCN: An Augmented Graph Convolutional Network with Node Interaction Patterns

Manish Chandra
manish.chandra@iitkgp.ac.in
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India

Debasis Ganguly*
Debasis.Ganguly@glasgow.ac.uk
University of Glasgow
Glasgow, Scotland

Pabitra Mitra
pabitra@cse.iitkgp.ac.in
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India

Bithika Pal
bithikapal@iitkgp.ac.in
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India

James Thomas
james.thomas@ucl.ac.uk
University College London
London, United Kingdom

ABSTRACT

In this paper, we propose an augmented Graph Convolutional Network (GCN) mechanism wherein additional information of local interaction patterns between a node with its neighbors (specifically, in the form of distribution of cosine similarity values of a pre-trained node vector with its neighbors) is used to enrich a node's representation prior to training a GCN. This provides additional information about the structural properties of a node, which the standard convolution operation in a GCN can then leverage for obtaining potentially improved effectiveness in a down-stream task. Our experiments demonstrate that adding these node interaction patterns (NIPs) along with an additional noise-contrastive pairwise document similarity objective within a GCN improves the linked document classification task.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning by classification**.

KEYWORDS

Graph Convolution, Linked-document Classification

ACM Reference Format:
Manish Chandra, Debasis Ganguly, Pabitra Mitra, Bithika Pal, and James Thomas. 2021. NIP-GCN: An Augmented Graph Convolutional Network with Node Interaction Patterns. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463082

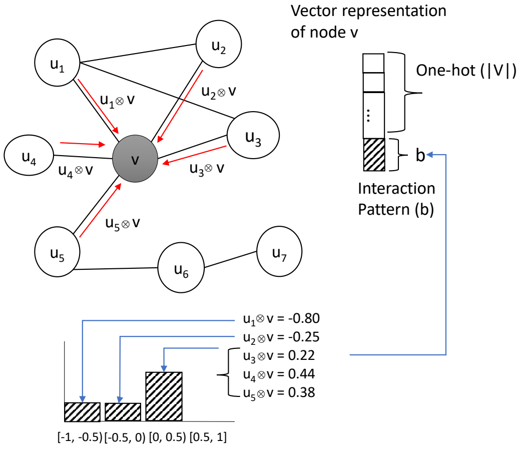


Figure 1: A schematic illustration of incorporating node interaction patterns to initialize a Graph Convolutional Network (GCN). The figure shows how the distribution of the cosine similarity values between a node v and its neighbors u_i, $u_i \otimes v$ (obtained with a node embedding algorithm such as node2vec [5]), are discretized into a fixed length vector (a histogram), and then appended to v's initial 1-hot vector representation.

1 INTRODUCTION

Graphs are one of the most expressive data-structures which have been used to model a variety of problems. Traditional neural network architectures such as the Convolutional Neural Networks and Recurrent Neural Networks are constrained to handle only metric data. Recently, Graph Convolutional Networks (GCNs) [1, 3, 9] have been proposed to address this shortcoming. It has been successfully applied to several domains such as social networks [7], knowledge graphs [17], and natural language processing [14]. In the GCN framework, a collection of linked documents can be represented as a heterogeneous graph with documents and words as nodes. Containment edges between documents and word nodes embody the bag of words representations of documents, while document-document citation edges represent document linkage information. With reference to document classification, information from a document's content as well its inter-relationships with other documents

[*] Work done at the author's previous affiliation - IBM Research Dublin.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463082

has been shown to provide useful cues [9, 19, 22] for document classification.

A standard approach of text classification with GCN [22] initializes each node representation as independent of each other (specifically, one-hot orthogonal representations of the nodes) and then relies on the convolution operators and the known class labels to learn abstract representations of the nodes. However, in our work, we incorporate prior information into each node of a GCN in the form of node interaction patterns, so that the graph convolution layers can harness this additional structural cues of the nodes ito its learning phase.

Interaction-focused models are reported to be useful for the ad-hoc retrieval task [6, 8, 13, 16]. An interaction based approach computes the histogram of similarities between the vector representations of word pairs in a query and the retrieved documents. These histograms are then used to train a ranking model. In our work, we exploit the same idea by computing the histogram of the local interactions of a node's dense vector representation with its neighbors, and then using it as an additional information to initialize a GCN (see Figure 1). Since this histogram mapping expresses a node's local structure, we hypothesize that this augmented representation of a node may lead to a more effective learning of a down-stream task via GCN.

2 RELATED WORK

Graph Convolution Networks. GCNs, introduced in [1] and then made scalable through efficient localized filters [3], follows a message passing framework [4] for node aggregation. A broad survey on GCNs is presented in [20]. In Text-GCN, an approach that applied GCN for text classification [22], an entire corpus is modelled as a heterogeneous graph comprising of nodes each corresponding to a document or a word. This formulation allowed provision to learn word and document embedding jointly with GCN.

Interaction Modeling. Existing deep matching models for document retrieval can broadly be categorized into two major types - *representation*-focused, and *interaction*-driven. ARC-II is an interaction-focused model [8] which involves learning hierarchical matching patterns from local interactions using a CNN. DRMM [6], on the other hand, first builds local interactions between each pair of terms from a query and a document based on their embedded vector representations. For each query term, it then maps the variable-length local interactions into a fixed-length matching histogram, which is then supplied as an input to a feed-forward network to learn a similarity score. The KNRM replaces the histogram computation of DRMM with kernel smoothing operations along each row of the query-document interaction matrix [21].

3 PROPOSED METHODOLOGY

Background. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{\mathcal{X}})$, where $\mathcal{V}$ is the set of vertices, $\mathcal{E}$ is the set of edges and $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{|\mathcal{V}| \times d_0}$ represents d_0-dimensional input features of each node. The node representation obtained from a single GCN layer is defined as: $\mathbf{H} = f(\hat{\mathbf{A}}\boldsymbol{\mathcal{X}}\mathbf{W})$. Here, $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is the normalized adjacency matrix with added self-connections and $\tilde{\mathbf{D}}$ is defined as $\tilde{\mathbf{D}}_{\mathbf{ii}} = \sum_j (\mathbf{A}+\mathbf{I})_{ij}$. The model parameter is denoted by $\mathbf{W} \in \mathbb{R}^{d_0 \times d_1}$ and f is an activation function. The GCN representation $\mathbf{H}$ encodes the immediate neighborhood of each node in the graph. For capturing multi-hop dependencies in the graph, several GCN layers can be stacked, one on the top of another as follows: $\mathbf{H}^{k+1} = f(\hat{\mathbf{A}}\mathbf{H}^k\mathbf{W}^k)$, where k denotes the layer number, $\mathbf{W}^k \in \mathbb{R}^{d_k \times d_{k+1}}$ is layer-specific parameter and $\mathbf{H}^0 = \boldsymbol{\mathcal{X}}$.

Graph Creation. We create the graph, by the methodology similar to [22]. The constructed graph models a) the containment relationships of words in documents, b) window-based co-occurrences between words, and c) the citation relationship between documents. The weight of an edge between nodes i and j is specified as:

$$\mathbf{A}_{ij} = \begin{cases} \mathrm{PMI}(i,j) & \text{if } i, j \text{ are words, } \mathrm{PMI}(i,j) > 0 \\ \text{TF-IDF}_{ij} & \text{if } i \text{ is document, j is word} \\ 1 & \text{if } \exists \text{ a citation relationship} \\ & \text{between documents } i \text{ and } j \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where PMI denotes the point-wise mutual information between words i and j counted within local word windows, similar to [10]. TF-IDF$_{ij}$ represents the tf-idf score of word j within document i.

Node Interaction Pattern Computation. The computation of node interaction patterns (NIP) proceeds as shown in the NIP Computation block of Figure 2. We first execute node2vec [5] on a pruned graph that models only the containment relationships of words in documents and the inter-document citation links (in our initial experiments, we found that including word-word edges would make the graph too dense and difficult to learn meaningful node representations). For each document node in this pruned graph, we then compute local interactions as

$$\mathcal{IS}_i = \{\boldsymbol{\mathcal{NV}}_i \otimes \boldsymbol{\mathcal{NV}}_k, \forall k \in \mathcal{N}(i)\}, \tag{2}$$

where $\mathcal{N}(i)$ denotes the set of the neighbouring nodes of node i, $\otimes$ denotes the interaction operator (specifically, the cosine similarity), $\boldsymbol{\mathcal{NV}} \in \mathbb{R}^{|\mathcal{V}| \times n_0}$ denotes the node2vec embeddings of the nodes in the graph such that $\boldsymbol{\mathcal{NV}}_i$ is the node2vec embedding of node i (n_0 being the dimension of the embedding).

It should be noted that the size of the set $\mathcal{IS}_i$ is not fixed due to the variations in the degrees of the nodes. We next partition local interactions ($\mathcal{IS}_i$) according to different levels of interaction weights, similar to [6]. Specifically, since interaction weight values are within the interval $[-1, 1]$, we discretize the interval into a set of ordered bins and accumulate the count of local interactions in each bin. We investigate the following two ways of histogram mapping, as prescribed in [6] -

(1) **Normalized Histogram (NH):** An L_2 normalization is applied on the frequencies of each bin to focus on the relative rather than the absolute number of interactions.
(2) **Log Count-based Histogram (LCH):** We apply logarithm over the frequencies mainly to reduce their variations. This is reported to provide better results in [2].

Next, let $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{|\mathcal{V}| \times b}$ denote the interaction vectors (the mapped histograms) of the nodes in the graph such that $\boldsymbol{\mathcal{I}}_i$ is the interaction vector of node i, where b is the number of bins for histogram computation. Since we conduct the interaction operations only for document nodes, the interaction vectors for the word nodes are initialized to zeros.

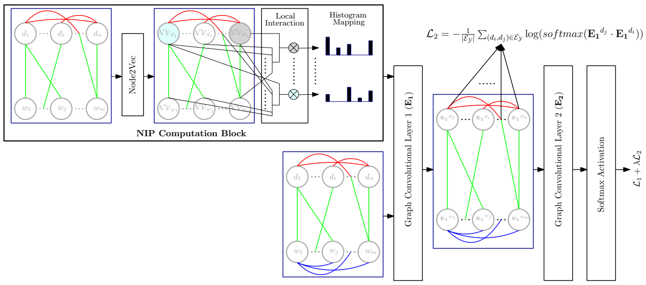


Figure 2: A schematic representation of NIP-GCN. The green edges denote the containment relation of words in a document, the red edges denote the document citation relationships and the blue edges denote the word co-occurrence relationships.

Two-Layer GCN with Additional Noise-Contrastive Loss. In this work, we use two graph convolutional layers followed by softmax activation on each node (as shown in Figure 2), i.e.,

$$\mathbf{Z} = \text{softmax}(\hat{\mathbf{A}} \max(0, \hat{\mathbf{A}}\boldsymbol{\mathcal{X}}\mathbf{W_0})\mathbf{W_1}) \tag{3}$$

where, $\mathbf{W_0}$ and $\mathbf{W_1}$ are the updatable parameter matrices for layers 1 and 2 respectively. The way we set $\boldsymbol{\mathcal{X}}$ is as follows: Let $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ denote the matrix of one-hot encodings of the nodes in the graph such that $\boldsymbol{\mathcal{O}}_i$ is the one-hot vector of node i. We then set $\boldsymbol{\mathcal{X}}$ to be the concatenation of $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{I}}$, i.e., $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|+b}$.

We then optimize the cross-entropy loss over the node labels corresponding to the documents with known classes, i.e.,

$$\mathcal{L}_1 = -\sum_{d\in\mathcal{Y}}\sum_{f=1}^{F} \mathbf{Y_{df}} \ln \mathbf{Z_{df}} \tag{4}$$

where $\mathbf{Y}$ is the label indicator matrix, $\mathcal{Y}$ is the set of document indices that are associated with known labels (document classes), and F is the dimension of the output features being identical to the number of classes. In Equation 3, $\mathbf{E_1} = \max(0, \hat{\mathbf{A}}\boldsymbol{\mathcal{X}}\mathbf{W_0})$ denotes the intermediate outputs of the first GCN layer with ReLU activation, whereas $\mathbf{E_2} = \hat{\mathbf{A}} \max(0, \hat{\mathbf{A}}\boldsymbol{\mathcal{X}}\mathbf{W_0})\mathbf{W_1}$ denotes the second GCN layer embeddings.

Pairwise Document Similarities. In addition to the cross entropy loss of a standard GCN, we incorporate an additional loss component which is a function of the first layer document embeddings (as shown in Figure 2). This additional component (defined later in Equation 7) seeks to make the vector representations of two linked documents similar to each other. The motivation behind this step is that it is less likely that a document of one class would have a link to a document of another class, e.g., a sports article will likely have links to other sports articles only and so on. Formally,

$$\mathcal{L}_2' = -\frac{1}{|\mathcal{E}_\mathcal{Y}|}\sum_{(d_i,d_j)\in\mathcal{E}_\mathcal{Y}} \log(\text{softmax}(\mathbf{E_1}^{d_j} \cdot \mathbf{E_1}^{d_i})) \tag{5}$$

where, $\mathcal{E}_\mathcal{Y}$ is the set of available information regarding links across training set document pairs, and $\mathbf{E_1}^d$ denotes the row of $\mathbf{E_1}$ corresponding to the document node d. The negative sign at the front transforms the likelihood maximization problem into a loss minimization one. The intractable softmax calculation of Equation 5 is approximated with noise-contrastive estimation with negative sampling [15], i.e.,

Dataset	#Docs	#Links	#Classes	Avg. Len
M10	10,310	77,218	10	7.3
DBLP	60,744	52,890	4	7.5
Covid-Full (title+abstract)	13,678	179,909	15	116.3
Covid-Title (titles only)				10.8

Table 1: Dataset Characteristics

$$\log(\text{softmax}(\mathbf{E_1}^{d_j} \cdot \mathbf{E_1}^{d_i})) \approx \\ \log\sigma(\mathbf{E_1}^{d_j} \cdot \mathbf{E_1}^{d_i}) - \sum_{t=1}^{K} \mathbb{E}_{d_t\sim\mathcal{NG}(i,j)} \log\sigma(\mathbf{E_1}^{d_t} \cdot \mathbf{E_1}^{d_i}) \tag{6}$$

where d_t in Equation (6) is a randomly sampled document from the set of documents that are not linked with d_i and have different labels than that of d_i. This set of negative samples is denoted as $\mathcal{NG}(i,j)$. For each linked document pair, the number of random negative sample used is K (a hyper-parameter). Denoting $\mathcal{NG}$ as a multiset, $\mathcal{NG} = \bigcup_{i,j:(d_i,d_j)\in\mathcal{E}_\mathcal{Y}} \mathcal{NG}(i,j)$, and combining Equations (5) and (6), the overall loss is expressed as

$$\mathcal{L}_2 = \sum_{(d_i,d_j)\in\mathcal{E}_\mathcal{Y}} \frac{-\log\sigma(\mathbf{E_1}^{d_j} \cdot \mathbf{E_1}^{d_i})}{|\mathcal{E}_\mathcal{Y}|} + \sum_{(d_i,d_t)\in\mathcal{NG}} \frac{\log\sigma(\mathbf{E_1}^{d_t} \cdot \mathbf{E_1}^{d_i})}{K|\mathcal{E}_\mathcal{Y}|} \tag{7}$$

In our experiments, we use a combination of the cross-entropy loss of Equation 4 with a regularized version of Equation 7, i.e.,

$$\mathcal{L} = \mathcal{L}_1 + \lambda\mathcal{L}_2, \tag{8}$$

where $0 < \lambda \leq 1$ is a regularization parameter, the optimal values of which, for the different datasets in our experiments, were found to be within a range of 0.001 to 0.004.

4 EXPERIMENTS

Datasets. We conducted experiments on three linked collections, namely DBLP, CiteSeer-M10, and a newly crawled document collection on Covid-related research articles. All the three document collections are citation networks with explicit links across documents. We used the title field of each document of DBLP [18] and CiteSeer-M10 (denoted in this paper as M10) [11] to represent its content. For the Covid dataset, we used two versions of the collection - one without the abstracts (i.e. title only) and the other with both.

The Covid dataset[1] was created from weekly searches of PubMed and Embase between March and October 2020. Records were retrieved and then manually checked for eligibility for inclusion in a database of research on COVID-19. After checking for eligibility, records were then manually assigned to one of 15 classes, including 'case reports', 'vaccine development', 'health impacts' etc. More details about the dataset could be found in [12].

[1] https://github.com/ManishChandra12/NIP-GCN/tree/master/data

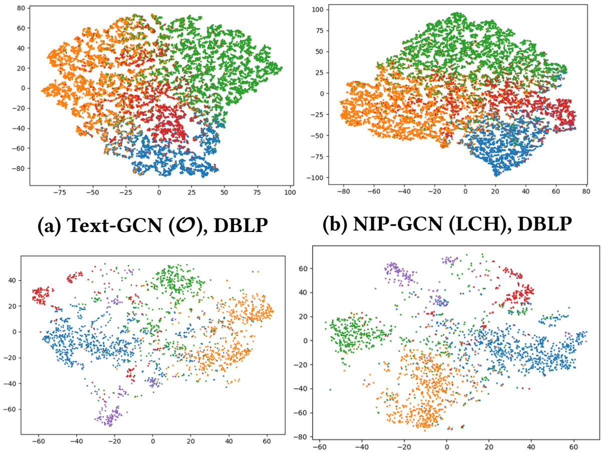
Table 2: Macro-averaged F1 (%) on the test splits of the datasets.

Model	$\mathcal{L}2$	$\mathcal{X}$	M10	DBLP	Covid-Title	Covid-Full
GCN	N	BoW	71.61	74.01	37.05	34.51
Text-GCN	N	$\mathcal{O}$	74.23	74.42	54.76	60.48
Text-GCN	N	$\mathcal{NV}$	72.87	72.42	55.23	53.86
NIP-GCN	N	$\mathcal{O}+\mathcal{I}$	75.13	74.90	57.93	62.73
(LCH)	Y	$\mathcal{O}+\mathcal{I}$	**75.77**	75.47	**58.19**	63.22
NIP-GCN	N	$\mathcal{O}+\mathcal{I}$	75.03	75.39	57.45	63.24
(NH)	Y	$\mathcal{O}+\mathcal{I}$	75.65	**75.51**	58.01	**63.76**

Baselines. We compare our proposed method, NIP-GCN[2], with the following baselines. First, we employ **GCN** - A graph CNN model using the first-order approximation of GCNs with Chebyshev polynomials [9]. The graph models only the document citation relationships and $\mathcal{X}$ corresponds to the bag-of-words representation of documents. Second, in **Text-GCN** an entire corpus is modelled as a heterogeneous graph with nodes corresponding to a document or a word [22]. For linked document classification, we additionally represent links between documents as edges. In addition to the one-hot encodings of Text-GCN, we also employ another baseline by setting $\mathcal{X}$ to the normalized node2vec representations of the nodes ($\mathcal{NV}$) as a straight-forward application of node2vec embeddings within a GCN (without the interaction patterns). These two baselines correspond to the 2^{nd} and the 3^{rd} rows of Table 2.

Hyper-parameter settings. We used a train:test split of 70:30 with 10% of the training set for validation purposes. We used the validation set for optimizing each hyper-parameter, namely the dropout rate across both layers, embedding size of the first convolution layer, learning rate, λ and the number of bins. The hyper-parameters were tuned on the Covid-Title dataset and then subsequently used for the experiments on the othe datasets. We noted that the hyper-parameter λ did not generalize well across the datasets and hence was tuned individually on each.

We set the embedding size of the first convolution layer to 200, the sliding window size for PMI calculation to 20, the learning rate to 0.02 and the dropout rate to 0.5. We set the number of negative samples (K) to 7 for every linked document pair. We trained our model for a maximum of 200 epochs using the Adam optimizer. We set the number of bins (b) to 10. To ensure fair comparisons, we used an identical setup for hyper-parameter optimization for each baseline model. For GCN, embedding size of the first convolution layer was set to 64, learning rate to 0.01, dropout rate to 0.5 and L_2 regularization to 10^{-3}. For Text-GCN, we set embedding size of the first convolution layer to 200, the sliding window size for PMI calculation to 20, the learning rate to 0.02 and the dropout rate to 0.5. The optimal learning rate of the Text-GCN initialized with node2vec embeddings was found to be 0.05.



(a) Text-GCN ($\mathcal{O}$), DBLP

(b) NIP-GCN (LCH), DBLP

(c) Text-GCN ($\mathcal{O}$), Covid-Full

(d) NIP-GCN (LCH), Covid-Full

Figure 3: 2D t-SNE visualization of the embedded vectors of each document node in the test split (each ground-truth class plotted with a different color). Only 5 classes are shown for Covid-Full dataset to avoid clutter.

Results. The macro-averaged F1 score values obtained by our model on different datasets are shown in Table 2. For all the experiments involving node2vec, the reported numbers are the average of 5 different runs of it. First, we observe that initializing Text-GCN with node2vec embeddings does not yield effective results (3^{rd} row of Table 2), which shows that simply feeding in a pre-trained representation of the nodes is not helpful for a GCN. However, the use of interaction patterns (NIP prefixed approaches in Table 2) are shown to yield effective results. Indeed, we observe the following - i) LCH-based interactions outperform their NH-based counterparts for smaller datasets (M10 and Covid-Title) while it is the other way around for larger datasets (DBLP and Covid-Full), and ii) the proposed noise-contrastive loss (Equation 8) improves the performance across all the datasets.

For comparison purposes, Figure 3 illustrates the 2D visualizations (t-SNE) of the document embeddings learned by NIP-GCN and Text-GCN. Figure 3 demonstrates that the document embeddings learned with NIP-GCN are more discriminative. Specifically, in the DBLP dataset, the documents belonging to the same topic forms more compact clusters (as perceived visually) with NIP-GCN (compare the orange and the blue colored points of NIP-GCN vs. the Text-GCN). Similarly for Covid-Full, the cluster formed by the documents with the orange colored topic is more compact in the case of NIP-GCN than with Text-GCN.

5 CONCLUSION AND FUTURE WORK

In this study, we incorporated the node interaction patterns induced from the linkage structure of a graph into the GCN framework. This provides the graph convolutional layers with prior information which can be harnessed to learn more discriminative node embeddings. Combining this prior information with an additional pairwise document similarity objective is demonstrated to yield superior results to the Text-GCN (with document-document edges included). In future, we would like to investigate if interaction-focused models could be used to effectively model the relations between other types of entities such as authorship, latent topics of documents etc.

[2] The code for NIP-GCN is available for research purposes at https://github.com/ManishChandra12/NIP-GCN.

REFERENCES

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

[2] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning* (Bonn, Germany) *(ICML '05)*. Association for Computing Machinery, New York, NY, USA, 89–96.

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 3844–3852.

[4] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) *(ICML'17)*. JMLR.org, 1263–1272.

[5] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 855–864.

[6] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-Hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 55–64.

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[8] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc.

[9] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations* (Palais des Congrès Neptune, Toulon, France) *(ICLR '17)*.

[10] Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'14)*. MIT Press, Cambridge, MA, USA, 2177–2185.

[11] Kar Wai Lim and Wray Buntine. 2015. Bibliographic Analysis with the Citation Network Topic Model *(Proceedings of Machine Learning Research, Vol. 39)*, Dinh Phung and Hang Li (Eds.). PMLR, Nha Trang City, Vietnam, 142–158.

[12] T Lorenc, C Khouja, G Raine, I Shemilt, K Sutcliffe, P D'Souza, H Burchett, K Hinds, W Macdowall, H Melton, M Richardson, E South, C Stansfield, S Thomas, I Kwan, K Wright, A Sowden, and J Thomas. 2020. COVID-19: living map of the evidence. http://eppi.ioe.ac.uk/cms/Projects/DepartmentofHealthandSocialCare/Publishedreviews/COVID-19Livingsystematicmapoftheevidence/tabid/3765/Default.aspx. Accessed: 2021-05-11.

[13] Zhengdong Lu and Hang Li. 2013. A Deep Architecture for Matching Short Texts. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 26. Curran Associates, Inc.

[14] Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1506–1515.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[16] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 2793–2799.

[17] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Proceedings (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics))*. Springer/Verlag, 593–607. 15th International Conference on Extended Semantic Web Conference, ESWC 2018 ; Conference date: 03-06-2018 Through 07-06-2018.

[18] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Las Vegas, Nevada, USA) *(KDD '08)*. ACM, New York, NY, USA, 990–998.

[19] Suhang Wang, Jiliang Tang, Charu Aggarwal, and Huan Liu. 2016. Linked Document Embedding for Classification. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 115–124.

[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. 2020. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* (2020), 1–21.

[21] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-Hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. Association for Computing Machinery, 55–64.

[22] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification *(33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019)*. AAAI Press, 7370–7377.