IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 February 2023, accepted 15 March 2023, date of publication 20 March 2023, date of current version 23 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3259234

RESEARCH ARTICLE

Scientific Documents Retrieval Based on Graph Convolutional Network and Hesitant Fuzzy Set

XIN LI[1,2,3], BINGJIE TIAN[4], AND XUEDONG TIAN[1,2,3]

[1]School of Cyber Security and Computer, Hebei University, Baoding 071002, China
[2]Institute of Intelligent Image and Document Information Processing, Hebei University, Baoding 071002, China
[3]Hebei Machine Vision Engineering Research Center, Hebei University, Baoding 071002, China
[4]International Education College, Hebei Finance University, Baoding 071051, China

Corresponding author: Xuedong Tian (xuedong_tian@126.com)

This work was supported by the Natural Science Foundation of Hebei Province of China under Grant F2019201329.

ABSTRACT Previous scientific literature retrieval methods, which are based on mathematical expression, ignore the literature attributes and the association between the literature, and the retrieval accuracy was affected. In this study, literature retrieval model based on Graph Convolutional Network (GCN) is proposed. By extracting document attributes from a structured document dataset, an Attribute Relation Graph (ARG) is constructed. Using GCN to capture the dependencies among literature nodes and generate literature representations by information aggregation to realize graph-based literature modeling; Introducing the advantages of Hesitant Fuzzy Set (HFS) theory in multi-attribute decision-making to realize the similarity evaluation between mathematical query expressions and mathematical retrieval result expressions. Finally, the similarity between literature features and mathematical expressions is integrated to obtain the ordered output of scientific literature retrieval results. Experiments were conducted on the arXiv public dataset, and the average precision of the top 10 retrieval results was 0.892, and the average NDCG value of the top 10 rankings was 0.875.

INDEX TERMS Scientific document retrieval, GCN (graph convolutional network), mathematical expressions, HFS (hesitant fuzzy set).

I. INTRODUCTION

Mathematical expressions are an important part of scientific literature, and the integration of mathematical expressions, text, and literature attribute features in scientific literature retrieval systems is an inevitable trend to achieve high-performance retrieval. However, the special structure and properties of mathematical expressions lead to conventional full-text retrieval methods that cannot make good use of mathematical expression information in the scientific literature [1], [2]. In addition, literature retrieval systems should fully explore and discover the association information implied between scientific documents to make the retrieval results more reasonable. Therefore, it is of great theoretical and usage value to study the model of scientific literature retrieval that can integrate the attributes.

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

Graph Convolutional Network (GCN) [3], [4] can work directly on graphs and take advantage of their structural information. The main idea of "Convolutional" was from images and then graphs. The general idea of GCN is to represent nodes in the graph as low-dimensional, real-valued, dense vector forms by defining various aggregation functions to aggregate the central nodes and nodes' neighbors. Using GCN to learn the feature representation of documents and apply it to scientific literature retrieval. It can directly represent the attributes of documents and the association relationships among different documents, and provide a richer corpus for retrieval; therefore, the use of graph data structure for literature retrieval helps to improve the retrieval effect.

The similarity of mathematical expressions may be evaluated as a set of different degrees among numerous evaluation methods. Therefore, the process of similarity calculation can be considered as multi-criteria decision-making. HFS [5] has been shown to be one of the effective means

to solve multi-criteria decision-making problems [6]. Define the mathematical expression multi-attribute evaluation index and membership function to calculate the hesitant fuzzy set. The distance measure of hesitant fuzzy sets [7] differs from the traditional standard Euclidean distance by requiring the hesitant fuzzy elements to be sorted first. It focuses on more relevant membership and mitigates the negative impact of irrelevant membership on distance. A more objective description of the distance between hesitant fuzzy sets makes the calculation of expression similarity values more reasonable.

In summary, a scientific literature retrieval method based on GCN and HFS is proposed. Uses GCN for graph-based modeling of scientific literature retrieval and fully considers the important constituent attributes of documents and the association information between different documents. HFS theory is used in expression similarity calculation. Taking advantage of the simplicity and effectiveness of HFS theory, the similarity decision problem is solved by a multi-value hesitation method from multi-attribute evaluation. The contributions of this paper are summarized as follows:

1) Constructing an Attribute Relation Graph. It can correlate the titles, authors, keywords, categories, publications, and citations of documents and reveal the common attributes and differences among documents.
2) Proposed a GCN-based method to calculate the relevance score of the documents. GCN is applied to ARG to enrich node features by aggregating neighbor information through different aggregation functions to learn a better embedding representation for the documents.
3) Evaluate the similarity of mathematical expressions from multiple aspects using HFS. By extracting subexpressions and introducing HFS to calculate the expression similarity, the effect of expressions with multivariate forms is avoided.

II. RELATED WORK

A. SCIENTIFIC LITERATURE RETRIEVAL

Traditional document-based retrieval methods such as the probabilistic model [8], the boolean model [9], the topic model [10], and the vector space model [11]. Also, there are many retrieval methods based on neural network methods, Dense Passage Retrieval (DPR) [12], kernel-based neural model for document ranking (K-NRM) [13], etc. ColBERT [14] introduces a late interaction system to obtain fine-grained similarity for queries and documents. The ability to pre-compute document representations offline speeds up queries.

However, full-text matching-based retrieval methods ignore the semantic information between content entities; some related studies have proposed graph model-based retrieval methods. ESR [15] uses knowledge graph embedding techniques for semantic connectivity for academic search problems to improve the ability to solve hard queries. KESM [16] proposed a kernel entity salience model to realize better text retrieval by combining article context and entity knowledge representation to rank entity and article pairs. JointSem [17] addressed the semantic similarity problem in text ranking by extracting features of entity-entity links and adding them to the model, and the ranking was improved. Zhang et al. [18] discussed graph definition, graph construction, graph similarity calculation based on maximum common subgraph, and document scoring in graph-based document retrieval. Wise et al. [19] constructed knowledge graphs for literature of Covid-19, combining semantic information with topological information for document retrieval.

When the above studies were applied to scientific literature retrieval, document-based retrieval did not make use of the important components of literature, such as title, author, keywords, and other information, and ignored the correlation between different documents. Meanwhile, mathematical expressions have complex two-dimensional structures and variable syntactic semantic expressions, the use of mathematical information for scientific literature retrieval still needs further research.

B. GRAPH CONVOLUTIONAL NETWORK

Graph neural network (GNN) has been increasingly used in many fields, such as social networks, knowledge graphs, and recommender systems. The concept of GNN was first proposed by Gori et al. [20], and the theoretical basis for graph neural networks was further elaborated by Scarselli [21]. Cui et al. [22] constructed concept graphs in documents and studied structure-oriented complex GNNs and semantic-oriented graph functions for document retrieval. Yu et al. [23] used session graphs of users accessing documents to construct text recommendation models based on GNNs and attention mechanisms.

GCN is a type of GNN that uses a convolutional operator for information aggregation. The idea of ''convolution'' came from images and was later introduced into graphs. Among the applications of GCN, PH-GCN [24] uses GCN to learn the context of layered graphs of person images for person re-identification. Yu et al. [25] use GCN for graph-based representation of text modeling for coupled feature learning in cross-modal information retrieval. GraphSAGE [26] uses node feature information to efficiently generate embeddings for node data in graphs for various downstream tasks.

GCN can capture the information propagated between nodes in a graph and represent the nodes as low-dimensional dense vectors by defining various aggregation functions to aggregate the central nodes and the nodes' neighbors, effectively extracting spatial features for machine learning.

C. HESITANT FUZZY SET

When performing some decision-making tasks, the decision value is often not a particular exact result but a set of values obtained from different evaluation methods. Zadeh [27] proposed Fuzzy Set (FS) theory to solve the problems associated with fuzzy, subjective, and imprecise judgments. It allows the membership degree of an element to a fuzzy set to be

expressed as any value in [0,1]. For example, the membership of different evaluation elements x_1, x_2, x_3 to A is u_1, u_2, u_3, which are all single values of [0, 1]. But many times, the process of defining the membership of an element to a set is indecisive between several possible membership values. For example, two experts discuss the affiliation of element x_1 belonging to A. One expert gives the evaluation value of 0.6 and the other gives the evaluation value of 0.8, so the membership hesitates between 0.6 and 0.8. Torra [5] proposed HFS based on FS, where the membership of an object to a fuzzy set is given in the form of a set of several possible values. The membership of element x_1 to A is denoted by $\{0.6, 0.8\}$. Therefore, HFS is more objective and effective than FS in describing the hesitant attitudes that appear in the evaluation process.

HFS is a powerful tool to solve problems involving many uncertainties, such as the evaluation and selection of ideal suppliers [28] and fuzzy decision framework for the selection of drugs for the treatment of coronavirus disease [29]. *Rani* et al. [30] developed an integrated MCDM framework to handle sustainable supplier selection and confer various outlets of HFSs and information measures in the context of HFSs. Ahn et al. [31] evaluated and ranked sustainable suppliers using the hesitant fuzzy stepwise weight assessment ratio analysis complex proportional assessment. Tian and Wang [32] computed the affiliation of multiple attributes on the FDS structure of mathematical expressions to obtain the similarity between expressions. Zhu and Xu [33] computed different fuzzy values by different affiliation functions in scientific literature retrieval.

The similarity of mathematical expressions can be evaluated in terms of length, level, and content, and the membership between different evaluation elements has ambiguity. Meanwhile, the membership of expressions hesitates among the possible values of multiple sub-expressions. Therefore, how to reasonably apply HFS to mathematical expression similarity calculation needs further study.

III. METHOD

The model of scientific document retrieval based on GCN and HFS is shown in Fig.1.

First, construct ARG by extracting attributes from structured literature. For the nodes in the graph, a pre-trained model BERT [34] is used to obtain the embedding representation. The multi-layer GCN aggregates information about the nodes and neighboring nodes to get a feature representation of the document. For the input query text, use BERT to extract its features and calculate the cosine similarity with the document features to obtain the relevance score.

Next, in the mathematical expression module, an expression in MathML format is extracted, its symbolic layout tree is constructed, and then sub-expressions are extracted from this tree. The multi-attribute evaluation metrics and the corresponding membership functions of the sub-expressions between length, level, content, and normalized content are defined to obtain the HFS of the sub-expressions. Finally,

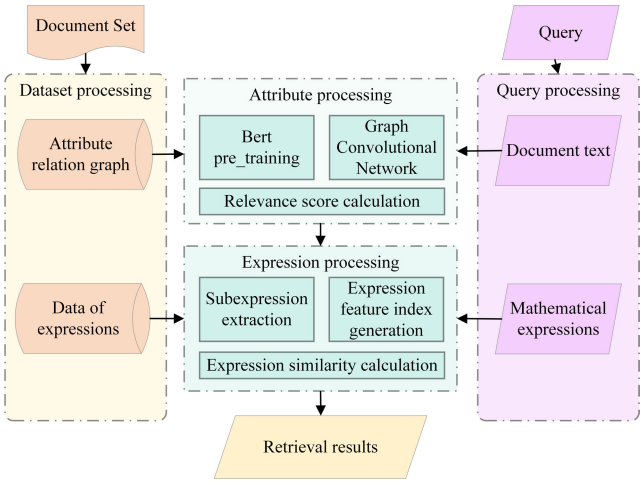

FIGURE 1. Flow chart of scientific document retrieval based on GCN and HFS.

the similarity of the expressions is obtained by the distance function.

Finally, the similarity of literature features and mathematical expressions are integrated to obtain an ordered output of scientific literature retrieval results.

A. DOCUMENT ATTRIBUTE RELATION GRAPH CONSTRUCTION

In this study, a graph is a topological graph in which corresponding relationships connect vertices and vertices. The graph structure can directly express the relationship between entities and query the associated data very efficiently while discovering the commonalities and differences between different literature. To improve the retrieval accuracy, the ARG is constructed.

ARG was constructed for the arXiv public dataset [35]. The dataset details are noted in section IV-A. The documents are stored in HTML format, which is a structured document with a fixed data format. The document attributes in the dataset are extracted using inductive rules, such as regular expressions and repeated patterns, using similar elements of web pages to extract data. Then a parser is written based on Python to parse and normalize the data and extract the key attributes.

The attributes related to each document are extracted from the dataset, including title, authors, keywords, publications (journals, magazines, conferences), categories, and references. The extracted attributes are represented as nodes. Among them, the title is the main node, and the other attribute nodes are the information nodes of the document.

Next is the construction of relationships, where a relationship defines a specific association that exists between two nodes. In ARG, it describes edges that connect two attribute nodes. ''WRITE'' is the relationship between title and author. ''EXIST'' is the relationship between the title and the keyword. ''PUBLISH'' is the relationship between the title and the publication, and ''CITE'' is the relationship between the title and the citation. ''BELONG'' is the relationship between

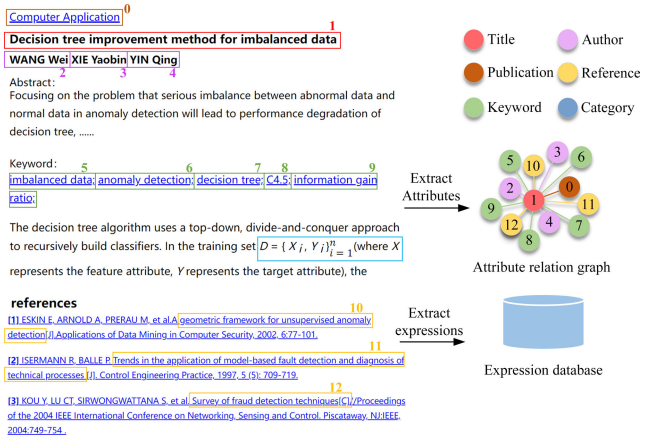

FIGURE 2. Document data extraction process. The document attribute information is extracted first, and then the mathematical expressions are extracted. This paper is from [36].

title and category. Using the extracted data, the documentary attribute relationship graph is constructed. An example of the construction is shown in Fig.2.

The ARG is constructed as a pre-processing step of the retrieval system. Its storage uses the graph database Neo4j. Neo4j is the world's leading open-source graph database [37], which stores data in the form of nodes and relations. Cypher is a declarative query language for Neo4j with a simple and powerful syntax. Data operations for retrieving relational graphs are performed using the Cypher language. The Neo4j Browser is a visualization designed for Neo4j. It displays entities and relationships of data stored in the database.

B. USING GRAPH CONVOLUTIONAL NETWORK TO LEARN THE DOCUMENT FEATURE REPRESENTATION

1) NOTATIONS

The ARG is represented as a global graph $G = \{V, E\}$, where V is the set of nodes in the graph, $v_i \in V$ is a node in the graph. E is the set of edges in the graph, $e_{i,j} \in E$ is an edge in the graph connecting nodes v_i and v_j. i is the number of the node and $i = 1, 2, \ldots, n$, where n is the number of all nodes in the graph. Each document in the dataset can be modeled as a partial graph $G_d = \{V_d, E_d\}$, where V_d is the set of nodes in this document. E_d is the set of edges between the nodes.

2) PRE-TRAINING

The BERT (Bidirectional Encoder Representations from Transformer) [34] model will be used as a pre-training model. BERT is trained based on a large untagged corpus, which can better extract contextual information and obtain more accurate semantic features of the text and solve the problem of multiple meanings of a word. For the node in ARG, its initial feature vector is obtained by BERT. Regarding a node v_i in the graph, after inputting it into the BERT model, its characteristic node representation a_i is obtained. $a_i \in \mathbb{R}^d$ and the node uses a d-dimensional embedding representation. The feature matrix $A \in \mathbb{R}^{n \times d}$ of the whole graph can be expressed as $A = [a_1, a_2, \ldots, a_n]^T$.

3) USING GRAPH CONVOLUTIONAL NETWORK TO LEARN THE DOCUMENT FEATURE REPRESENTATION

The flowchart for learning document feature representation using GCN is shown in Fig.3. A node v_i in the graph aggregates its own feature a_i with its neighboring features $a_j \left(v_j \in N\left(v_i\right)\right)$ to generate a new representation h_i. $N(v_i)$ is used to get the neighbor nodes of v_i. Define the aggregation operation as

$$h_i^{k+1} = AGGREGATE\left(h_i^k, h_j^k, \forall v_j \in N\left(v_i\right)\right) \tag{1}$$

where h_i^k denotes the feature of node v_i in layer k network, h_j^k denotes the feature of node v_j in layer k network. *AGGREGATE* () function is a pooling function that takes the target node and all its neighboring nodes as input.

Three aggregation functions for ARG are discussed from a semantic perspective, and the effectiveness of GCN in facilitating literature retrieval is discussed by comparing the retrieval results of different aggregation functions.

Me-Aggregate: The weighted features of neighbor nodes are averaged, while adding information about its own features.

$$h_i^{k+1} = h_i^k + \frac{1}{|N\left(v_i\right)|} \sum\nolimits_{v_j \in N(v_i)} w_{ij}^k h_j^k \tag{2}$$

where $|N\left(v_i\right)|$ is the number of neighboring nodes of node v_i. w_{ij}^k is the weight.

S-Aggregate: Use the weighted sum of all neighbors as the aggregation function, with information about its own features added.

$$h_i^{k+1} = h_i^k + \sum\nolimits_{v_j \in N(v_i)} w_{ij}^k h_j^k \tag{3}$$

Ma-Aggregate: Taking the maximum value of weighted features of all neighboring nodes for aggregation, and it adds its own feature information.

$$h_i^{k+1} = h_i^k + \max_{v_j \in N(v_i)} w_{ij}^k h_j^k \tag{4}$$

When doing the $k+1$-th aggregation for node v_i, the features of its neighboring nodes are first aggregated using the *AGGREGATE* () function, and then the embeddings obtained from the k-th aggregation are summed. Define the weighting function as

$$w_{ij}^k = \frac{\exp(w^{l^T}[h_i^k || h_j^k])}{\sum_{v_j \in N(v_i)} \exp(w^{l^T}[h_i^k || h_j^k])} \tag{5}$$

where $\bullet^T$ denotes the transpose operation, $||$ denotes stitching the two features together, w^l is the parameter to be learned, and $w^l \in \mathbb{R}^{2d}$. The weights can highlight important node

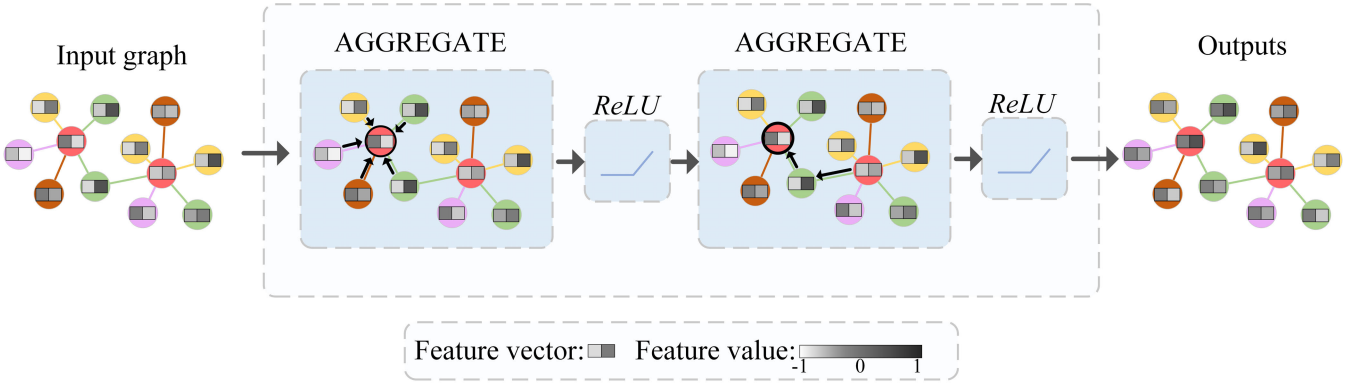

FIGURE 3. Flowchart for learning document feature representation using GCN.

features and weaken the features of nodes with lesser roles in the aggregation process. Use z_i to denote the final output representation after the k-layer GCN as

$$z_i = \sum_{k=0}^{K} ReLU\left(h_i^k\right) \tag{6}$$

The activation function ($ReLU$) is used after each layer of convolution to increase the nonlinearity. Retain useful features and remove irrelevant features to improve GCN training performance. In a multilayer GCN, the aggregation process fuses the local structure information of the nodes. In the first GCN layer, each node contains the feature information of its direct neighbors, and when the GCN enters the second layer, it is able to aggregate the features of its neighbors' neighboring nodes, so that the information involved in the documented feature representation will be more adequate. The more GCN layers, the more node information will be involved in the operation. As the number of GCN layers increases, the potential association information between attributes is mined more fully.

In order for the GCN model to learn useful document representations, a retrieval-based loss function is applied to the output. Given the query Q_T, the documents relevant to the query denoted as G_d and the irrelevant documents denoted as G_n. A common loss function in retrieval tasks, triplet loss, is used, which is represented as

$$L\left(Q_T, G_d, G_n\right) = \max\left(0, D\left(Q_T, G_d\right) - D\left(Q_T, G_n\right) + m\right) \tag{7}$$

where $D\left(G_d|Q_T\right) = \frac{z_{G_d} \cdot z_{Q_T}}{|z_{G_d}||z_{Q_T}|}$. The loss is calculated for the obtained embedding using a loss function, and the gradient descent algorithm is applied to back-propagate the optimized parameters w^l. After GCN aggregation, the Title node contains all the attributes of the document. z_{G_d} is the final representation of G_d.The main node of G_d aggregates the information of its neighbor nodes to get the complete information of the document. $z_{G_d} = z_{G_d\_main}$, where $z_{G_d\_main}$ denotes the main node feature representation of G_d after entering the K-layer GCN. For the test query information Q_T,

Algorithm 1 Learn the Node Representation by GCN

Input: graph $G(V, E)$, level K
Output: node representations

$h_i^{(0)} = a_i, \forall_{v_i} \in G$
for $k = K$ to 1
  for $i \in G$
    $w_{ij}^{(k)} = \frac{\exp(w^{lT}[h_i^{(k-1)}||h_j^{(k-1)}])}{\sum_{v_i \in N^{(k)}(v_i)} \exp(w^{lT}[h_i^{(k-1)}||h_j^{(k-1)}])}$
    $h_i^{(k)} = AGGREGATE(h_i^{k-1}, h_j^{k-1}, \forall j \in N^{(k)}(v_i))$
    $h_i^{(k)} = ReLU(h_i^{(k)})$
  end for
  Update w^l
end for
$z_i = h_i^{(K)}, \forall_{v_i} \in G$

preprocess it using Bert to get its embedding representation a_{Q_T}. The triplet can close the distance between positive samples and push away the distance between negative samples. For the input query Q_T, rank the documents according to the relevance score $D\left(G_d|Q_T\right)$. The Algorithm 1 shown the process of using graph convolutional network to learn the node feature representation.

The example of using GCN learning literature feature representation is shown in Fig.4. For each node in the sub graph, the node content is first input into the BERT model to obtain the embedding $a_0, a_1, \ldots, a_{12}$. In the first feature aggregation, the directly adjacent neighbors are aggregated. In the node representation part of Fig.4, node v_0 aggregates the features a_1, a_2 of v_1, v_2 and sums them with its own features to obtain the first aggregated result h_0^1. After the activation function, the second layer of neighbor nodes are aggregated again to obtain h_0^2. After the multilayer GCN process, the final node representation z_0 is obtained. To simplify the amount of relevance score calculation, all node features are sampled and the main node features are retained, and then the relevance score is obtained with the query by cosine distance.

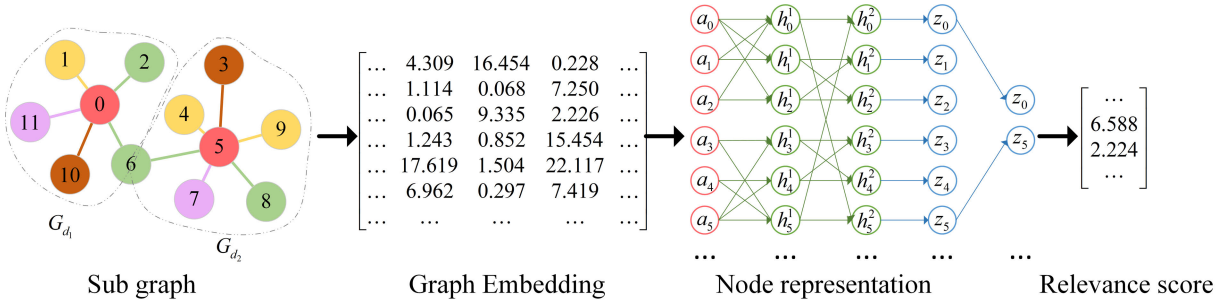

FIGURE 4. Example of GCN-based literature feature learning.

C. SIMILARITY CALCULATION OF MATHEMATICAL EXPRESSIONS BASED ON HFS

The similarity of multiple attributes of a mathematical expression is evaluated comprehensively using HFS. The hesitation fuzzy features of the mathematical expression are formed by constructing a symbolic layout tree of the mathematical expression, extracting sub-expressions, and defining multi-attribute evaluation indexes and the corresponding affiliation functions.

1) HESITANT FUZZY SET

Let the set of fixed attributes $X = \{x_1, x_2, \ldots, x_n\}$, the hesitant fuzzy set [5], [26] on the set of attributes X can be expressed as

$$E = \{\langle x, h_E(x) | x \in X \rangle\} \tag{8}$$

where $h_E(x)$ denotes the hesitant fuzzy element. Each hesitant fuzzy element can contain one or more evaluation values in the range of [0, 1]. All the evaluation values are arranged in descending order, where the element in the j-th position is noted as $h_E^{\sigma(j)}(x)$.

Xu and Xia [7] proposed a distance measure for hesitant fuzzy sets. Based on the hesitant standard Hamming distance and Euclidean distance formula, the generalized hesitant normalized distance calculated as

$$d_{ghn} = (M, N) = \left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{l_{x_i}}\sum_{j=1}^{l_{x_i}}\left|h_M^{\sigma(j)}(x_i) - h_N^{\sigma(j)}(x_i)\right|^{\lambda}\right)\right]^{\frac{1}{\lambda}} \tag{9}$$

where M and N are two hesitant fuzzy set samples, l_{x_i} is the number of evaluation values. $\lambda = 1$ for the hesitant standard Hamming distance formula, $\lambda = 2$ for the hesitant standard Euclidean distance formula, and n is the number of hesitant fuzzy attribute sets. The smaller the distance, the greater the similarity between the two samples.

2) SUB-EXPRESSION

Mathematical expressions are stored in MathML format. According to the hierarchical progression of the MathML format, we construct the symbolic layout tree of the expressions.

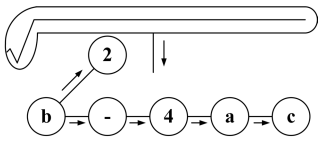

FIGURE 5. The symbol layout tree of $\sqrt{b^2 - 4ac}$.

TABLE 1. Sub expression of $\sqrt{b^2 - 4ac}$.

No.	1	2	3	4	5	6	7
sub expression	a	ac	$4ac$	$-4ac$	2	b^2-4ac	$\sqrt{b^2-4ac}$
length	1	2	3	4	1	6	7
level	1	1	1	1	1	2	3

Each level of the tree is analyzed to obtain sub-expressions. The symbolic layout tree of $\sqrt{b^2 - 4ac}$ is shown in Fig.5. The sub-expressions of $\sqrt{b^2 - 4ac}$ are shown in Table 1. At the same time, the lengths and levels of word expressions are listed in Table 1.

3) EXPRESSION SIMILARITY CALCULATION

Let Q be the query mathematical expression, E_i $(i = 1, 2, \ldots, n)$ be the set of mathematical expressions in the literature. *Qsub* and *Esub* are the sets of sub-expressions of Q and E. $Qsub\_i$ $(sub\_i = 1, 2, \ldots, Q\_s)$ is the i-th sub-expression of Q, and Q_s is the total number of sub-expressions of Q. $Esub\_j$ $(sub\_j = 1, 2, \ldots, E\_s)$ is the j-th sub-expression of E, and E_s is the total number of sub-expressions of E. HFS_Q is the hesitant fuzzy set of Q, and HFS_E is the hesitant fuzzy set of E.

According to the characteristics of mathematical expressions, the similarity of expression is evaluated in four aspects: length, level, content and normalized content. The hesitation fuzzy evaluation attributes set is constructed as $X = \{length, level, content, n\_content\}$, where *content* is the symbolic content of expression, *n_content* is the normalized content, which is used to avoid the influence of variable name, operand, operator change on the expression content, hesitant fuzzy element is defined as $h_E(x) = \{h_{length}, h_{level}, h_{content}, h_{n\_content}\}$. Different membership functions are defined in Table 2.

TABLE 2. Membership function for expression.

Evaluation attribute	Membership function	Description
Length	$h_{length}(Qsub\_i, Esub\_j) = \frac{\min(\text{length}(Qsub\_i), \text{length}(Esub\_j))}{\max(\text{length}(Qsub\_i), \text{length}(Esub\_j))}$	$\min(\ )$ for the minimum value function, $\max(\ )$ for the maximum value function. $\text{length}(\ )$ is used to calculate the length of the expression, that is, the number of symbols contained.
Level	$h_{level}(Qsub\_i, Esub\_j) = \frac{\min(\text{level}(Qsub\_i), \text{level}(Esub\_j))}{\max(\text{level}(Qsub\_i), \text{level}(Esub\_j))}$	$\text{level}(\ )$ is used to calculate the level of the expression. The relative position of the symbols for the main expression on or above will superimpose the level of the expression. For example, $\text{level}(b^2) = 2$ and $\text{level}(\sqrt{b^2}) = 3$.
Content	$h_{content}(Qsub\_i, Esub\_j) = \frac{Qsub\_i_{sy} \odot Esub\_j_{sy}}{\max(\text{length}(Qsub\_i), \text{length}(Esub\_j))}$	$Qsub\_i_{sy}$ and $Esub\_j_{sy}$ denote the symbols in $Qsub\_i$ and $Esub\_j$. $Qsub\_i_{sy} \odot Esub\_j_{sy}$ is used to count the number of identical symbols.
Normalized content	$h_{n\_content}(Qsub\_i, Esub\_j) = \frac{NOR(Qsub\_i_{sy}) \odot NOR(Esub\_j_{sy})}{\max(\text{length}(Qsub\_i), \text{length}(Esub\_j))}$ $NOR(sub) = Operator \xrightarrow{replace} S, Operand \xrightarrow{replace} I, Number \xrightarrow{replace} N$	$NOR(sub)$ is a sub-expression normalization method that replaces the operator in the sub-expression with S, the operand with I, and the number with N. For example, the expressions $a+2$ and $3+b$, after normalization, are represented as ISN and NSI, and both of them the value of $h_{n\_content}$ is 1. The normalization operation further solves the problem of renaming formula variables.

The membership function converts the relationship between the query and the mathematical expression into the corresponding hesitant fuzzy set. For the mathematical expression Q with itself all membership values are set to 1 to construct the ideal hesitant fuzzy set, which ideal value will be used as a measure of similarity between different samples. For example, for the query expression Q as b^2 and the expression E in the dataset as a, $Qsub = \{2, b^2\}$ and $Esub = \{a\}$, then HFS_E can be expressed as $\{\langle x_1, \{1, 0.5\}\rangle, \langle x_2, \{1, 0.5\}\rangle, \langle x_3, \{0, 0\}\rangle, \langle x_4, \{0, 0.5\}\rangle\}$ and HFS_Q as the ideal hesitation fuzzy set $\{\{1, 1\}, \{1, 1\}, \{1, 1\}, \{1, 1\}\}$.

Correspondingly, according to the hesitation fuzzy set distance measure calculation method, the hesitation fuzzy distance calculation formula for the mathematical expression Q and E can be derived as

$$dis\left(HFS_Q, HFS_E\right) = \left[\frac{1}{4}\sum_{i=1}^{4}\left(\frac{1}{l}\sum_{j=1}^{l}\left|h_{HFS_Q}^{\sigma(j)}(x_i) - h_{HFS_E}^{\sigma(j)}(x_i)\right|^{\lambda}\right)\right]^{\frac{1}{\lambda}} \quad (10)$$

where λ is the control parameter, $l = Q\_s \times E\_s$. This distance measure not only takes into account the similarity ambiguity brought by the four evaluated attributes of the expressions, but also resolves the hesitancy between the similarity of the sub-expressions of the expressions. The larger the distance the smaller the similarity, so the similarity is calculated by

$$SIM(Q, E) = 1 - dis\left(HFS_Q, HFS_E\right) \quad (11)$$

The algorithm for expressing similarity calculation is shown in Algorithm 2. First, the symbolic layout tree of the input mathematical expressions Q and E is constructed by *Analysis_SLT*(), and the set of sub-expressions is obtained by analyzing each layer of the tree. For each sub-expression, the corresponding fuzzy elements are obtained by the membership function of four different elements in Table 2, and the hesitant fuzzy set of the expression is obtained by integrating the fuzzy elements of all sub-expressions. At this time, the time complexity of the algorithm is $O(MN)$, where $M = E\_s$ and $N = Q\_s$. The space complexity of the algorithm is $O(1)$. Finally, the similarity between expressions is obtained by distance measure. Zhang [38] demonstrated that different λ values do not have an impact on the retrieval ranking results. For the convenience of calculation, $\lambda = 1$ is chosen, at which the distance function is the standard Hamming distance.

Algorithm 2 Mathematical Expression Similarity Calculation

Input: Q, E_i $(i = 1, 2, \ldots, n)$
Output: *simList* //A collection of expressions similar to expression Q

```
// Construct the symbolic layout tree and get the set of subexpressions
Qsub =Analysis_SLT(E)
Esub =Analysis_SLT(E)
for Qsub_i in Qsub:
  for Esub_j in Esub:
    // Calculating hesitant fuzzy elements
    h_E =[h_length (Qsub_i, Esub_j),
      h_level (Qsub_i, Esub_j),
      h_content (Qsub_i, Esub_j),
      h_n_content (Qsub_i, Esub_j)]
    HFS_E.add(Esub_j, h_E)
  end for
end for
// Ideal hesitant fuzzy set
HFS_Q =[{1,...,1},{1,...,1},{1,...,1},{1,...,1}]
dis(HFS_Q, HFS_E) = 1/4 Σ_{i=1}^{4} ( 1/|HFS_E| Σ_{j=1}^{|HFS_E|} |h^{σ(j)}_{HFS_Q}(x_i) − h^{σ(j)}_{HFS_E}(x_i)| )
simList= 1 − dis(HFS_Q, HFS_E)
RETURN simList
```

IV. EXPERIMENTAL PROCESS AND RESULT ANALYSIS

A. EXPERIMENTAL DATA

The arXiv provides open access to scholarly articles in many various fields of scientific research, encompassing many sub-disciplines from physics to computer science. It offers a free, open machine-readable dataset on Kaggle [35]: a repository of articles that contains relevant data such as article title, author, category, abstract, full-text content, etc. This dataset was used for the study. One of the documents from 2022 was selected, containing 132,920 articles and more than one million mathematical expressions. However, since it contains only English documents, Chinese documents [32] were introduced to extend the dataset. The source data of the literature dataset is stored in HTML format.

B. EVALUATION INDICATORS

The experiment uses the MAP and NDCG evaluation index to evaluate the retrieval results.

NDCG (normalized discounted cumulative gain) [39] is a common measure of the quality of a set of search results. The NDCG method assesses the degree of relevance by scoring the retrieved documents. It assumes that highly relevant results have a greater impact on the final score and that the earlier they appear, the higher the score. The calculation formula is

$$DCG_P = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \tag{12}$$

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{13}$$

where rel_i is the relevance score, and p means the top p result.

The AP calculation method is

$$AP = \frac{1}{r} \sum_{i}^{r} \frac{i}{pos\,(i)} \tag{14}$$

where r is the serial number of the relevant document, $pos\,(i)$ is the position of the document, and r is the total number of relevant documents. MAP is the mean of the AP scores. MAP is a common metric used in information retrieval. MAP is the mean of the average accuracy of multiple queries, which can reflect the performance of retrieval as a whole.

C. ATTRIBUTE RELATION GRAPH

The node statistics of ARG are shown in Table 3. There are 2,170,140 nodes in the graph, of which 142,923 are title nodes. The number of publication nodes is 854, and the number of keyword nodes is 5,452. The number of publication nodes and keyword nodes is low because some documents in the dataset are missing relevant data. There are 1,652,024 citation nodes in total. On average, one article refers to 12 other articles.

In the ARG, a title node and all its directly connected nodes form a scientific document. The minimum the number of nodes directly connected to each title is 2 and the maximum value is 197. Each title node has 20 directly neighboring nodes on average. For example, two documents, a and b, are published in the same publication. Then the title node of document a is connected to the publication node in which it is published. At the same time, the title node of document b is also connected to this publication node.

TABLE 3. Attribute relation graph node statistics.

Node label	significance	Number	Total
Title	Title of the document	142,923	2,170,140
Author	Author of the document	365,975	
Publication	Name of the publication to which the document published	854	
Keyword	Keywords included in the document	5,452	
Category	Category of the literature	2,912	
Reference	Reference names appearing in the document	1,652,024	

D. DOCUMENT RETRIEVAL BASED ON GCN

1) PARAMETER SETTING

By analyzing the nodes in the ARG graph, the number of title node that can propagate information with two steps is higher, a two-layer GCN network is used. At the same time, the two-layer GCN network can avoid the over-smoothing problem caused by too many layers. The parameters of the pre-training model BERT are provided publicly by Google, and BERT transforms the node text in the graph into a 768-bit feature vector. The dimension of the embedding vector of the response in GCN is 768.

2) RESULT ANALYSIS

Ten input query texts were selected, as shown in Table 4, to count the performance of the AP values of the retrieved results and the DCG of the ranked results of the documents under different aggregation functions. The comparison experiments were conducted using BM25 [8] and GraphSAGE [26]. BM25 is one of the classical algorithms for calculating query-document similarity scores, which uses the traditional method to calculate the relevance of each word and article, and then sums these scores to obtain the text similarity results. GraphSAGE is a graph neural network algorithm that uses the nodes' neighborhood information and fuses the neighbor information through a multilayer aggregation function.

Table 5 shows the MAP and NDCG values under different methods, which are the average of 10 query inputs. It can be found that the GCN-based document retrieval method outperforms the BM25 method. BM25 searches in full-text, but the GCN-based method aggregates document attributes and also aggregates different document information associated with them. GraphSAGE is less effective than Me-Aggregate. The possible reason is it aggregates the neighbor node information with the same weight. When using the GCN model, the selection of Me-Aggregate as the aggregation function performs better in comparison. This is because in a local graph G_d, the title node contains more information representing the document. The feature vector of the document can be better obtained by adding the title node feature after the neighbor

TABLE 4. 10 query expressions and query text.

No.	Query expression	Query text
1	$f(x)=\frac{f(x_0)}{0!}+\frac{f'(x_0)}{1!}(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\ldots+\frac{f(n)(x_0)}{n!}+Rn(x)$	Talking about the application of Taylor's formula in the finale of college entrance examination mathematics
2	$\sqrt{ab}\le\frac{a+b}{2}$	Knowledge and understanding of mathematical operations
3	$E=\frac{1}{2}mv^2$	Simulation study on the effect of impact stress on the performance of electrical connectors
4	$f(x)$	Intuition helps to think and solve difficult problems——Analysis and enlightenment of the 2017 New Curriculum Standard I Derivative Final Question
5	$n=x+y+a$	BPNN-based maneuver recognition method
6	$q=-\frac{Q}{St}$	Test and Analysis of Thermophysical Properties of Aeolian Sand Modified Soil
7	$W=\frac{U^2}{R}$	Research on Mathematical Modeling of Electrode Boiler Based on Dynamic Neural Network
8	$it+1$	Distributed Intrusion Attack Detection System Based on Artificial Bee Colony Algorithm
9	$\lim_{t\to\infty}e_i(t)=0$	Research on Robust Adaptive Synchronous Control Algorithm for Doubly Uncertain Fractional-Order Chaotic Systems
10	$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$	Exquisite Concepts Get to the Bottom of the Root——Take "A Unit Test Question" as an Example

TABLE 5. MAP and NDCG values under different methods.

method	MAP		NDCG	
	K=10	K=20	K=10	K=20
BM25	0.563	0.524	0.537	0.518
GraphSAGE	0.737	0.729	0.692	0.675
S-Aggregate	0.719	0.673	0.715	0.663
Me-Aggregate	**0.748**	**0.723**	**0.769**	**0.744**
Ma-Aggregate	0.724	0.698	0.681	0.677

node feature is averaged. And Ma-Aggregate performs poorly because it can only perform graph convolution from a single aspect. It tends not to represent the embedding of the documents better. Therefore, in the subsequent literature retrieval experiments, mean will be used as the aggregation function.

The NDCG(K=10) values for the 10 query texts at BM25, GraphSAGE and, the GCN model based on different aggregation functions are shown in Fig.6. GraphSAGE's NDCG@10 has a mean of 0.692. The mean of NDCG@10 for the Me-Aggregate is 0.769 They all perform node feature aggregation effectively. However, under experimental conditions, Me-Aggregate performs better because it uses the average value between the primary and information nodes for aggregation in response to the structural characteristics of the ARG graph. The different aggregation functions work best for the 8-th input of the query. This is because there are more articles in the dataset that are similar to query 8. Query 8 belongs to the category of engineering technology articles.

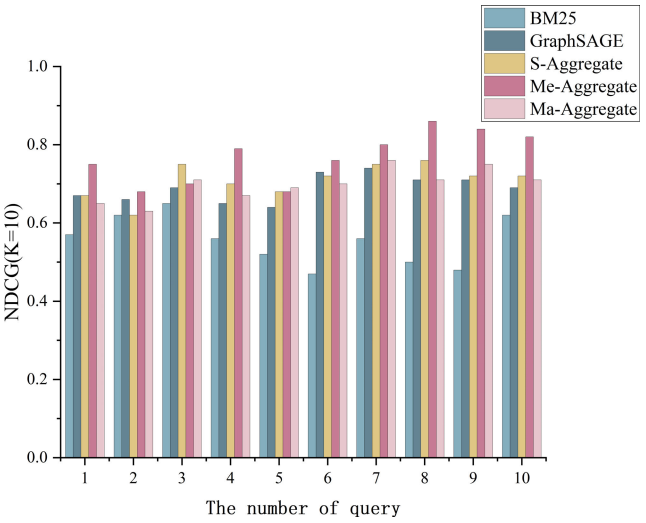

FIGURE 6. The value of NDCG(K=10) under different methods.

E. SIMILARITY CALCULATION OF MATHEMATICAL EXPRESSIONS

In order to verify the soundness of the HFS-based similarity assessment method for mathematical expressions, a query expression was given and retrieved to obtain TOP-5 results to analyze the relationship between similarities. For the query expression $\sqrt{b^2-4ac}$, the retrieved results TOP-5 are shown in Table 6.

The TOP-5 results show that HFS focuses more on sub-expression matching of formulas. If the formula exists in the same sub-expression of the query formula, the relevance score increases. Statistical analysis is performed for the 10 query expressions containing different mathematical symbols in Table 4. The average values of the search results NDCG and MAP are shown in Table 7.

TABLE 6. Search results TOP-5 of $\sqrt{b^2-4ac}$.

TOP	Search result	The literature
1	$g=\sqrt{b^2-4ac}$	Periodic and localized waves in parabolic-law media with third- and fourth-order dispersions
2	$g=\sqrt{b^2-4ac}$	Propagation of periodic and solitary waves in a highly dispersive cubic-quintic medium with self-frequency shift and self-steepening nonlinearity
3	$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$	DIRECT, a low-cost system for high-speed, low-noise imaging of fluorescent biological samples
4	$x^{(+)}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$	Infrared scalar one-loop three point integrals in loop regularization
5	$r_+^2=\frac{-b+\sqrt{b^2-4ac}}{2a}$	Thermodynamics of the Reissner-Nordstr¨om-de Sitter Spacetime with Quintessence

TABLE 7. 10 query mathematical expressions retrieval results NDCG and MAP average value.

	MAP			NDCG		
	K=5	K=10	K=20	K=5	K=10	K=20
Average value	0.897	0.846	0.803	0.868	0.847	0.784

The data in Table 7 shows that MAP decreases as the value of K increases. The number of highly similar expressions from the set of is less than 20, which is one of the reasons for the lower MAP(K=20). It can be seen from table that the average accuracy of the expression retrieval system based on the mathematical expression similarity evaluation model is 85%, and the ranking accuracy of retrieval results is 83%. It shows that the HFS-based similarity analysis of mathematical expressions can obtain higher scores for mathematical expressions with higher similarity. This is because the method can calculate similarity from multiple dimensions while considering the content and structural features of mathematical expressions, and the retrieval results have high accuracy.

F. DOCUMENT SEARCH BASED ON GCN AND HFS

In this study, the focus is on the performance improvement of mathematical expression-based models for scientific literature retrieval. Therefore, two widely used and simple models, Tangent-cft [40] and SearchOnMath [41], are used as baselines. Tangent-cft is a mathematical expression embedding model that uses a depth-first search algorithm to convert the paths between symbols of an expression tree into a list of symbolic tuples and then uses fastText to retrieve the expressions by averaging. SearchOnMath is a mathematical formula retrieval tool that locates scientific documents by precisely matching mathematical expressions for "mathematical expressions - scientific documents" retrieval. ColBERT will also be used for comparison experiments as a representation-based text-matching method. The AP and NDCG values of the literature search results under different methods are shown in Table 8.

TABLE 8. Results of the document search under different methods.

method	MAP		NDCG	
	K=10	K=20	K=10	K=20
SearchOnMath	0.609	0.561	0.613	0.588
Tangent-cft	0.773	0.741	0.752	0.689
GraphSAGE	0.737	0.729	0.692	0.675
ColBERT	0.676	0.661	0.649	0.633
GCN	0.748	0.723	0.769	0.744
HFS	0.846	0.803	0.847	0.784
GCN+HFS	**0.892**	**0.831**	**0.875**	**0.854**

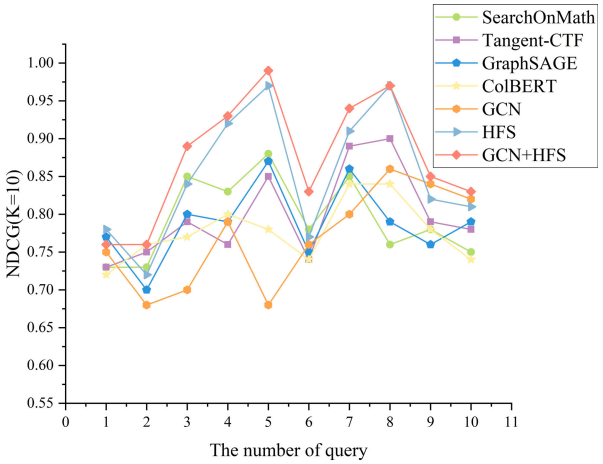

FIGURE 7. NDCG(K=10) under different search methods.

From Table 8 shows that SearchOnMath uses a single query condition and does not perform a joint search on the input text or mathematical expressions, so it does not perform as well as other models. Tangent-cft focuses on the mathematical expressions themselves and ignores information about the literature attributes. ColBERT models the content of the literature. It may be noisy and have a negative impact on the retrieval results. HFS-based retrieval of mathematical expressions works better because HFS can evaluate similarity from multiple attributes, and it can adapt to variable mathematical expressions. GCN does not work as well as HFS because it considers the similarity of the text content. But the best results are achieved by fusing GCN and HFS. This also proves that the introduction of GCN can learn the document representation well and can improve the retrieval accuracy.

Fig.7 shows the average of 10 query expressions NDCG(K=10) under different methods. SearchOnMath and Tangent-CTF both query by expressions, GraphSAGE, ColBERT, and GCN query based on text, but GraphSAGE and GCN both use graph structure information, while ColBERT queries by full-text semantic information. Among these methods GCN+HFS perform more prominently, it not only for mathematical expressions, but also incorporates information from graphs. It can be seen from the figure that query 5 performs best due to the simple structure of its mathematical expressions. Many similar mathematical expressions exist in the dataset. Moreover, it is more general in terms of the similarity of article attributes. As for query 2, it has the lowest

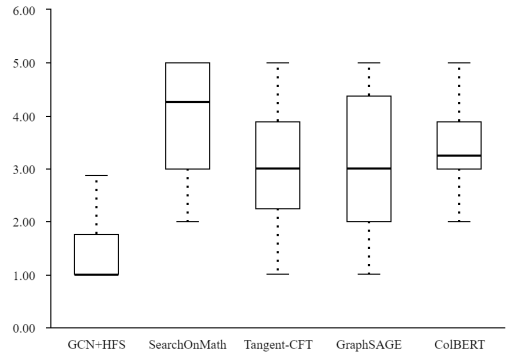

FIGURE 8. The Friedman test results under different methods.

value. The reason is that it lacks information about the author, keywords, and cited attributes of the document, so the result is low when calculating the similarity of the document. Comparative experimental data showed that enhanced document retrieval using mathematical expressions worked best. This is because mathematical expressions are very important in scientific document, and the presence of similar mathematical expressions may indicate that the document's content is also relevant.

The statistical analysis (Friedman test) is performed using the values of relevance score to find significant differences between the methods. As shown in the Fig.8, the abscissa axis represents the method and the ordinate axis represents the ranking value. The five methods showed significant differences ($p = 0.009 < 0.01$). It verifies that our proposed method has better overall retrieval performance.

The running times of the 10 queries are shown in Fig.9. It can be seen from the figure that the structural complexity of the expression has a great influence on the running time. The more complex the expression is, the longer the runtime is. Expression 1 has the most complex structure, so the overall running time is longer. Since the GraphSAGE, ColBERT, and GCN models do not use mathematical expressions as input, the retrieval time is not affected by mathematical expressions, and these models are also unable to perform mathematical expression-based literature retrieval. Retrieval based on GCN and HFS does not have a higher run time advantage because similarity is considered from both the attributes in the graph and the mathematical expressions, and some time is sacrificed for the purpose of higher retrieval accuracy. However, the method proposed in this paper has certain execution time advantages compared with the existing mathematical expression retrieval models SearchOnMath and Tangent-cft.

The running time used for 10,000 user queries is counted and the results are shown in Table 9. From the table, we can see that the average running time for the first 10 queries is 1.943 seconds and the average running time for the first 10,000 queries is 2.551 seconds. As the number of queries increases, the average query time is about 2.55 seconds. The main time factor for running queries is the complexity of the mathematical expressions. The fastest query time for a

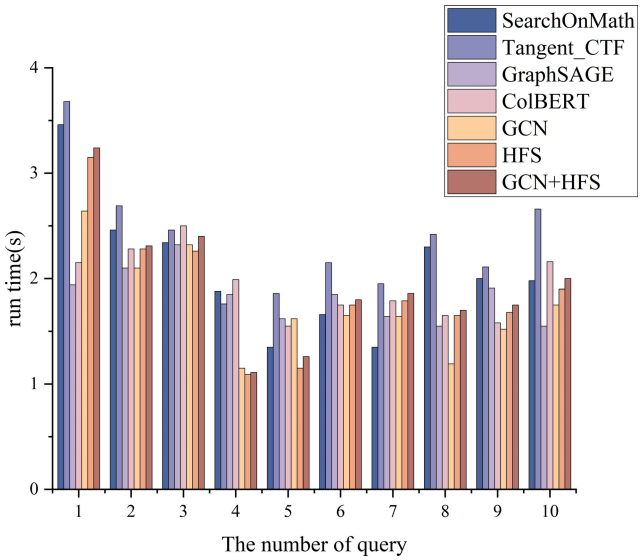

FIGURE 9. Run time under different search methods.

TABLE 9. Run time statistics.

Query number	10	100	1000	10000
Average run time(s)	1.943	2.385	2.549	2.551

simple structure is 0.467 seconds. The longest run time is 7.397 seconds.

As can be seen from the running time, the method proposed in this paper has certain limitations in execution time, with no major advantage in query time, and is influenced by the complexity of the expression structure. In future system optimization, methods such as referencing the cut module can be considered to improve the query efficiency.

In summary, literature retrieval based on GCN and HFS can effectively retrieve scientific literature related to expressions and literature. This is because the GCN network can aggregate the node information in the ARG graph, containing not only directly connected nodes but also the information of the neighbors. These features play a great role in retrieval. At the same time, ARG can effectively supplement the missing information of literature attributes in previous studies of mathematical expression retrieval, further improving the accuracy of literature retrieval and meeting more retrieval needs.

V. CONCLUSION

This paper proposed a scientific literature retrieval method based on GCN and HFS. The GCN fully explores the correlation between document attributes and the differences among different documents, and aggregates the information of neighboring nodes by means of aggregation functions. The aggregation of node features yields a more accurate representation of document embeddings. For the problem of similarity evaluation of mathematical expressions, the membership is calculated from four aspects of the length, level, content and

normalized content of the expressions, and the hesitant fuzzy set is constructed using the hesitancy of the similarity of sub-expressions, and the similarity is evaluated by the distance measure. It remedies the shortcomings of existing methods using a single measure in similarity evaluation. Experiments on the arXiv public dataset show that the average precision of the top 10 search results was 0.892 and the average NDCG value of the top 10 ranking results was 0.875. The results demonstrated the effectiveness of the search method, which exceeded existing mathematical expression-based scientific literature search methods in terms of retrieval accuracy.

The model proposed in this paper provides a graph-based solution idea for the literature retrieval task, replacing the traditional full-text retrieval model with a graph model. At the same time, the hesitant fuzzy property existing in the mathematical expression is found, and the HFS theory is effectively used for the downstream task. However, it still needs to be optimized. In the future research work, the model will be improved in two aspects:

1) Supplement the nodes in the ARG. The context of mathematical expressions is added to the ARG to improve the connection between the literature information and the expressions. The topics of the documents are further extracted so that the ARG graph can represent the content of the documents more comprehensively.
2) Optimize the calculation method for complex expression similarity. Introduce the complex mathematical expression slicing module to convert complex expressions into simple expressions and reduce the retrieval running time. Consider hesitant fuzzy weighted average and hesitant fuzzy weighted geometric operator to effectively integrate hesitant fuzzy information and make the expression similarity more reasonable.

REFERENCES

[1] P.-Y. Chien and P.-J. Cheng, "Semantic tagging of mathematical expressions," in *Proc. WWW*, May 2015, pp. 195–204, doi: 10.1145/2736277.2741108.
[2] M. Schubotz, A. Greiner-Petter, P. Scharpf, N. Meuschke, H. S. Cohl, and B. Gipp, "Improving the representation and conversion of mathematical formulae by considering their textual context," in *Proc. ACM/IEEE JCDL*, New York, NY, USA, May 2018, pp. 233–242, doi: 10.1145/3197026.3197058.
[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," Feb. 2017, *arXiv:1609.02907*.
[4] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.
[5] V. Torra, "Hesitant fuzzy sets," *Int. J. Intell. Syst.*, vol. 25, no. 6, pp. 529–539, Jun. 2010, doi: 10.1002/int.20418.
[6] J. Lan, R. Jin, Z. Zheng, and M. Hu, "Priority degrees for hesitant fuzzy sets: Application to multiple attribute decision making," *Oper. Res. Perspect.*, vol. 4, pp. 67–73, Jan. 2017, doi: 10.1016/j.orp.2017.05.001.
[7] Z. Xu and M. Xia, "Distance and similarity measures for hesitant fuzzy sets," *Inf. Sci.*, vol. 181, pp. 2128–2138, Jun. 2011, doi: 10.1016/j.ins.2011.01.028.
[8] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, Apr. 2009, doi: 10.1561/1500000019.
[9] H. Scells, G. Zuccon, and B. Koopman, "Automatic Boolean query refinement for systematic review literature search," in *Proc. WebConf*, New York, NY, USA, May 2019, pp. 1646–1656, doi: 10.1145/3308558.3313544.
[10] A. Sharaff, J. K. Dewangan, and D. S. Sisodia, "Prospecting the effect of topic modeling in information retrieval," *Int. J. Semantic Web Inf. Syst.*, vol. 17, no. 3, pp. 18–34, Jul. 2021, doi: 10.4018/IJSWIS.2021070102.
[11] M. Eminagaoglu, "A new similarity measure for vector space models in text classification and information retrieval," *J. Inf. Sci.*, vol. 48, no. 4, pp. 463–476, Aug. 2022, doi: 10.1177/0165551520968055.
[12] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," Sep. 2020, *arXiv:2004.04906*.
[13] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proc. SIGIR*, Aug. 2017, pp. 55–64, doi: 10.1145/3077136.3080809.
[14] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," Jun. 2020, *arXiv:2004.12832*.
[15] C. Xiong, R. Power, and J. Callan, "Explicit semantic ranking for academic search via knowledge graph embedding," in *Proc. WWW*, Perth, WA, Australia, Apr. 2017, pp. 1271–1279, doi: 10.1145/3038912.3052558.
[16] C. Xiong, Z. Liu, J. Callan, and T.-Y. Liu, "Towards better text understanding and retrieval through kernel entity salience modeling," in *Proc. SIGIR*, Ann Arbor, MI, USA, Jun. 2018, pp. 575–584, doi: 10.1145/3209978.3209982.
[17] C. Xiong, Z. Liu, J. Callan, and E. Hovy, "JointSem: Combining query entity linking and entity based document ranking," in *Proc. CIKM*, Nov. 2017, pp. 2391–2394.
[18] Z. Zhang, L. Wang, X. Xie, and H. Pan, "A graph based document retrieval method," in *Proc. CSCWD*, May 2018, pp. 426–432, doi: 10.1109/CSCWD.2018.8465295.
[19] C. Wise, V. N. Ioannidis, M. Romero Calvo, X. Song, G. Price, N. Kulkarni, R. Brand, P. Bhatia, and G. Karypis, "COVID-19 knowledge graph: Accelerating information retrieval and discovery for scientific literature," Jul. 2020, *arXiv:2007.12731*.
[20] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IJCNN*, vol. 2, Jul./Aug. 2005, pp. 729–734, doi: 10.1109/IJCNN.2005.1555942.
[21] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.
[22] H. Cui, J. Lu, Y. Ge, and C. Yang, "How can graph neural networks help document retrieval: A case study on CORD19 with concept map generation," in *Advances in Information Retrieval*. Stavanger, Norway: ECIR, 2022, pp. 75–83, doi: 10.1007/978-3-030-99739-7_9.
[23] J. Yu, C. Pan, Y. Li, and J. Wang, "An academic text recommendation method based on graph neural network," *Information*, vol. 12, no. 4, p. 172, Apr. 2021, doi: 10.3390/info12040172.
[24] B. Jiang, X. Wang, A. Zheng, J. Tang, and B. Luo, "PH-GCN: Person retrieval with part-based hierarchical graph convolutional network," *IEEE Trans. Multimedia*, vol. 24, pp. 3218–3228, 2022, doi: 10.1109/TMM.2021.3095789.
[25] J. Yu, Y. Lu, Z. Qin, Y. Liu, J. Tan, L. Guo, and W. Zhang, "Modeling text with graph convolutional network for cross-modal information retrieval," 2018, *arXiv:1802.00985*.
[26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2019, pp. 1–11.
[27] L. A. Zadeh, G. J. Klir, and B. Yuan, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers*. Singapore: World Scientific, 1996.
[28] R. Krishankumar, K. S. Ravichandran, K. K. Murthy, and A. B. Saeid, "A scientific decision-making framework for supplier outsourcing using hesitant fuzzy information," *Soft Comput.*, vol. 22, no. 22, pp. 7445–7461, Nov. 2018, doi: 10.1007/s00500-018-3346-z.
[29] A. R. Mishra, P. Rani, R. Krishankumar, K. S. Ravichandran, and S. Kar, "An extended fuzzy decision-making framework using hesitant fuzzy sets for the drug selection to treat the mild symptoms of coronavirus disease 2019 (COVID-19)," *Appl. Soft Comput.*, vol. 103, May 2021, Art. no. 107155, doi: 10.1016/j.asoc.2021.107155.
[30] P. Rani, A. R. Mishra, R. Krishankumar, A. Mardani, F. Cavallaro, K. S. Ravichandran, and K. Balasubramanian, "Hesitant fuzzy SWARA-complex proportional assessment approach for sustainable supplier selection (HF-SWARA-COPRAS)," *Symmetry*, vol. 12, no. 7, p. 1152, Jul. 2020, doi: 10.3390/sym12071152.

[31] Y. Ahn, S.-G. Lee, J. Shim, and J. Park, "Retrieval-augmented response generation for knowledge-grounded conversation in the wild," *IEEE Access*, vol. 10, pp. 131374–131385, 2022, doi: 10.1109/ACCESS.2022.3228964.
[32] X. Tian and J. Wang, "Retrieval of scientific documents based on HFS and BERT," *IEEE Access*, vol. 9, pp. 8708–8717, 2021, doi: 10.1109/ACCESS.2021.3049391.
[33] B. Zhu and Z. Xu, "Probability-hesitant fuzzy sets and the representation of preference relations," *Technol. Econ. Develop. Economy*, vol. 24, no. 3, pp. 1029–1040, May 2018, doi: 10.3846/20294913.2016.1266529.
[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Amsterdam, The Netherlands, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
[35] arXiv.org, "ArXiv dataset," Kaggle, Tech. Rep., 2023, doi: 10.34740/KAGGLE/DSV/4852963.
[36] W. Wang, Y. Xie, and Q. Yin, "Decision tree improvement method for imbalanced data," *J. Comput. Appl.*, vol. 39, no. 3, pp. 623–628, 2019.
[37] J. J. Miller, "Graph database applications and concepts with Neo4j," in *Proc. SAIC*, 2013, vol. 2324, no. 36, pp. 1–7.
[38] K. Zhang, "Research on ranking mathematical retrieval results based on hesitant fuzzy Set," Dept. Comput. Sci. Technol., HBU, Houston, TX, USA, Tech. Rep., 2017.
[39] K. Järvelin and J. Kekäläinen, "IR evaluation methods for retrieving highly relevant documents," *ACM SIGIR Forum*, vol. 51, no. 2, pp. 243–250, Aug. 2017, doi: 10.1145/3130348.3130374.
[40] B. Mansouri, S. Rohatgi, D. W. Oard, J. Wu, C. L. Giles, and R. Zanibbi, "Tangent-CFT: An embedding model for mathematical formulas," in *Proc. ICTIR*, Santa Clara, CA, USA, Sep. 2019, pp. 11–18, doi: 10.1145/3341981.3344235.
[41] R. M. Oliveira, F. B. Gonzaga, V. C. Barbosa, and G. B. Xexéo, "A distributed system for search on math based on the Microsoft BizSpark program," Nov. 2017, *arXiv:1711.04189*.

XIN LI was born in Chengdu, Sichuan, China, in 1998. She received the B.E. degree from Chengdu Neusoft University, Chengdu, in 2020. She is currently pursuing the master's degree with Hebei University, Baoding, China. Her main research interests include intelligent image and text information retrieval.

BINGJIE TIAN received the B.A. degree from Industrial and Commercial College, Hebei University, China, in 2010, and the MTCSOL degree from the College of International Exchange and Education, Hebei University, in 2012. She is currently a Lecturer with the School of International Education College, Hebei Finance University. Her research interest includes the management of information resources.

XUEDONG TIAN received the B.S. degree from the Department of Automation Engineering, Hebei University of Technology, China, in 1984, the M.S. degree from the Department of Electronics and Information Engineering, Hebei University, Baoding, China, in 1998, and the Ph.D. degree from the College of Physics Science and Technology, Hebei University, in 2007. He is currently a Professor with the School of Cyber Security and Computer, Hebei University. His research interests include information retrieval and pattern recognition.