Received: 3 July 2020 | Revised: 13 January 2021 | Accepted: 24 January 2021
DOI: 10.1002/cpe.6227

RESEARCH ARTICLE

Content-based and knowledge graph-based paper recommendation: Exploring user preferences with the knowledge graphs for scientific paper recommendation

Hao Tang | Baisong Liu | Jiangbo Qian

Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China

Correspondence
Baisong Liu, Faculty of Electrical Engineering and Computer Science, Ningbo University, Fenghua Road, Jiangbei District, Ningbo, Zhejiang, China.
Email: lbs@nbu.edu.cn

Funding information
Zhejiang NSF, Grant/Award Number: LZ20F020001; China NSF, Grant/Award Number: 61472194; K.C. Wong Magna Fund in Ningbo University

Abstract

Researchers usually face difficulties in finding scientific papers relevant to their research interests due to increasing growth. Recommender systems emerge as a leading solution to filter valuable items intelligently. Recently, deep learning algorithms, such as convolutional neural network, improved traditional recommendation technologies, for example, the graph-based or content-based methods. However, existing graph-based methods ignore high-order association between users and items on graphs, and content-based methods ignore global features of texts for explicit user preferences. Therefore, this paper proposes a **C**ontent-based and knowledge **G**raph-based **P**aper **Rec**ommendation method (**CGPRec**), which uses a two-layer self-attention block to obtain global features of texts for more complete explicit user preferences, and proposes an improved graph convolutional network for modeling high-order associations on the knowledge graph to mine implicit user preferences. And the knowledge graph in this paper is constructed with concept nodes, user nodes, paper nodes, and other meta-data nodes. Experimental results on a public dataset, CiteULike-a, and a real application log dataset, AHData, show that our model outperforms compared with baseline methods.

KEYWORDS
graph neural network, high-order associations, knowledge graph, scientific paper recommendation, self-attention mechanism.

1 | INTRODUCTION

With the development of information technology and the Internet of Things,[1,2] we face an information explosion. It is difficult for researchers to find out what articles they really care about from a large number of scientific papers. Scientific papers are essential for researchers to keep up with the latest research progresses relevant to their research areas. Paper recommender systems have simplified the above process, effectively producing relevant papers for researchers.[3] Existing recommendation methods[4] could be divided into Collaborative Filtering (CF) methods, Content-Based Filtering (CBF) methods, and Graph-Based (GB) methods.

CF methods think that users with similar behaviors to items could have the same preferences. Generally, a recommendation score that measures how much a user likes a paper is gained by calculating the similarity between the user vector and the paper vector. Nazmus et al.[5] proposed a CF method for paper recommendation using citation context rather than nonpublic information due to private preservation.[6] Suffering from data sparsity, existing CF models usually perform weakly. Recently, deep learning technologies with strong learning capacity have remarkably improved the performance of some CF methods. For example, Travis Ebesu et al.[7] proposed a collaborative memory network that has achieved excellent results, which unified the global factor model and local CF models. However, CF models would ignore the most explicit and implicit association information between users and papers when just providing interaction data.

CBF methods are relatively mature both in academic and industry fields, which recommend papers similar to items browsed by the user. CBF methods usually obtain a matching score between the user profile and the paper. For example, ER[3] algorithm combines content features with non-content features for obtaining a user profile, which aims to recommend multiple types of academic resources. The Microsoft academic recommender system[8] is a hybrid content-based and graph-based system, in which most papers are recalled by a CBF method for enhancing its recommendation coverage. Though CBF methods are beneficial in paper recommendation scenarios, which appears difficult to capture diverse user interests only using text semantic similarity.[4] Besides, users have definite reading purposes when searching for papers, but CBF methods would limit users in a very narrow reading vision referring to individuals' background knowledge.[9]

GB methods represent users and papers into nodes on graphs instead of specific user behaviors and paper content. In paper recommendation scenarios, the graphs usually include citation networks, social networks, and heterogeneous information networks. Cai et al. represent related recommendation elements as nodes and associations as edges on a graph, such as query texts, papers, authors, and venues etc., could be constructed into a heterogeneous network[10] or a bibliographic network,[11] which could provide extra features by a graph representation embedding for a paper recommendation. Liu et al.[12] utilize the correlation relations between two papers on an undirected citation graph, while HGRec[13] thinks the heterogeneity graphs more than only citation relations. Waleed et al.[14] use both the citation network and the relationship network between authors to find significant papers. Though these approaches above are popular and effective, they have wasted users' interaction data and papers' content characteristics, resulting in a cold start problem because new papers have fewer citations.

The emergence of representation learning technology allows side information to easily be embedded into recommender systems to alleviate data sparsity. As excellent auxiliary information, knowledge graphs (KGs) are rich in entities and relations. Zhao et al.[15] construct a concept-level KG to fill the knowledge gap between users and papers , and then extract user-paper semantic paths from the KG that fit users' cognitive patterns to acquire target papers. Frederick et al.[16] links terms to an external KG, DBpedia, which could provide recommender systems more semantic features of query manuscripts. However, the methods above fail to consider high-order associations between user entities and paper entities on KGs.

This article focuses on the following two aspects: (1) How to recommend relevant papers to users on the premise that interaction history is sparse, (2) how can researchers comprehensively obtain papers related to their domain with limited domain knowledge. This article proposes a Content and Knowledge Graph-based Paper Recommendation model (CGPRec) based on previous study progress. First, the model uses Word2vec and DCNN (Double Convolutional Neural Network) to process text for sentence representation, inspired by DKN[9] using multichannel CNN. Kim CNN[17] only focuses on more local features in a text but ignores global features, and the complete user preferences cannot be obtained. Thence we propose a feature extraction module with two layer self-attention to supplement explicit user preference. With an external knowledge database, CGPRec extracts concepts from paper texts to construct a concept-level KG with user nodes, paper nodes, and other metadata nodes. Finally, CGPRec uses an improved graph convolutional network (GCN) to learn high-order associations between users and papers on the KG because it can effectively propagate high-order node embeddings.[18-21] The main contributions of this article are as follows:

(1) This article proposes a new paper recommendation model, CGPRec, based on both knowledge graph features and text features. The explicit user preferences are generated separately from the local and global text features, where a two-layer self-attention mechanism is proposed to model global text features.
(2) This article uses unstructured data and semi-structured data of papers to build a KG with the linked open data (LOD). To obtain implicit user preferences, we propose a neighborhood aggregation to improve the GCN, which could not consider various relation types in the KG.
(3) The experiments on real-world datasets demonstrate that our model produces a noticeable improvement in accuracy and click rate, compared with traditional models and KG improved models.

2 RECOMMENDATION WITH KGS

We divided the current recommendation methods using KGs into two categories. The first is the methods of designing and using semantic paths on KGs. Researchs[22-24] use meta-paths to calculate the relevance of users and items under similar paths. PER[22] diffuses user preferences along different meta-paths and outputs implicit representations of users and items under the semantic assumption of Matrix Factorization (MF). HINE[25] method uses a random walk strategy based on meta-path to integrate the node sequence into the extended MF model. The article[26] proposed a general rule derivation module, which uses random walks to generate the probability of each path between two items, improving Bayesian Personalized Ranking (BPR) and Neural Collaborative Filtering (NCF). This method automatically learns user preference rules on the knowledge graph to complete the collaborative filtering of existing neural network models. However, the path design process above is complicated and requires domain knowledge, ignoring other semantic paths in the user-item graph.

The second is the methods with feature representation learning. Such methods map entities and relationships into low-dimensional and dense vectors to enhance the representation of users and items. TransE-CF[27] integrates semantic neighbors between items and corresponding entities to solve the cold start problem to a certain extent. CKE[28] incorporates text information, structural information, image information simultaneously in collaborative filtering and uses TransR to learn structural information representation. DKN[9] learn news semantics by a three-channel CNN, which combines the entity characteristics of a knowledge graph. As the item-based CF only uses collaborative similarity relationships and ignores the attributes of multiple relationships between items in the real scene, RCF[29] builds a knowledge graph based on items and relations to consider relationship types and relationship values hierarchically. For data sparsity and cold start problems, Wang proposed KGCN[18] and KGCN-LS[31] to automatically mine high-order structural information between users and items on KGs with GCNs to obtain implicit user preferences. To make full use of the relations between items or instances, KGAT[20] builds a collaborative knowledge graph based on the user-item bipartite graph and the item knowledge graph for high-order associations. MKR[32] design a general cross-compression unit as an information interaction channel for KG representation learning task and recommendation task.

3 | PROBLEM DEFINITION

Assume that there are N users and M papers in a paper recommendation system, in where $U = \{u_1, u_2, \dots, u_N\}$ and $P = \{u_1, u_2, \dots, u_N\}$. For $u \in U$ and $p \in P$ the interaction is expressed as:

$$y_{up} = \begin{cases} 1, & u \text{ had interacted with } p \\ 0, & \end{cases}, \tag{1}$$

User behaviors could be implicit feedbacks or explicit ratings. This article chooses the former because it is closer to actual scenes. According to the Equation (1), one user's all implicit feedbacks form into a user-paper interaction matrix, $Y \in \mathbb{R}^{m\times n}$, where $y_{ij} \in Y$ represents the interaction between the ith($i = 1, 2, \dots, m$) user and the jth($j = 1, 2, \dots, n$) paper. For the graphs in this article, our model converts Y to a user-paper bipartite graph G_1. An edge in the graph represents a user interaction with an item. The graph with more nodes (such as concepts, keywords, and instances, etc.) and their corresponding relation forms a knowledge graph $G = \langle V, E\rangle$, in where V represents an entity set and E represents a relation set. The text data form as $C = \{T, A, K\}$, where T denotes a title set, A denotes an abstract set, and K denotes a keyword set. Given a user $u \in U$ and a candidate paper $p \in P$, our model aims to predict u's interest to p based on graphs and text characteristics. The objective function is denoted as $\tilde{y}_{u,p} = F(u, p|\theta, G, C)$, where $\tilde{y}_{u,p}$ represents the click probability of user u on the paper p, and θ is the model train parameter.

4 | PROPOSED METHOD

4.1 | CGPRec framework

The framework of CGPRec is shown in Figures 1 and 2. User preference representations include explicit preferences from paper content and implicit preferences from the high-order association on a knowledge graph. The former is composed of text local and global features, as shown in Figure 1.

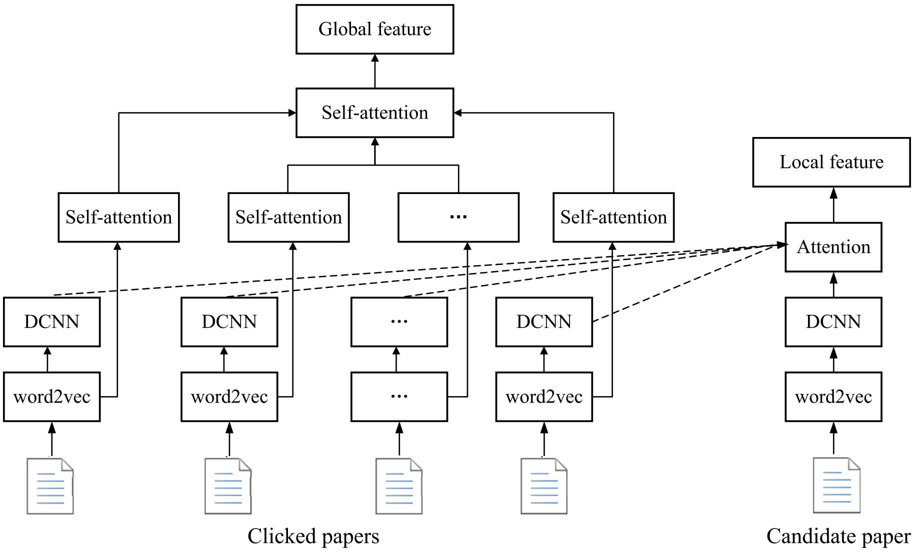


FIGURE 1 Global and local text features for explicit user references

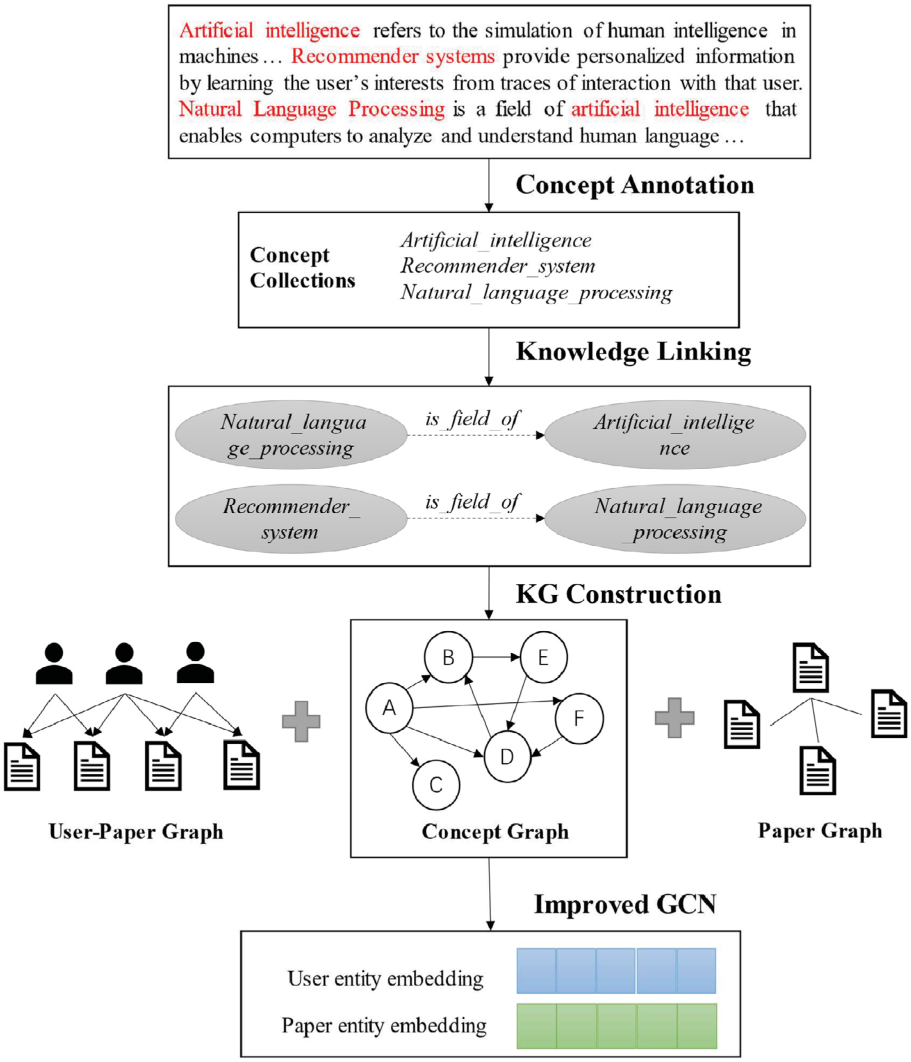

Artificial intelligence refers to the simulation of human intelligence in machines… Recommender systems provide personalized information by learning the user's interests from traces of interaction with that user. Natural Language Processing is a field of artificial intelligence that enables computers to analyze and understand human language…
Concept Annotation
Concept Collections
Artificial_intelligence
Recommender_system
Natural_language_processing
Knowledge Linking
Natural_language_processing
is_field_of
Artificial_intelligence
Recommender_system
is_field_of
Natural_language_processing
KG Construction
B
E
A
F
D
C
User-Paper Graph
Concept Graph
Paper Graph
Improved GCN
User entity embedding
Paper entity embedding

FIGURE 2 High-order associations from the knowledge graphs for implicit user preferences

Implicit user preferences are included in one user's node vector after the knowledge graph is processed by the improved GCN. This article represents all users, papers, and their relevant metadata as nodes on a knowledge graph, as shown in Figure 2. Finally, we use a multilayer perceptron (MLP) to learn the matching function and output the correlation score.

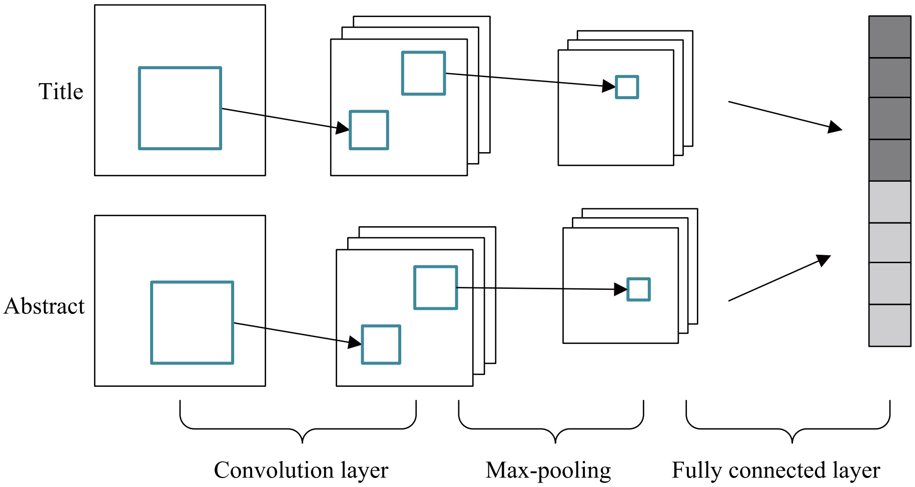
4.2 | Explicit user preferences

Our model synthesizes explicit user preferences from two aspects: local text features and global text features. The local features aim to find the most important token features in the sentence, and the global features focus on the semantic features of the entire sentence. To represent a paper in word space, we use title, abstract as well as keywords, which are easy to find to alternative key content features in one paper. This paper proposes a local feature extractor DCNN to obtain the semantics of candidate papers and user click history. And its input is a sentence representation matrix composed of word vectors by word2vec. The attention mechanism combines the click history of a specific user with the candidate paper to train the representation vector of local text features. Moreover, one user's history click papers are processed by the proposed two-layer self-attention block to output the final global text features representation vector. Explicit user preferences are represented as a combination of two vectors.

4.2.1 | Local feature extractor DCNN

Drawing on the views of DKN[9] and GNewsRec[33] which use CNN[17] to construct text feature extractors, our model proposes a local text feature extractor DCNN. Its structure is shown in Figure 3. The title and abstract in a paper have different characteristics, including input, respectively, to

FIGURE 3 The structure of the double convolutional neural network



parallel CNNs with independent parameters. The output is used as a local feature representation of the paper. Given a title matrix and an abstract matrix denoted as $T = [w_{t1}, w_{t2}, \dots, w_{tn}] \in \mathbb{R}^{d \times n}$ and $A = [w_{a1}, w_{a2}, \dots, w_{am}] \in \mathbb{R}^{d \times m}$, the columns of which are the word vectors trained by word2vec. The CNN in DCNN consists of a convolutional layer, a pooling layer, and several fully connected layers. For paper p, DCNN separately processes its title and abstract, then outputs $\tilde{T}$ and $\tilde{A}$, And the output of DCNN is denoted as $e(p) = [\tilde{T}; \tilde{A}] \in \mathbb{R}^D$, where D is the dimension of the output vector.

To judge which extent a candidate paper satisfies the user's historical interests, our model distinguishes the contribution of click history to user interests based on the attention mechanisms. We represents u's click history as $\{p_1^u, p_2^u, \dots, p_k^u\}$, and the click history embeddings processed by DCNNs as $\{e(p_1^u), e(p_2^u), \dots, e(p_k^u)\}$, the candidate paper embeddings as $e(p_j)$. Finally, we concatenate (p_i^u) with $e(p_j)$, which is input to a neural network for nonlinear transformation. The normalized weight score is calculated by the softmax function:

$$s_{p_i^u,p_j} = \text{softmax}(\text{DNN}(e(p_i^u), e(p_j))), \tag{2}$$

$$\text{softmax}(\text{DNN}(e(p_i^u), e(p_j))) = \frac{\exp(\text{DNN}(e(p_i^u), e(p_j)))}{\sum_{i=1}^{k} \exp(\text{DNN}(e(p_i^u), e(p_j)))}, \tag{3}$$

Therefore, the explicit reading preference obtained from the local features are expressed as follows:

$$\mathbf{u}_1 = \sum_{i=1}^{k} s_{p_i^u,p_j} * e(p_i^u), \tag{4}$$

4.2.2 | Global feature extractor

To obtain the global features, CGPRec firstly utilizes self-attention mechanism to process short texts consisting of titles, keywords, and abstracts. The self-attention mechanism effectively obtains the long-distance dependence of a sentence and determines the importance of one word with global information. This paper uses a metric function to calculate a score for each token, denoted as $f_{\text{self}}(\mathbf{w}_i)$. The score indicates the importance of a token in the sentence, which needs no additional guide information. Taking the representation of the tth title as an example, the weight of the ith token is:

$$a_{ti}^{\text{self}} = \frac{\exp(f_{\text{self}}(\mathbf{w}_{ti}))}{\sum_{i=1}^{n} \exp(f_{\text{self}}(\mathbf{w}_{tk}))},$$

Then the sentence with global dependency is represented as:

$$\mathbf{S}_t^{self} = \sum_{i=1}^{n} a_{ti}^{self} * \mathbf{w}_{ti}, \tag{5}$$

To find one user's reading intent affected by different click papers, the model designs a global feature extractor with a two-layer self-attention layer. When user u has clicked h papers, his explicit preferences of reading obtained from global characteristics is expressed as follows:

$$\mathbf{u}_2 = \sum_{t=1}^{h} a_t^{\text{self2}} * \mathbf{S}_t^{\text{self}} \tag{6}$$

in where a_t^{self2} is the weight of the sentence. The local and global features of the previous section are stitched together and input into a fully connected network to obtain the user's final explicit reading preference, $\mathbf{u}_{et} = \mathbf{W}_1 * [\mathbf{u}_1; \mathbf{u}_2]$, where $\mathbf{W}_1 \in \mathbb{R}^{D\times 2D}$.

4.3 | Implicit user preferences

For more complete user preferences through high-order associations on KGs, this section constructs a knowledge graph consisting of users, papers, related concepts, and paper metadata. We propose an improved GCN to learn user node representations that contain high-order association information, which implies potential user interests.

4.3.1 | Knowledge graph construction

LOD stores rich entities and relations knowledge. In this article, the concepts and other entities obtained from papers are used to construct knowledge graph G, which aims to correlate the known concepts with unknown related concepts and integrate them into users and papers representation through high-order associations more complete user preferences.

For titles, abstracts, keywords, and other text data, we first extract the words with higher TF-IDF weights and then link the accurate concepts in LOD and several hopping neighborhoods, such as the hypernyms of the linked concepts. Regarding the relation between the user and the paper, there is a labeled edge "interaction" between the user node and the paper node if there is an interaction. We merge the above conceptual subgraph with a user-paper bipartite graph G_1, forming the final knowledge graph G, which contains entity types: users, papers, concepts, and other metadata. The relation types in the KG include the relationships as user-paper interaction relation, the paper-paper citation relation, the "contain" relation between concepts and papers, the *is_A* relation between instances and concepts, and the subordinate relation between the concepts. Therefore, through the organization of the knowledge graph, the semantics contained in the title, abstract, and keywords of one paper can be displayed through the fined-grain correlation between the concepts.

4.3.2 | The improved GCN

GCN could learn high-order associations between entities. First, GCN obtains a distributed representation of users and papers through propagation embedding, which contains the user's potential reading preferences, that is, the user's implicit preferences mentioned above. However, GCN generally treats the knowledge graph as an undirected graph and ignores the distinction between relation types. Therefore, this article considers a user's implicit preference distribution for all relations in advance.[18,20] The following is the general form of computing a node v embedding in a single GCN layer:

$$\mathbf{h}_{N_v} = f_{agg_N}(\{\mathbf{e}_v, \mathbf{e}_{N_v}\}), \tag{7}$$

$$\mathbf{h}_v = \sigma(\mathbf{W}_2 * \mathbf{h}_{N_v} + \mathbf{b}_1), \tag{8}$$

where $f_{agg_N} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$,represents the neighborhood aggregation function used to aggregate information from the neighborhoods. We select the *Concat aggregator*[18] as an aggregation function that concatenates two vectors before the nonlinear transformation. In this paper, the e_v and e_{N_v} are vector representations of item v and its adjacent topological structure N_v, respectively. Given the neighborhood set $N_v = \{(r, ent)|(v, r, ent) \in G, (ent, r, v) \in G\}$ with k nodes. In general, e_{N_v} is computed in an average manner:

$$e_{N_v} = \frac{1}{k} * \sum_{i=1}^{k} (e_{ent})_i. \tag{9}$$

Our model incorporates the relation types when calculating neighborhood information. Each neighborhood entity's contribution to the neighborhood representation depends on the matching value of the user entity and the relation. For example, a user prefers to find papers referring to the relation, which affects embedded propagation in the embedding process. We define a mapping $f_{ur} : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ (e.g., inner product) to calculate the matching value of user u and relation r. Therefore, the vector of the neighborhood is expressed as:

$$e_{N_v} = \sum_{(r,ent)\in N_v} f_{ur}(\mathbf{u}, \mathbf{e}_r) * ent \tag{10}$$

After aggregation with L_1 layers neighborhood, the paper entity vector is denoted as $\mathbf{e}_p$, and the user entity vector, that the implicit user preferences from high-order associations on KGs, is expressed as $\mathbf{u}_imt$. Finally, the paper representation vector is denoted as $\mathbf{p} = \mathbf{e}_p$, the user representation vector is $\mathbf{u} = \mathbf{W}_3 * [\mathbf{u}_{et}; \mathbf{u}_{imt}], \mathbf{W}_3 \in \mathbb{R}^{D \times 2D}$.

4.4 | Recommendation calculation

Existing methods based on deep learning focus on two aspects when calculating a recommendation list, user and item representation learning or the user-item interaction modeling. Unlike previous studies, the above introduction shows that CGPRec has learned the representation of users and papers. Now we use the trained user vectors and paper vectors to make recommendation predictions. In the interaction modeling, based on the user-paper interaction pair, this paper stitches the trained user vector and paper vector as the input $\mathbf{x}_1$ of the interaction modeling layer and obtains the prediction score after L_2 times nonlinear transformation. Through the above calculations, the user vector is ultimately expressed as $\mathbf{u}$ and the candidate paper vector is $\mathbf{p}$. The final user-paper pair vector is input into the interactive modeling layer, an MLP, for calculation:

$$\begin{aligned}
\mathbf{x}_1 &= [\mathbf{u}; \mathbf{p}], \\
\mathbf{x}_2 &= a_1(\mathbf{W}_4 * \mathbf{x}_1 + \mathbf{b}_2), \\
&\cdots \cdots, \\
\mathbf{x}_{L_2} &= a_{L_2-1}(\mathbf{W}_{L_2+1} * \mathbf{x}_{L_2-1} + \mathbf{b}_{L_2}), \\
\tilde{y}_{u,p} &= \sigma(a_{L_2}(\mathbf{h}_{L_2}^T \mathbf{x}_{L_2})),
\end{aligned} \tag{11}$$

where $\mathbf{W}_i$, $\mathbf{b}_i$, and σ represent the weight matrix, bias vector, and activation function of the ith-level perceptron, respectively.

To effectively train, unknown papers are sampled from implicit feedback for specific users as negative samples, which is the same number as positive samples. For example, a training sample can be represented as (u, x, y), where x is a candidate paper that predicts whether to click. For each positive sample, $y = 1$, otherwise $y = 0$. We use cross-entropy as a loss function:

$$L = -\left\{ \sum_{x \in \triangle^+} y_{u,p} \ln \tilde{y}_{u,p} + \sum_{x \in \triangle^-} (1 - y_{u,p}) \ln(1 - \tilde{y}_{u,p}) \right\} + \lambda \|\mathbf{W}\|_2, \tag{12}$$

where $\triangle^+$ is a set of positive samples, $\triangle^-$ is a set of negative samples, and $\lambda\|\mathbf{W}\|_2$ is L_2 regular term.

5 | EXPERIMENTAL RESULTS

This section gives the experimental design details and corresponding results. To prove the effectiveness of our model, we compared CGPRec with the benchmark models in the experiment. This paper will analyze the experiment from the following two Research Questions (RQs).

- RQ1: How to effectively recommend a paper on the premise that the user-paper interaction record is very sparse? That is, compared with the benchmark model, does the experimental effect of the model on sparse data sets exceed the performance of state-of-the-art?
- RQ2: How can researchers obtain more diverse papers in a known limited domain knowledge? That is to say, what is the impact of the components of the CGPRec model on the model, especially the integration of the knowledge graph to improve the experimental results?

5.1 | Experimental design

Datasets We use two common paper recommendation public datasets, CiteULike-a and AHData, to verify our model. CiteULike-a is an online article storage and sharing platform that allows users to create paper collections that they are interested in. The reason for choosing the platform data is that the users' paper collections could reflect true reading preferences and provide the titles and metadata. CiteULike-a is an implicit feedback data set collected and preprocessed from the platform by reference 30. The AHData is a recommendation application deployed in some academic institutions that we are maintaining. The experimental dataset is the click logs generated after the user interacts with the system.

Knowledge graph. For knowledge graph construction, the text is cleaned and extracted concepts in order. Then we select the noun terms with higher weight. The terms are linked to an external knowledge base, Xlore, to obtain semi-structured data, clean and keep the triples. The statistics are shown in Table 1. Finally, the data is divided into a training set, a validation set, and a test set by a 7:2:1 ratio, and the validation set is used to optimize hyperparameters.

Dataset	User	Paper	Interactions	Sparsity
CiteULike-a	5551	16,980	204,986	99.78%
AHData	5000	20,000	189,141	99.81%

TABLE 1 The statistics of datasets

	Prec@N		Rec@N		F1-score		
Model	$N = 5$	$N = 10$	$N = 5$	$N = 10$	$N = 5$	$N = 10$	AUC
BPRMF	0.179	0.104	0.113	0.194	0.139	0.135	0.508
NeuMF	0.204	0.132	0.120	0.189	0.151	0.155	0.515
CML	0.262	0.145	0.188	0.227	0.219	0.177	0.534
KGAT	0.233	0.204	0.220	0.289	0.226	0.239	0.556
DKN	0.422	0.346	0.383	0.407	0.402	0.374	0.603
CGPRec	**0.459**	**0.374**	**0.438**	**0.487**	**0.448**	**0.422**	**0.703**

TABLE 2 Experimental results about the *CiteULike-a*

5.1.1 | Baselines

- BPRMF: A standard matrix factorization based on Bayesian optimization. This paper uses user-paper interaction matrix *Y*.
- NeuMF:34 A instance of the NCF framework, which combines generalized matrix factorization and MLP at the user and item embedding layers. This paper uses the same inputs as BPRMF.
- CML:[35] A metric learning algorithm that simultaneously encodes user preferences and user-user, item-item similarity. This paper uses the same input as BPRMF.
- KGAT:[20] It explicitly models the high-order associations between users and items on the knowledge graph, using an aggregation method of attention.
- DKN:[9] A content-based deep learning recommendation framework that integrates multichannel CNNs to represent the semantic and knowledge layers of a paper. In this paper, the features of content *C* are taken as the features of the semantic layer, and the features of the knowledge graph *G* are taken as the features of the knowledge layer.

5.1.2 | Evaluation indicators

The *precision* indicates the proportion of papers predicted to be true by the recommendation list, and the *recall* is an evaluation index of coverage, indicating the proportion of papers predicted to be true in the recommendation list to all papers related to the paper. *F1-score* is the weighted average of the *precision*and the *recall*. The larger the value, the more accurate the result. *AUC*: The area under the ROC curve, where the abscissa of the ROC curve is the false positive rate of the prediction result, and the ordinate is the true positive rate.

5.1.3 | Experimental results (RQ1)

First, this article introduces the overall performance compared with baselines on the CiteULike-a and AHData, as shown in Tables 2 and 3. The best performance is shown in bold.

Comparing the results of this model and other baselines, F1-score, and *AUC* in the two datasets are better than most baselines, respectively. The results with BPRMF, NeuMF, and CML show that the performance of CGPRec does not suffer an enormous impact when the data is sparse (see Table 1), which can better alleviate the problem of data sparseness. The possible reason is that the model in this paper has more abundant content features and knowledge features. Although DKN also has sufficient content features, the reason why the CGPRec model results are better may be due to global text features. Both KGCN and CGPRec use graph neural networks. The reason for the experimental difference may be the fusion of content features. The experiment also found that all content-based models have better performance than CF-based models. The reason is that the CF-based method is affected by the performance of recommended scenarios with sparse data. Our model is a hybrid model that combines the advantages of content-based and graph-based methods, so for the papers without a click history, the text and the graph can produce relation.

TABLE 3 Experimental results about the *CiteULike-a*

	Prec@N		Rec@N		F1-score		
Model	$N=5$	$N=10$	$N=5$	$N=10$	$N=5$	$N=10$	AUC
BPRMF	0.296	0.190	0.129	0.153	0.180	0.170	0.532
NeuMF	0.300	0.236	0.157	0.246	0.206	0.241	0.557
CML	0.389	0.311	0.245	0.266	0.301	0.287	0.595
KGAT	0.422	0.366	0.287	0.377	0.342	0.371	0.588
DKN	0.471	0.452	0.413	0.426	0.440	0.439	0.606
CGPRec	**0.599**	**0.488**	**0.571**	**0.635**	**0.585**	**0.552**	**0.686**

TABLE 4 Experimental results of CGPRec variants on the *AHData*

Model	Pre@10	Rec@10	F1-score	AUC
RGF	0.349	0.354	0.352	0.575
RSA	0.404	0.439	0.421	0.521
WTE	0.442	0.427	0.434	0.553
WIP	0.470	0.589	0.523	0.688
CGPRec	**0.488**	**0.635**	**0.552**	**0.703**

TABLE 5 The GCN and multilayer perceptron (MLP) layer number on the *AHData*

Model	Layer	Pre@10	Rec@10	F1-score	AUC
GCNL	$L_1=1$	0.472	0.549	0.508	0546
	$L_1=2$	**0.488**	**0.635**	**0.552**	**0.686**
	$L_1=3$	0.432	0.619	0.509	0.585
MLP	$L_2=1$	0.404	0.589	0.479	0.588
	$L_2=2$	0.436	0.611	0.510	0.593
	$L_2=3$	**0.488**	**0.635**	**0.552**	**0.686**

5.1.4 | Model analysis (RQ2)

Taking the *AHData* dataset as an example. The experimental results of CGPRec variants are compared to prove that the model design has the following validity: (a) The experimental results can be improved by extra content features. (b) The effect of the self-attention mechanism on the user's explicit preference. (c) The effect of the high-order association compared with the classic KG representation learning method. (d) The impact of MLP on interaction modeling. Moreover, the experimental results are shown in Table 4, and the detailed settings are as follows:

- **Remove Graph Features (RGF)**: Only keep the effect of the user's explicit reading preference module described in Section 3.2.
- **Remove Self-Attention (RSA)**: Remove the effect of global text features of user explicit reading preference module described in Section 3.2.
- **Replace GCN with TransE[36] (WTE)**: Replace the TransE instead of GCN described in Section 3.3 to get the user's high-order user preference.
- **With Inner Product (WIP)**: Replace MLP with the inner product.

It can be concluded from Table 4:

- The CGPRec performs best, indicating the effectiveness of different components of the model;
- The lack of global features of the text has a greater impact on the results;
- The embedding of the knowledge graph greatly improves the performance of the GCN in this paper and the results is better than TransE.

Parameter analysis. CGPRec involves the selection of multiple parameters. Next, taking *AHData* as an example, the model considers the impact of L_1 and L_2 to the evaluation indicators *F1-score* and *AUC*. The results are shown in Table 5.

6 | CONCLUSIONS

The enormous amount of information over the internet makes it difficult for researchers to locate the most important scientific papers for their current work or study. This study has successfully proposed a hybrid recommendation model, CGPRec. Firstly, the tow-layer self-attention mechanism considers global text features and can obtain complete explicit user reading preferences. Secondly, our model links the concepts extracted from papers to the external knowledge base and obtains the potential correlation between users and papers. At the same time, the integration of knowledge effectively alleviates data sparsity. The high-order association obtained by propagating embeddings can effectively mine implicit user reading preferences. On the real datasets in the paper recommendation, CiteULike-a and AHData, the experimental results show that our model is significantly better than the baselines in *F1*-score and *AUC* indicators. To some extent, CGPRec can expand user profile with implicit user preferences hidden in the KG, which generate diverse recommendation candidates, while the CBF approach produce similar results all the time. However, incomplete or inaccurate KGs could introduce errors into the recommendation process, and large data volume of KGs cause complicated calculation. This stage integrates the subgraphs of the knowledge graph into the representation of user portraits and improves the performance of content-based methods in public datasets and real recommendation applications. Compared with current methods with better performance, the results have certain advantages. In future work, we will try to reduce the computational complexity of knowledge graph representation learning.

ACKNOWLEDGEMENTS

This work was supported in part by Zhejiang NSF Grant No. LZ20F020001, China NSF Grants No. 61472194 as well as programs sponsored by K.C. Wong Magna Fund in Ningbo University.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Hao Tang https://orcid.org/0000-0002-0159-4360
Jiangbo Qian https://orcid.org/0000-0003-4245-3246

REFERENCES

1. Cai X, Geng S, Wu D, Cai J, Chen J. A multi-cloud model based many-objective intelligent algorithm for efficient task scheduling in Internet of Things. *IEEE IoT J*. 2020. https://doi.org/10.1109/JIOT.2020.3040019.
2. Hassan MU, Rehmani MH, Chen J. DEAL: differentially private auction for blockchain based microgrids energy trading. *IEEE Trans Serv Comput*. 2020;13(2):263-275.
3. Yin L, Liu B, Wang Y. Research on cross-type excellent recommendation algorithm for academic resources. *J China Soc Sci Techn Inf*. 2017;36(7):715-722.
4. Bai X, Wang M, Lee I. Scientific paper recommendation: a survey. *IEEE Access*. 2019;7:9324-9339.
5. Nazmus S, Rodina B, Khalid H. A collaborative approach toward scientific paper recommendation using citation context. *IEEE Access*. 2020;8:51246-51255.
6. Hassan M, Rehmani M, Chen J. Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions. *Future Generat Comput Syst*. 2019;97:512-529.
7. Ebesu T, Shen B, Fang Y. Collaborative memory network for recommendation systems. *Research & Development in Information Retrieval*. New York, NY: ACM; 2018:515-524.
8. Kanakia A, Shen Z, Eide D. A scalable hybrid research paper recommender system for microsoft academic. Paper presented at: Proceedings of the 2018 World Wide Web Conference on World Wide Web. San Francisco, CA; 2019:2893-2899.
9. Wang H, Zhang F, Xie X. DKN: deep knowledge-aware network for news recommendation. Paper presented at: Proceedings of the 2018 World Wide Web Conference on World Wide Web. Lyon, France; 2018:1835-1844.
10. Cai X, Han J, Pan S. Heterogeneous information network embedding based personalized query-focused astronomy reference paper recommendation. *Inte J Comput Intell Syst*. 2018;11(1):591-599.
11. Cai X, Zheng Y, Yang L. Bibliographic network representation based personalized citation recommendation. *IEEE Access*. 2019;7:457-467.
12. Liu H, Kou H, Yan C, Qi L. Keywords-driven and popularity-aware paper recommendation based on undirected paper citation graph. *Complexity*. 2020;2020:1-15.
13. Ma X, Ranr W. Personalized scientific paper recommendation based on heterogeneous graph representation. *IEEE Access*. 2019;7:79887-79894.
14. Waleed W, Muhammad I, Basit R. A hybrid approach toward research paper recommendation using centrality measures and author ranking. *IEEE Access*. 2019;7:33145-33158.
15. Zhao W, Wu R, Liu H. Paper recommendation based on the knowledge gap between a researcher's background knowledge and research target. *Inf Process Manag*. 2016;52(5):976-988.
16. Ayala-Gómez F, Daróczy B, Benczúr A. Global citation recommendation using knowledge graphs. *J Intell Fuzzy Syst*. 2018;34(5):3089-3100.
17. Kim Y. Convolutional neural networks for sentence classification. Paper presented at: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar; 2014:1746-1751.
18. Wang H, Zhao M, Xie X. Knowledge graph convolutional networks for recommender systems. Paper presented at: Proceedings of the 2019 World Wide Web Conference on World Wide Web. San Francisco, CA; 2019:3307-3113.
19. Wang H, Zhang F, Wang J. Exploring high-order user preference on the knowledge graph for recommender systems. *ACM Trans Inf Syst*. 2019;37(3):1-26.

20. Wang X, He X, Cao Y. KGAT: knowledge graph attention network for recommendation. Paper presented at: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage, AK; 2019:950-958.
21. Sha X, Sun Z, Zhang J. Attentive knowledge graph embedding for personalized recommendation; 2019. arXiv preprint, arXiv:1910.08288.
22. Yu X, Ren X, Sun Y. Personalized entity recommendation. Proceedings of the 7th ACM International Conference on Web Search and Data Mining; 2014:283-292; ACM Press, New York, NY.
23. Shi C, Zhang Z, Luo P. Semantic path based personalized recommendation on weighted heterogeneous information networks. Paper presented at: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management; 2015:453-462; New York, ACM Press.
24. Luo C, Pang W, Wang Z: Hete-CF: social-based collaborative filtering recommendation using heterogeneous relations. Proceedings of 2014 IEEE International Conference on Data Mining; 2014:917-922; IEEE Press, Washington DC.
25. Shi C, Hu B. Heterogeneous information network embedding for recommendation. *IEEE Trans Knowl Data Eng*. 2019;31(2):357-370.
26. Ma W, Zhang M, Cao Y. Jointly learning explainable rules for recommendation with knowledge graph. Paper presented at: Proceedings of WWW'19; 2019:1-10; ACM Press, New York, NY.
27. Wu X, Chen Q, Liu H. Collaborative filtering recommendation algorithm based on representation learning of knowledge graph. *Comput Eng*. 2018;44(2):226-232.
28. Zhang F, Yuan N, Lian D. Collaborative knowledge base embedding for recommender systems. Paper presented at: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016:353-362; ACM Press, New York, NY.
29. Xin X, He X, Zhang Y. Relational collaborative filtering: modeling multiple item relations for recommendation [EB/OL]; October 14, 2019. https://arxiv.org/abs/1904.12796.
30. Wang C, David M. Collaborative topic modeling for recommending scientific articles. Paper presented at: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, CA; 2011:448-456.
31. Wang H, Zhang F, Zhang M. Knowledge graph convolutional networks for recommender systems with label smoothness regularization. Paper presented at: Proceedings of KDD'19; 2019:1-7; ACM Press, New York, NY.
32. Wang H, Zhang F, Zhao M. Multi-task feature learning for knowledge graph enhanced recommendation. Paper presented at: Proceedings of WWW'19; 2019:1-8; ACM Press, New York, NY.
33. Hu L, Li C, Shi C. Graph neural news recommendation with long-term and short-term interest modeling. *Inf Process Manag*. 2020;57(2):102-142.
34. He X, Liao L, Zhang H. Neural collaborative filtering. Paper presented at: Proceedings of the 26th International Conference on World Wide Web. Perth, Australia; 2017:173-182.
35. Hsieh C, Yang L, Cui Y. Collaborative metric learning. Paper presented at: Proceedings of the 26th International Conference on World Wide Web. Perth, Australia; 2017:193-201.
36. Bordes A, Usunier N, Garcia D. Translating embeddings for modeling multi-relational data. Paper presented at: Proceedings of the 27th Annual Conference on Neural Information Processing Systems. Lake Tahoe, Nevada; 2013:2787-2795.

How to cite this article: Tang H, Liu B, Qian J. Content-based and knowledge graph-based paper recommendation: Exploring user preferences with the knowledge graphs for scientific paper recommendation. *Concurrency Computat Pract Exper*. 2021;33:e6227. https://doi.org/10.1002/cpe.6227