Graph Convolutional Network-Strengthened Topic Modeling for Scientific Papers

Jia Zhang[1],Junhao Shen[1],Beichen Hu[1],Rahul Ramachandran[2],Tsengdar J. Lee[3],Kwo-Sen Kuo[4],Manil Maskey[2],Seungwon Lee[5]
[1]*Department of Computer Science, Southern Methodist University, USA*
[2]*NASA/MSFC, USA*
[3]*Science Mission Directorate, NASA Headquarters, USA*
[4]*NASA/GSFC, USA*
[5]*NASA/JPL, USA*
{jiazhang,junhaos,beichenh}@smu.edu;{rahul.ramachandran,tsengdar.j.lee,kwo-sen.kuo}@nasa.gov;seungwon.lee@jpl.nasa.gov

***Abstract*—Machine learning has been woven into statistics to modernize topic modeling over textual documents written in natural language, and scientific paper search and recommendation can consequently offer higher accuracy instead of counting on traditional keyword-based search. However, topic distribution of a paper resulted from existing topic modeling techniques only relies on the statistics of words contained in the paper itself. We argue that community users' views of a paper may also provide insights at the time of recommendation. For example, if a paper on fake image detection has been cited heavily by machine learning papers, such a feature should be absorbed in the embedding of this paper, so that it can be recommended for future query on machine learning. In this paper, we present a Graph Convolutional Network-strengthened Topic Modeling (GCN-TM) method, which employs GCN technique to refine topic modeling of scientific papers. A citation-oriented knowledge graph is constructed, and topic modeling is mapped to feature embedding of the comprising papers. On top of its own topics carried in its content, each paper learns topics from its neighbors and revise its embedding accordingly. Our empirical studies over real-life scientific literature has proved the necessity and effectiveness of our proposed approach.**

***Index Terms*—Science knowledge graph, topic modeling, graph convolutional network**

I. Introduction

Paper recommendation remains an extremely important service that researchers highly demand in modern society. One critical criterion for high-quality papers is to rigorously compare with related work in the literature. In the knowledge explosion era nowadays with numerous publications in a variety of channels, it has become increasingly necessary to inform researchers related work that they shall be aware of and compare with. Toward this goal, recent years have witnessed a collection of online portals dedicated for paper search and recommendation purposes, such as DBLP[1], Google Scholar[2], Microsoft Academic Knowledge Services[3], AMiner[4], ResearchGate[5], as well as portals hosted by paper publishers like IEEE Xplore[6], ACM Digital Library[7], and Elsevier[8].

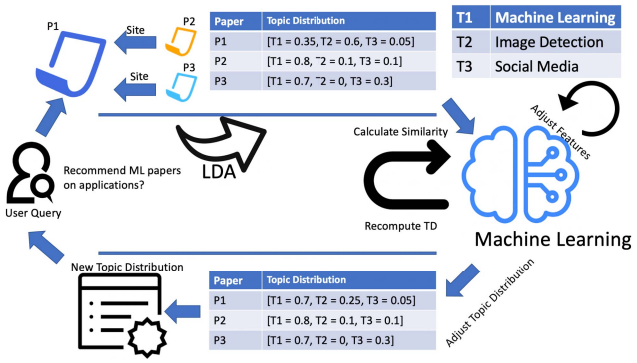

Fig. 1. **Motivating Example.**

These portals all provide rich keyword-based search function, ranging from simple keyword search to autofill support. The advancements in natural language processing (NLP) and machine learning have enabled more advanced paper search based on topic modeling. Topic modeling in NLP is a widely used technique capable of clustering textual documents. Topic modeling algorithms, represented by Latent Dirichlet Allocation (LDA) [1], exploit statistics and machine learning to assign a distribution of topics to each document, and a distribution of words to each topic to provide probabilistic descriptions of textual documents. However, existing topic modeling algorithms merely rely on the statistics of words in each paper alone, and readers' opinions are not taken into consideration. In reality, such methods bear some limitations. For example, the selling points of a paper by its authors may not always be the reason why community members cite the paper.

Fig. 1 shows one example scenario that directly motivates this research. Assume a paper p_1 claims that its focus is on

[1]https://dblp.uni-trier.de/
[2]https://scholar.google.com/
[3]https://www.microsoft.com/en-us/research/academic-program/academic-services/
[4]https://www.aminer.org/
[5]https://www.researchgate.net/
[6]https://ieeexplore.ieee.org/
[7]https://dl.acm.org/
[8]https://www.elsevier.com/

978-1-6654-0058-9/21/$31.00 ©2021 IEEE
DOI 10.1109/SMDS53860.2021.00014

fake image detection algorithm and its applications on social media, and their algorithm is based on an extension of a convolutional neural network (CNN) algorithm. Following the traditional topic modeling techniques, such as LDA, the topic distribution of the paper may be 60% of image detection (T2), 5% of social media, and 35% on machine learning. As a result, the paper may not be recommended to a search query on "recommend ML papers on applications." However, if a number of machine learning papers (e.g., p_2, p_3, etc) cite paper p_1 on its novel CNN-based method, then this paper p_1 might need to be recommended to machine learning readers. This example shows that such readers' opinions or views should be properly recorded and taken into account, so as to enhance paper recommendation. Such scenarios have also been observed in general object recommendation. As social science reveals, people usually watch and consider their peers' behaviors [2]. As an analogous example, Amazon considers recorded other users' decisions when recommending items.

In recent years, researchers have started to leverage representation learning to gather feature representations of nodes in complex networks, and map them to low-dimension embedding space for various types of tasks such as classification and link prediction. Based on graph neural networks, a collection of deep encoders have emerged to encode a node into node embeddings through multiple layers of graph convolutions, non-linear transformations of network structures, and regularization (e.g., dropout). Throughout this paper, we will use the following terms interchangeably: node embedding, feature vector of node, node representation, node signals, and topic distribution.

In this work, we propose a Graph Convolutional Network-strengthened Topic Modeling (GCN-TM) method. We first construct a directed knowledge graph based on citation relationships in the literature. All nodes represent all the papers in the corpus. Each edge represents a citation relationship between two papers. If an edge points to paper p_1 from paper p_2 ($p_1 \leftarrow p_2$), it means paper p_2 cites paper p_1. As an analogy, we can view paper p_2 is a user of paper p_1, also as its neighbor in the knowledge graph.

Each paper is mapped to the embedding space, and each search query is also mapped to the same embedding space. We first run a regular topic modeling algorithm, e.g., vanilla LDA, to learn a topic distribution for each paper contained in the knowledge graph. Such a topic distribution is treated as the individual features of each paper, and create its initial embedding. In other words, the LDA-resulted topic distribution is used as the initial feature vector for each paper. Afterwards, each paper will aggregate features (i.e., topic distribution) from all of its incoming neighbors (i.e., papers that cite the paper), and in turn their incoming neighbors, and so on. The final embedding of each paper will be the revised feature vector for the paper, which will better represent its citation relationships with other papers. To our best knowledge, our work is the first effort that applies graph convolutional network techniques to incorporate readers' opinions to improve paper recommendation.

The major contributions of this paper are two-fold.

1) We present a GCN-TM method, which employs Graph Convolutional Networks to learn from peer researchers' views to adjust the feature embeddings of scientific papers, in order to enhance paper recommendation.
2) Our experiments over real-world dataset have proved the necessity and effectiveness of our approach.

The remainder of this paper is organized as follows. In Section 2, we discuss related work. In Sections III and IV, we introduce preliminaries, followed by details of our GCN-TM technique. In Section V, we discuss experimental studies. Finally in Section VI, we draw conclusions.

II. Related Work

Our work is closely related to two categories of research in the literature: topic modeling and graph neural network.

A. Topic Modeling

A scientific paper typically touches more than one topic, thus traditional classification methods do not apply well. That is why topic modeling has been widely adopted to learn fine-grained topic distributions of scientific papers. Based on statistics and machine learning, Latent Dirichlet Allocation (LDA) [1] and its variants have been considered as representative techniques for topic modeling. This paper applies graph neural network technique on top of LDA to leverage their citation relationships to better support paper recommendation.

This paper is also inspired by our earlier finding [9] that user opinions should be taken into consideration to enrich software service profiles. In our earlier work [9], we developed a machine learning model called Service Representation-Latent Dirichlet Allocation (SR-LDA), which extends LDA to enrich service profiles from their involved mashup profiles. The core hypothesis is that each mashup profile is co-authored by all of its invoked services. That technique, however, cannot be applied to paper topic enrichment. The reason is that a scientific paper typically cites a collection of papers for comparison purpose, and such reference papers do not contribute to the core claims of the paper.

B. Graph Neural Network

Graph neural networks (GNNs) have gained significant momentum since its inception in 2017, as many real-world data and relationships can be represented by graphs. Perozzi et al. [6] present the *DeepWalk* algorithm that learns structural representations of nodes, through random walks. Grover and Leskovec [3] propose *node2vec* to further consider the possibility of favoring Depth-First Search (DFS) and Breadth-First Search (BFS) when doing random walking. However, they both are transductive.

Graph convolution networks (GCNs) [5] allow graph nodes to borrow features from their neighbors, and neighbors of the neighbors, and combines with the features of the nodes to learn more enriched embeddings. GraphSAGE [4] changes the node representation propagation rules in two ways: one is to

introduce a more general node aggregation operator, and the other is to separate the features of a node and those from its neighbors. In our neural network design, we combine the computing rules of GCN and GraphSAGE. Due to the fact that reference papers usually share the same roles, we adopted a mean average aggregator [5] to accumulate features from related papers, instead of commonly used operator such as max pooling, min pooling, or neural graph collaborative filtering operator [8].

III. Preliminaries

In this section, we will introduce some preliminaries of this work and define our target problem.

A. Topic Modeling

Topic modeling in NLP is a widely used technique capable of clustering textual documents. Topic modeling algorithms, represented by Latent Dirichlet Allocation (LDA) [1], exploit statistics and machine learning to assign a distribution of topics to each document, and a distribution of words to each topic to provide probabilistic descriptions of textual documents.

B. Notations and Problem Definitions

Definition 1 (Citation Knowledge Graph)**.** A citation knowledge graph is a directed 4-tuple graph $\mathbb{CKG} = G(P, E, A, T)$:

1) P denotes a collection of nodes where each node represents a paper.
2) E denotes a collection of edges between nodes. Each directed edge represents a citation relationship between two paper nodes: $\forall e(p_i, p_j) \in E, p_i, p_j \in P, p_i \leftarrow p_j$, it means paper p_j cites paper p_i.
3) A denotes an adjacency matrix of the graph. If $\exists e(p_i, p_j) \in E \Rightarrow a_{i,j} = 1$; otherwise $a_{i,j} = 0$.
4) T denotes a matrix of node features, represented by topic distributions: $T \in \mathbb{R}^{N \times |P|}$, where N represents a predefined number of topics over the paper corpus underneath the graph.

Definition 2 (Embedding Space)**.** The embedding space of a citation knowledge graph (CKG) is an N-dimension space $\mathbb{R}^{\mathbb{N}}$, where N is a predefined number of topics for the specific domain to which all papers in the $\mathbb{CKG}$ belong.

Each paper is represented as an N-dimension vector embedding. For example, if 20 topics are set to categorize all papers in a CKG, the embedding space will be 20 dimensions.

Table I summarizes all notations that we will use throughout the paper.

Problem Formulation. The paper recommendation problem can be defined as: given observed interaction records in a citation knowledge graph $\mathbb{CKG}$, for a search query q, we aim to find a collection of ranked papers based on their similarity to the search query in the embedding space.

TABLE I
Notations and Explanations

Notation	Explanation
$\mathbb{CKG}$	Citation Knowledge Graph
$\mathbb{CKG}'$	Adjusted Citation Knowledge Graph
$\mathbb{CG}$	Computation Graph
P	Paper set
E	Paper citation relationship set
T	Paper feature matrix
$\mathbb{NN}^L$	Neural network at layer L of $\mathbb{CG}$
$\mathbf{t}_p^k$	Embedding of paper p at level k
$C(p)$	All papers cite paper p
Ω	Transformation matrix
Φ	Transformation matrix

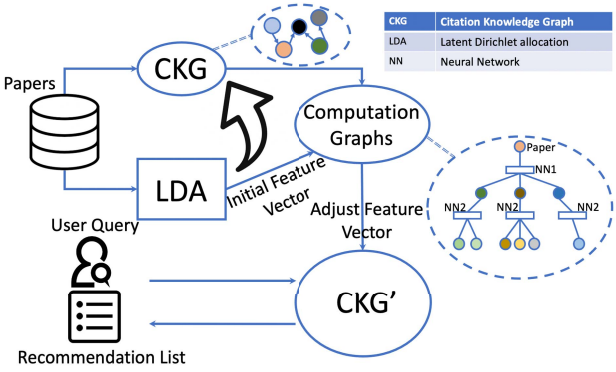

Fig. 2. **Blueprint of GCN-TM framework.**

IV. GCN-TM

In this section, we will introduce our Graph Convolutional Network-strengthened Topic Modeling (GCN-TM) technique.

Fig. 2 illustrates a high-level overview of our proposed GCN-TM framework. As shown in Fig. 2, the input of the framework is a corpus of papers on the left-hand side. The papers will be parsed, and their citation relationships will be extracted to construct a $\mathbb{CKG}$ network. For each of its comprising nodes, a computation graph is constructed, which is rooted by the node and contains its hierarchical neighbors in the $\mathbb{CKG}$ network. Each layer of each computation graph consists of a neural network, e.g., NN1 and NN2. The neural network at each layer is shared by all the nodes at the same layer in all computation graphs. As shown in Fig. 2, the papers are sent to train an LDA model, resulted in a topic distribution for each paper, which serves as the initial embedding (i.e., feature vector) of each paper in the computation graphs. For each computation graph, the root paper will run the neural networks to aggregate the topic features from its direct neighbor nodes (papers that cite the paper) into its own feature vector, which in turn from their own neighbor nodes. This simultaneous process will result in a $\mathbb{CKG}'$ network, while each of its comprised paper nodes carries an adjusted feature embedding. Such a yielded $\mathbb{CKG}'$ will accept a user query and return a recommended paper list sorted.

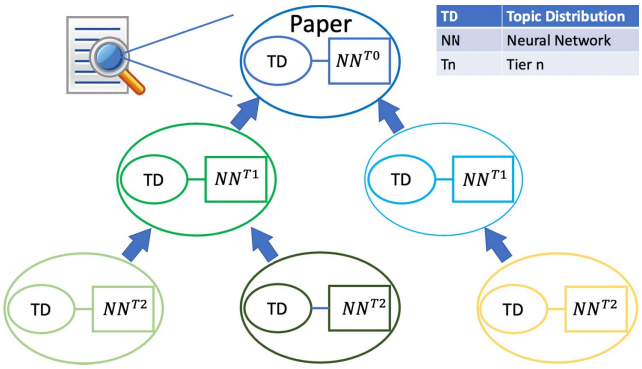

Fig. 3. **Computation Graph.**

A. Computation Graph Construction

For each paper, we build a dedicated neural network architecture based on its direct and indirect citation relationships. For each node (paper) in the $\mathbb{CKG}$, we first define a computation graph for it, based on its neighborhood structure in the network, as shown in Fig. 3. Each node in a computation graph encapsulates two components: a topic distribution (TD) and a neural network (NN). Analogous to an object in Object-Oriented (O-O) design, the TD component represents its data status, and the NN component can be viewed as a function carried to update its data. As shown in Fig. 3, the NN component in each node will absorb and aggregate the TD signals from its children nodes, and in turn from their children nodes, into its own TD features to compute an adjusted representation to update its TD status, i.e., its feature vector. Note that such a feature aggregation process will simultaneously learn to capture the structural information around the paper, as well as how to absorb signals from the papers that cite it to adjust its representation.

Definition 3 (Computation Graph)**.** The computation graph for a node $p \in P$ in a $\mathbb{CKG}$ network is a tree-like graph rooted by node p, $CG(p) = G(P', E', A', T', p, \mathbb{NN}^{\mathbb{L}}), L = 0, 1, 2, ...$

1) P' denotes a subset of the paper nodes in $\mathbb{CKG}$: $P' \subseteq P$.
2) E' denotes a subset of the paper citation edges in $\mathbb{CKG}$: $E' \subseteq E$.
3) A' denotes an adjacency matrix of the graph. If $\exists e(p_i, p_j) \in E' \Rightarrow a'_{i,j} = 1$; otherwise $a'_{i,j} = 0$.
4) T' denotes a matrix of node features, represented by topic distributions: $T' \in \mathbb{R}^{N \times |P'|}$, for all nodes in P'.
5) $\mathbb{NN}^L$ indicates a set of neural networks, each at one layer of $CG(p)$.
6) The length of each edge is 1. The layer of a node p' is counted as its distance to the root: $L(p') = path(p, p')$.

Each layer of $CG(p)$ has one neural network: $\mathbb{NN}^0$ for layer 0 at the root level, $\mathbb{NN}^1$ for layer 1, and so on. All nodes at the same layer L of $CG(p)$ share the same neural network at the layer $\mathbb{NN}^L$. A neural network helps a node to aggregate features from its children nodes, and incorporates them into its own features to result in an adjusted representation for the node.

Note that a computation graph is a connected acyclic directed graph. If a node is a leaf node, then its associated neural network has no children nodes to aggregate features from. In other words, each leaf node carries its own features only.

For every node in $\mathbb{CKG}$, it has its own computation graph as an neural network architecture. Note that all computation graphs share the same set of layer-oriented neural networks.

B. Neural Network Design

The core element of the neural network component $\mathbb{NN}$ is its graph convolutional operator across layers, which is defined as follows:

$$\mathbf{t}_p^k = ReLU(\mathbf{\Omega}^k \sum_{v \in C(p)} \frac{\mathbf{t}_v^{k+1}}{|C(p)|}, \mathbf{\Phi}^k \mathbf{t}_p^{k+1}) \quad (1)$$

where $\mathbf{t}_p^k$ denotes the representation of paper p at level k, $C(p)$ denotes all children nodes of node p, $ReLU$ is a common nonlinear activation function, Ω and Φ are transformation matrices.

For each inner layer k of $\mathbb{CG}$ with a total of layers K $(0 \leqslant k < K)$, the graph convolutional operator aggregates the incoming messages from all children nodes of the node (first portion), concatenates with its own topic distribution information (second portion), and then applies a non-linear transformation operation ($ReLU$). If a node p is at a leaf layer ($k = K$), its representation is its topic distribution TD: T'_p: $\mathbf{t}_p^K = T'_p$. The representation of the root layer 0 will be the final adjusted embedding of the node: $\mathbf{t}_p = \mathbf{t}_p^0$.

The design of our graph convolutional operator combines the computing rules from GCN [5] and GraphSAGE [4]. On the one hand, we adopt the mean average aggregator from the former when aggregating messages from neighbors, because we favor the quantitative topic impact from the audience of the paper (i.e., papers that cite the paper). On the other hand, we adopt the concatenation operator from the latter between aggregated messages and message from the paper itself, for finer-grained training. Note that the aggregation operator has to be order-invariant, and we choose the average operator due to the unique feature of papers. We hope to absorb topic information from neighbor papers to adjust its own embedding for recommendation purpose. In the last two years, researchers have proposed various aggregation operators. For example, Wang et al. [8] propose a neural graph collaborative filtering operator for neighborhood aggregation. It considers the degrees of both message sender and recipient, which is useful in the field of social network analysis while message impact may be decayed by a high popularity of either sender or receiver. Regarding paper citation though, if a paper is cited by many papers with the same view, this view should be confirmed thus strengthened instead of weakened. For similar reasons, we did not select other commonly used aggregation operators in GCNs such as max pooling or min pooling.

To carry out the propagation process for all papers in the CG, we transform the equation above into a matrix form as below. Note that the neural network (i.e., Ω and Φ) are shared by all nodes at the same layer of all computation graphs.

$$T^k = ReLU(\mathcal{L}\mathbf{\Omega}^k \Delta T^{(k+1)} + \mathbf{\Phi}^k T^{k+1}) \quad (2)$$

where T^k is the new embedding matrix after the last propagation, and $\mathcal{L}$ is the Laplacian matrix of $\mathbb{CG}$. Recall that $\mathbf{A}'$ and $\mathbf{D}'$ are adjacency matrix and diagonal degree matrix of $\mathbb{CG}$, respectively. The Laplacian matrix $\mathcal{L}$ can be calculated as follows:

$$\mathcal{L} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}. \quad (3)$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ denotes the adjacency matrix of $\mathbb{CG}$ with added self-connections, $\mathbf{I}$ is its identity matrix, and $\tilde{\mathbf{D}}_{ii} = \Sigma_j \tilde{\mathbf{A}}_{ij}$.

The matrix form of the learning process helps not only update all paper representations in the same $\mathbb{CG}$ simultaneously, but also facilitate the batch calculation.

C. Parameter Learning

In this section, we will analyze the parameter training aspects regarding the optimization and training efficiency of our GCN-TM model.

1) Loss Function: We adopt the hinge loss to construct our loss function, so that we can feed the neural network embeddings and run stochastic gradient descent to train the parameter matrices (Ω and Φ) at each layer. Hinge loss, also called max-margin objective function, is considered more robust and powerful than the regular cross entropy error and softmax function. The hinge function is defined as follows:

$$Ł = \sum_{v \in CG(p), u \notin CG(p)} \max(0, -\mathbf{t}_p^T \mathbf{t}_v + \mathbf{t}_p^T \mathbf{t}_u + \delta) \quad (4)$$

where δ denotes the margin, meaning the extent of larger positive paper pair similarity should be compared with negative paper pairs.

Note that here we calculate the cosine similarity of a pair of paper embeddings. The embeddings of papers similar to each other should be close to each other; and the embeddings of papers different from each other should be far apart. The physical meaning is that, as shown in Equation (4), pairs in a computation graph (i.e., neighbors) should be closer than those otherwise.

2) Unsupervised Learning: We employ unsupervised training. Intuitively, a papers is similar to its reference paper in the embedding space, and different from a paper not in the computation graph of the paper. In addition, papers cited by the same paper could be reviewed as fall in similar topics. In other words, the embeddings of two papers in the same computation graph may be closer than otherwise. In yet another words, a paper and another paper in the same computation graph can be treated as positive pairs; the paper and one paper randomly selected not from the same computation graph can be viewed as a negative pair. In this way, we can obtain thousands of training data.

Upon the training data, we train a graph neural network to generate embeddings for papers, so that the embeddings of positive pairs of papers are closer to each other comparing to those of negative pairs. After the GNN model is trained, we will create embeddings for all papers dynamically based on their computation graphs. Given a search query, we can do a nearest neighbor search in the embedding space and recommend similar papers.

D. Considerations

1) Neural Network Depth: As described in the previous sections, each paper is associated with a neural network. According to the small-world theory, the diameter of a network is six. Therefore, empirical studies typically suggest that the unfolding process goes to merely three or four steps, otherwise a large number of weakly-connected papers will be involved which are unnecessary. For scientific paper citation, be more specific, two papers apart more than two hops may not be very similar. One reason is that, authors who cite a paper typically also are aware of its reference papers. Thus, if necessary, they should have cited those reference papers as well. For example, if paper p_2 cites paper p_1 and paper p_3 cites paper p_2, it is reasonable to assume that the authors of p_3 have read or at least browsed through all reference papers of p_2, thus are aware of the existence of paper p_1. Thus in general, if paper p_3 does not cite paper p_1, it is reasonable to assume that the authors of paper p_3 do not think paper p_1 very similar to theirs. Therefore, when we build a neural network for a paper, we decide to only unroll two levels.

2) Acyclicity: A citation knowledge graph bears an acyclicity feature. The reason is simple, as paper publications have an inherent chronologicity. For example, if paper p_2 cites paper p_1 and paper p_3 cites paper p_2 ($p_3 \rightarrow p_2 \rightarrow p_1$), then paper p_2 is published later than paper p_1 and paper p_3 later than paper p_2. Thus, it is impossible for the earlier published paper p_1 to cite a later paper p_3.

3) Inductivity: Note that the trained neural networks are shared by the same layers of all computation graphs, i.e., they are inductive. For each layer l, two transformation matrices will be trained, Ω^l and Φ^l. Thus, when the citation knowledge graph $\mathbb{CKG}$ evolves, we do not have to retrain all the models. Instead, we could construct or revise the corresponding computation graphs, and conduct forward propagation to compute the new embeddings of the papers. In short, GCN-TM model will be able to be transferred to unknown or new papers.

4) Mini-Batch: As shown in Fig. 3, we build a dedicated neural network architecture for each paper, thus all computation graphs can be calculated simultaneously. Particular, a common practice is to batch multiple computation graphs to form a mini-batch. Thus, how to batch multiple computation graphs to train them together is core to accelerate the computation. Leveraging the idea from the Deep Graph Library [7], we package a batch of computation graphs with disjointed nodes

in a larger graph, and align their adjacency matrices along the diagonal of a new adjacency matrix.

V. Experiments

In this section, we will describe and discuss in detail the evaluation of our GCN-TM model over real-world dataset, compared with state-of-the-art methods regarding paper recommendation effectiveness.

A. Experimental Settings

*1) **Dataset Preparation**:* Our testbed includes all published referred papers at top journal and conferences in the field of services computing: IEEE Transactions on Services Computing (TSC), IEEE International Conference on Web Services (ICWS), and IEEE International Conference on Services Computing (SCC). One reason why we selected this controlled testbed is that, we hope to carefully study the potential impact of our work on a research community.

We crawled the published refereed papers from IEEE eXplore digital library from the inception of the venues up to February 2021: TSC from 2008, IEEE ICWS and SCC both from 2004. Non-referred articles, such as editorial prefaces, panel or tutorial introductions, and chair messages were not included. We implemented a Web Spider based on Scrapy[9] in Python. By adding Selenium[10] in the middleware of Scrapy, our spider is able to fetch data loaded by Javascript on web pages. For each referred paper, we collected their metadata (including paper title, authors, venue, publication date, references, keywords, and abstract) as well as its textual content. The original content of each paper was crawled in HTML format, and we used html2text[11] package in python to convert them to pure text files as our LDA corpus. There are 23 referred papers with PDF version only, so we manually transformed them into HTML format.

*2) **Dataset Description**:* Our dataset preparation process resulted in a corpus including 4,090 papers. All papers become the original nodes of the Citation Knowledge Graph $\mathbb{CKG}$. For each paper $p \in \mathbb{CKG}$, we analyzed its references. If one reference paper q exists in the $\mathbb{CKG}$, a citation edge $p \leftarrow q$ was added into $\mathbb{CKG}$. Because we aim to study network structure to refine paper embeddings, we removed the paper nodes that do not connect to any other nodes in the network, either in or out. Thus, the number of nodes resulted in the $\mathbb{CKG}$ is 2,602. Table II summarizes the numerical properties of the resulted dataset for our experiments.

*3) **Evaluation Metrics**:* All papers were sorted chronically, from the past to current years. We innovatively treated each paper as a test case. Its topic distribution was used as a search query to request for recommendation on reference papers.

For each paper in the test set, we used our trained model to predict top 10 and 20 papers as candidate reference papers.

[9]https://scrapy.org/

[10]https://www.selenium.dev/

[11]https://github.com/Alir3z4/html2text/

TABLE II
Statistics of Papers CKG in Services Computing

Notation	Explanation
Papers	2,602
Citation edges	5,445
Highest degree	92
Lowest degree	1
Highest indegree	85
Lowest indegree	0
Highest outdegree	39
Lowest outdegree	0
Highest # nodes in $\mathbb{CG}$	582
Lowest # nodes in $\mathbb{CG}$	2
Diameter of CKG	11
Number of cliques	107

If a candidate paper was actually cited by the paper, it was considered a hit; otherwise it was a miss.

Recall that the mission of our GCN-TM is to build accurate representations for papers from the literature for the sake of effective recommendation. Although the accuracy of paper representations from audience's view is hard to measure directly, high-quality representations should be able to: (1) reflect the similarities of papers cite each other; and (2) reflect the differences among papers not cited often. Based on such a perspective, we followed SR-LDA [9] and designed a two-phase method as below to evaluate the quality of paper representations generated by GCN-TM.

First, we performed a K-means clustering method over the generated paper representations, to divide them into clusters. Second, we calculated the Davies-Bouldin index (DBI) [10] between each pair of paper clusters resulted. DBI is defined as follows:

$$\text{DBI} = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\Big(\frac{\text{avg}(\mathcal{C}_i)+\text{avg}(\mathcal{C}_j)}{d_{cen}(c_i,c_j)}\Big), \tag{5}$$

where

$$\text{avg}(\mathcal{C}) = \frac{2}{|\mathcal{C}|(|\mathcal{C}|-1)}\sum_{1\leq i<j\leq|\mathcal{C}|}\text{dist}(p_i,p_j)$$

is the average distance within cluster $\mathcal{C}$, K is the number of clusters, and $d_{cen}(c_i,c_j)$ represents the distance between the centers of two clusters c_i and c_j.

A lower value of DBI indicates more effective paper representations, meaning that paper within a cluster are more similar to each each, and the clusters are separated better, which corresponds to the aforementioned intuition.

*4) **Baselines**:* To evaluate the performance of the Top-K recommendation, we compared our GCN-TM with the following four baseline methods.

- **Vanilla LDA**. For this baseline, we applied the vanilla LDA to extract topic distributions of each paper. This baseline can provide an evidence for how well the vanilla LDA works in paper recommendation.
- **GraphSAGE**. For this baseline, we applied GraphSAGE alone to learn network structure of each paper in the CKG network. This baseline can provide an evidence

for how well the citation relationships work in paper recommendation.

- **Keyword-based Search**. For this baseline, we conducted keyword-based search for user queries.
- **Similarity-based Search**. For this baseline, we treated user search query as a vector, and find recommended papers by calculating the similarity (i.e., dot product) between the search vector and the papers.

*5) **Experiment Implementation**:* We implemented our GCN-TM on the basis of StellarGraph Machine Learning Library[12], a widely used Python library for machine learning on graphs and networks. We applied the implementation of the GraphSAGE carried by StellarGraph for graph convolution. During the tuning process, we found that 0.001 can be a good initial learning rate with an embedding size equal to the number of selected topics, respectively. Borrowing the idea of autoencoder, the transformation size sequence should be non-increasing. By shrinking the transformation size, each propagation process can learn more abstract features. Empirically, we halved the transformation size for each successive propagation layer.

B. Dataset Analysis

We scrutinized the resulted dataset $\mathbb{CKG}$ network. For each of the three venues (TSC, ICWS, SCC), we examined its published papers in each year. For each paper, we calculated its outdegrees, and summarized such information in line charts in Fig. 4(a), (b), (c), respectively. Take Fig. X(c) as an example for IEEE SCC papers, the horizontal axis indicates the number of outdegrees, and the vertical axis indicates the number of papers with a specific outdegree. As explained in the last section on experiment preparation, if a paper receives no citation and it does not cite any paper published in these three venues, the paper is not shown in the $\mathbb{CKG}$ network. In the line charts however, it should be counted back as a paper with zero (0) outdegree for the year. In its inception year, all SCC papers bear zero outdegree because ICWS papers are included from 2004 and TSC papers from 2008, thus their referenced papers are not included in the $\mathbb{CKG}$ network. As time goes by, an SCC paper naturally shall compare their work with related papers published in top venues in the field: TSC, ICWS, and SCC. As shown in Fig. 4(c), each year has one dedicated multi-point line chart.

As shown in Fig. 4(c), a notable amount of SCC papers do not cite any papers in TSC, ICWS, and SCC. This phenomenon in general should be brought to attention of the SCC program committee, since one core requirement for a top conference paper is to rigorously compare its research with related work. One may argue that the page limit (eight pages) of a conference paper constraints its citation and comparison with related work. This argument is partially supported by the fact revealed by Fig. 4(a) where TSC papers typically bear higher outdegrees, as TSC page limit is 14 pages. For example, the highest outdegree of TSC papers is 39. The

[12]https://stellargraph.readthedocs.io/en/stable/README.html

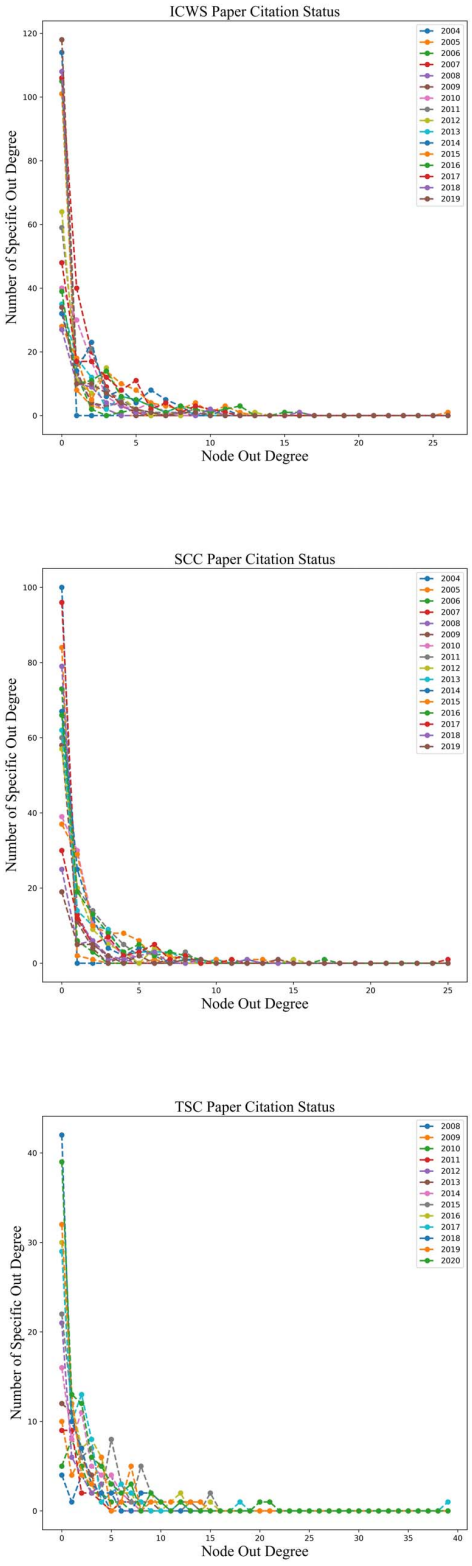

Fig. 4. **TSC/ICWS/SCC Paper Citation Analysis.**

organizing committee of the IEEE World Congress on Services (SERVICES), which endorses ICWS and SCC, has realized this constraint. Starting from 2021, the page limit has been revised to allow 10 pages for main text alone, plus reference pages. Our dataset study provides some historical data support for this committee decision.

C. Recommendation Performance

We evaluated our hypothesis of absorbing paper readers' views will help improve paper representation for recommendation. As shown in Fig. 5, we designed two scenarios, one is under LDA initiative and the other is under keyword initiative. Under the former LDA initiative, each paper takes its LDA topic distribution as its initial embedding. For each edge in the $\mathbb{CKG}$ network, i.e., pair of paper citation relation, we calculated the similarity (i.e., distance) of the nodes at the two ends. The distances of all edges are represented as the first boxplot in Fig. 5. After our GCN-TM training, for each edge we recalculated the similarity of its two ending nodes using their updated embeddings in the new space. The similarities of all edges are depicted by the second boxplot in Fig. 5. As shown in Fig. 5, 88.97% of such similarity goes higher significantly. The average increase is 62.65%. This experiment demonstrates that after aggregating topic information from neighbors (i.e., papers that cite it), the distance between a paper node and its neighbors becomes closer.

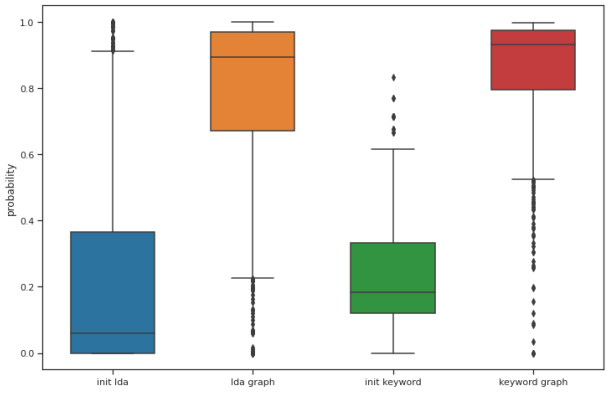

Fig. 5. **Graph Strengthened Paper Recommendation.**

We further demonstrated our hypothesis based on the keyword initiative. In contrast, each paper takes its IEEE keywords as its initial embedding, and its final embedding aggregates its neighbor embeddings from the $\mathbb{CKG}$ network. The similarities of the two papers with a citation relation using the initial and embeddings are represented by the two right boxplots in Fig. 5. 97.24% of such similarity goes higher significantly, and the average increase is 62.63%.

These two experiments demonstrated our hypothesis, that taking into account paper readers' views will help enhance paper representation for recommendation.

D. Loss Function

We examined the loss function used in our GCN-TM model for embedding learning impact. The hinge loss provides a semi-supervised learning approach thus shall be more stable. In our scenario, the citation relations are limited. In theory, the hinge loss shall be more appropriate. As a comparison, we experimented the binary cross entropy as the loss function as well. However, with the cross entropy method, the edge probability increases 88.97% after GCN-TM training; while the hinge loss only brings up 81.25%. We guess the reasons are two-fold. The first is that our testbed is relatively small. The second is that a paper may not cite another one with the highest similarity only based on word statistics.

E. DBI Analysis

The DBI results of our GCN-TM and the vanilla LDA are shown in Fig. 6. With different topic numbers (25 and 50), shown in Fig. 6(a) and Fig. 6(b), our GCN-TM always performs better under different number of clusters K, ranging from 10 to 200, with a step size 10. This experiment shows that considering the user views can increase the quality of paper representations toward paper recommendation.

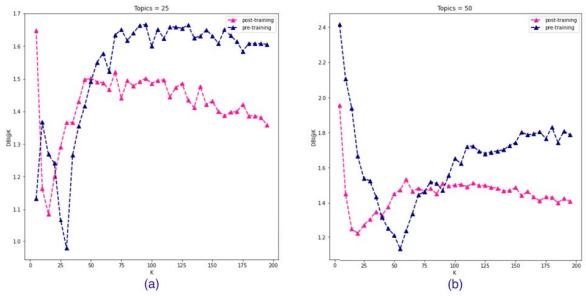

Fig. 6. **DBI Comparison between GCN-TM and Vanilla LDA.**

F. Case Study

To validate whether our resulted paper representations shall facilitate paper recommendation, we plotted the paper embeddings on 2-dimension using the Python t-Distributed Stochastic Neighbouring Entities (t-SNE)[13].

Fig. 7(a) and (b) show t-SNE plots of the testbed papers after applying Vanilla LDA, when the topic numbers are pre-set to 25 and 50, respectively. Particularly, the paper node highlighted blue is our identified paper with the highest outdegree in our testbed (cite most papers in the testbed); and the red nodes represent all papers that cited by it. Note that we enlarge the sizes of the highlighted nodes for better visualization. It can be seen that the cited nodes (i.e., papers highlighted in red) of the blue node are scattered in different clusters, most of them not located close to the sample node.

Fig. 7(c) and (d) show the resulted t-SNE plots of the testbed papers after applying our GCN-TM, when the topic numbers are pre-set to 25 and 50, respectively. The studied sample paper

[13]http://lvdmaaten.github.io/tsne/

node is again highlighted in blue, and its cited papers are highlighted in red as well. It can be easily visualized that most of the cited papers of the sample paper now move much closer to the sample paper in the embedding space.

This case study has demonstrated that the adjusted node embeddings have learned signals from their citation relationship; therefore, papers with citation relationships will become closer in the embedding space, and will assist in paper recommendation.

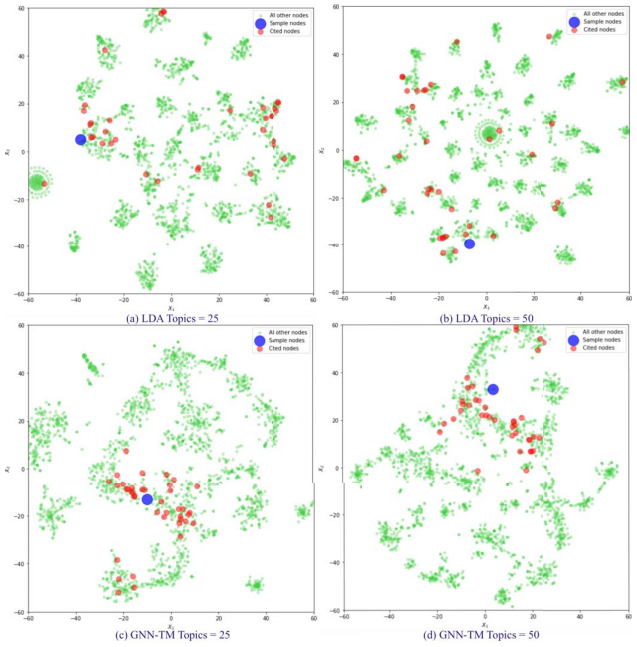

Fig. 7. **Case Study Using t-SNE.**

VI. Conclusions

Machine learning-based topic modeling techniques, represented by LDA, have been widely used to learn topic distributions as paper presentations to facilitate paper recommendation. Relying on word statistics however, such resulted paper presentations merely represent authors' views. How community users view the papers is valuable information, and may be of help to others to find interested papers to read and cite. Our hypothesis is that if paper A cites paper B, paper A thinks paper B as its related work. In other words, papers similar to paper A may also want to read and cite paper B. Thus in this paper, we have presented a novel technique that seamlessly integrate graph neural network into topic modeling, to refine paper representation from an audience perspective by learning from past paper citation relationships. Starting from paper content-based topic distributions, our GCN-TM model learns signals from papers that cite the paper and refines the paper representation. Our experimental results over the prestigious publication venues in the field of Services Computing (TSC, ICWS, SCC) have proved the effectiveness of our proposed GCN-TM model.

In our future work, we plan to further our research in the following three directions. First, we plan to consider different impacts (i.e., weights) over a paper from different papers. We may consider to apply the attention mechanism over the graph neural networks. Second, we plan to apply our technique to much broader domains, such as computer science and Earth science domains to evaluate the effectiveness and general applicability of our approach. Third, we plan to develop a portal to apply our technique to recommend papers to community users.

Acknowledgment

This work is partially sponsored by NASA 80NSSC21K0253 and 80NSSC21K0576.

References

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research, vol. 3, 2003, pp. 993-1022.

[2] A. Bandura, "Social Foundations of Thought and Action: A Social Cognitive Theory", Englewood Cliffs, N.J.: Prentice-Hall, 1986.

[3] Aditya Grover and Jure Leskovec, "Node2vec: Scalable Feature Learning for Networks", in Proceedings of The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016, pp. 855–864.

[4] Will Hamilton, Zhitao Ying, and Jure Leskovec, "Inductive Representation Learning on Large Graphs", in Proceedings of Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 1024–1034.

[5] Thomas N. Kipf and Max Welling, "Semi-supervised Classification with Graph Convolutional Networks", in Proceedings of 5th International Conference on Learning Representations (ICLR), 2017.

[6] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena, "Deepwalk: Online Learning of Social Representations", in Proceedings of The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2014, pp. 701–710.

[7] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J. Smola, and Zheng Zhang, "Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs", in Proceedings of ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[8] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua, "Neural Graph Collaborative Filtering", in Proceedings of The 42nd international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2019, pp. 165–174.

[9] J. Zhang, Y. Fan, J. Zhang, and B. Bai, "Learning to Build Accurate Service Representations and Visualization'", IEEE Transactions on Services Computing, in press.

[10] David L. Davies and Donald W. Bouldin, "A Cluster Separation Measure", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 2, 1979, pp. 224–227.