Mathematical Biosciences and Engineering

MBE, 19(4): 3748−3766.
DOI: 10.3934/mbe.2022172
Received: 26 October 2021
Revised: 13 January 2022
Accepted: 26 January 2022
Published: 10 February 2022

http://www.aimspress.com/journal/MBE

Research article

Multi-attribute scientific documents retrieval and ranking model based on GBDT and LR

Xuedong Tian[1,2,3,*], Jiameng Wang[1,2,3], Yu Wen[1] and Hongyan Ma[4]

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

* **Correspondence:** Email: xuedong_tian@126.com.

Abstract: Scientific documents contain a large number of mathematical expressions and texts containing mathematical semantics. Simply using mathematical expressions or text to retrieve scientific documents can hardly meet retrieval needs. The real difficulty in retrieving scientific documents is to effectively integrate mathematical expressions and related textual features. Therefore, this study proposes a multi-attribute scientific documents retrieval and ranking model based on GBDT (gradient boosting decision tree) and LR (logistic regression) by integrating the expressions and text contained in scientific documents. First, the similarities of the five attributes are calculated, including mathematical expression symbols, mathematical expression sub-forms, mathematical expression context, scientific document keywords and the frequency of mathematical expressions. Next, the GBDT model is used to discretize and reorganize the five attributes. Finally, the reorganized features are input into the LR model, and the final retrieval and ranking results of scientific documents are obtained. The experiment in this study was carried out on the NTCIR dataset. The average value of the final MAP@20 of the scientific document recall was 81.92%. The average value of the scientific document ranking nDCG@20 was 86.05%.

Keywords: scientific document retrieval and ranking; mathematical expression; multi-attribute feature; hesitation fuzzy sets; GBDT; LR

1. Introduction

Most existing search engines support text retrieval, but still have problems retrieving mathematical expressions, especially expressions without natural language annotations. While traditional search engines are losing their roles in this respect, recent research on mathematical expression retrieval has achieved relatively rich results [1−5].

Focusing on mathematical expressions in LaTeX format, Zhong et al. [6] proposed a mathematical formula retrieval algorithm based on Operator Tree. By matching multiple disjoint common subtrees with the same structure, the maximum number of sub-formulas is matched, which improves the efficiency of formula matching. Although the maximum number of matching sub-forms can improve retrieval accuracy, most sub-forms are more complicated. Therefore, the response time of real-time retrieval is approximately 20 s, which cannot meet the needs of real-time mathematical formula retrieval. To achieve faster sub-formulas retrieval, the team also proposed a strategy based on an inverted index and dynamic pruning [7], which improves the time efficiency of retrieval while ensuring that the retrieval results are still valid.

Focusing on mathematical expressions in MathML format, Schubotz et al. [8] proposed the VMEXT system, which can realize a visual tree of expressions in MathML format. It can also realize human-computer interaction, which is convenient for users to quickly find and improve the expression tree. In addition, similar or identical elements of two expressions can be visualized to calculate the similarity of expressions.

Focusing on mathematical expression images, Davila et al. [9] proposed a mathematical formula matching system. The system is mainly aimed at matching handwritten formulas on the teaching whiteboard with the formulas in course notes. First, the entire image was preprocessed, including formula search and structure correction. Then, the largest match in each image was identified by the symbolic consistent spatial alignment and similar relative sizes. Finally, each mathematical formula was divided into multiple symbol pairs. Symbol pairs are two symbols in a formula that are the nearest geometric neighbor of each other, which indicates the logical relations between them. The angle of a symbol pair is the angle between the line connecting the centers of the symbols and a horizontal line, which is helpful for judging the relationship between the two symbols. The images were sorted by the angle of the symbol pair.

With the development of deep learning, text embedding methods are widely used in natural language processing. Gao et al. [10] tried to apply the same method to formula embedding. They applied neural networks to mathematical information retrieval and proposed the “symbol2vec” model. This model was used to learn the vector representation of mathematical symbols and perform similarity calculations. Similarly, the NTFEM model [11] used an N-ary tree to convert the mathematical formula into a linear sequence. The word embedding model is used to embed the formula, and a weighted average embedding vector is obtained by using a weighting function. In mathematical formula retrieval, the BERT (bidirectional encoder representations from transformer)-based embedding model [12] is proposed to introduce more semantic information when the formula is embedded. The model uses the LaTeX format as the input and the BERT model is used to encode the formula. The index is built according to the embedded formula vector, formula id and post id from which the formula originates, and finally, the cosine similarity is used to obtain the final ranking of the formula.

In terms of fusion retrieval and ranking of mathematical expressions and scientific documents, Pathak et al. [13−15] committed to fusing expressions and related texts for retrieval. First, they proposed the MathIR system composed of three modules: "TS", "MS" and "TMS". This made scientific documents retrieval a similarity calculation of expression and text fusion rather than a simple expression search. Next, the "context-formula" pair was extracted, and the context of the formula was merged for retrieval. Finally, the modules of the system were optimized, and the formula retrieval was effectively integrated with the retrieval module for the text. Similarly, Schubotz et al. [16] regarded formulas and natural text as a single information source. The description of mathematical formula symbols was extracted from the surrounding text of the formula. These mathematical symbol descriptions were used to represent the definition of mathematical symbols. The namespace was formed as an internal data structure for mathematical information retrieval. This method can eliminate the ambiguity of mathematical symbols and better meet the retrieval needs of users. While retrieving mathematical expressions, Wang et al. [17] integrated other attributes to rank scientific documents, such as document category, types of journals to which scientific documents belong, and document citations. The sorting results were optimized by fusing these attributes of scientific documents. To better integrate mathematical expressions and text in scientific document retrieval, a weight parameter was proposed [18]. Based on formula similarity and text similarity, the proportion of text and mathematical expressions is manually adjusted.

In conclusion, current scientific document retrieval and ranking methods could be roughly divided into two types, the first type recalls by mathematical expression similarity and sorts by text similarity or recalls by text similarity and sorts by expression similarity. Regardless of what kind of similarity is used for the final sorting, it will weaken the similarity of another part. The second type manually adjusts the weight to fuse expression similarity and text similarity, but this method is difficult for users with less experience to control the specific values of the parameters. To solve the above problem, this study proposes a multi-attribute retrieval and ranking model of scientific documents that combines mathematical expressions and related texts. This model is an improvement of the second type, and can eliminate the need to manually adjust the weights of expressions and texts.

The similarity of five attributes is calculated: mathematical expression symbols (MESY), mathematical expression sub-forms (MESF), mathematical expression context (MECT), scientific document keywords (SDKY) and the frequency of mathematical expressions in scientific documents (FOME). A gradient boosting decision tree (GBDT) and logistic regression (LR) are used for feature reorganization and calculation to obtain the final search results, which improves the rationality of the retrieval.

2. Overview

Figure 1 shows a flow chart of the scientific documents retrieval and ranking system (solid lines denote online query flows and dotted lines denote offline index flows). The whole process consists of four parts: query preprocessing, scientific document preprocessing, multi-attribute similarity measure and scientific document retrieval and ranking.

The query preprocess module is used to process the input query. The query is a combination of mathematical expressions and text, which need to be split. The scientific document preprocessing module is used to extract mathematical expressions and related text, preliminarily decompose mathematical expression symbols and calculate the weights of related text. Then, the module interacts with the database module to store and index the information corresponding to the scientific documents to facilitate subsequent similarity calculations. The multi-attribute similarity measure module calculates the similarity of the five attributes of scientific documents. According to the different characteristics of each attribute, different similarity calculation algorithms are set up. The module interacts with the database module to store the calculated similarity. The scientific document retrieval and ranking module combines the similarity of the multiple attributes of scientific documents to fuse and calculate the attributes. Finally, the similarity between the scientific documents and the input query is obtained, and the scientific documents are ranked according to the similarity.

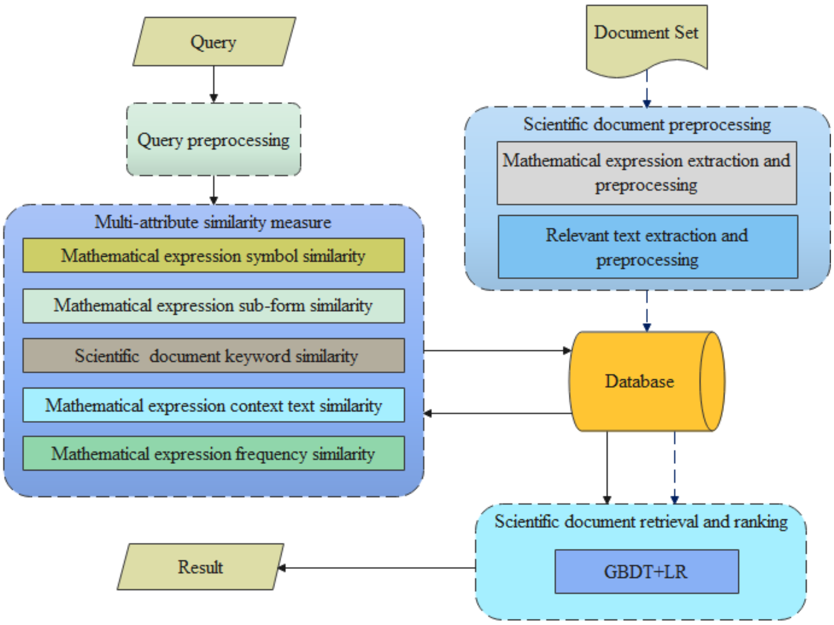
Query
Document Set
Query preprocessing
Scientific document preprocessing
Mathematical expression extraction and preprocessing
Relevant text extraction and preprocessing
Multi-attribute similarity measure
Mathematical expression symbol similarity
Mathematical expression sub-form similarity
Scientific document keyword similarity
Mathematical expression context text similarity
Mathematical expression frequency similarity
Database
Scientific document retrieval and ranking
GBDT+LR
Result

Figure 1. Flow chart of the scientific documents retrieval and ranking system.

3. Multi-attribute similarity calculation of scientific documents

3.1. Similarity calculation of mathematical expression symbols (MESY)

For the retrieval of mathematical expressions, there will be problems when inputting query expressions, such as inaccurate input and incorrect input of mathematical symbols. It is necessary to retrieve each mathematical symbol one by one to improve the fault-tolerant performance of the system.

Definition 1 ME_{Q} is the query expression, $ME_{\mathrm{D}t}(t=1,2,...,T_{\mathrm{ME}})$ is the mathematical expression dataset from the scientific documents, and T_{ME} is the number of mathematical expressions in the dataset.

First, FDS [19] is used to normalize the mathematical expressions in various formats into a unified form by decomposing them into multiple mathematical symbols with the corresponding five attribute values called level, flag, count, ratio, and operator.

The “level” attribute represents the level of a mathematical symbol, based on its position relative to the horizontal baseline. For example, in the mathematical expression $\frac{b^2}{a}$, the level values of $\frac{\mathrm{W}}{\mathrm{W}}$, a, b and 2 are 0, 1, 1 and 2, respectively. “Flag” represents the spatial flag bit of a symbol. Table 1 shows the value of the flag taking x as an example. “Count” refers to the sequential position of a symbol in the mathematical expression. “Ratio” refers to the frequency of the operator in the mathematical expression. “Operator” refers to whether a mathematical symbol is an operator. If a symbol is an operator, it is marked as 1; otherwise, it is marked as 0.

Table 1. Examples of flag value.

Meaning of flag	Right	Above	Superscript	Subscript	Below	Contains	Left-superscript	Left-subscript
Example	$2x$	$\frac{x}{2}$	2^x	2_x	$\frac{2}{x}$	$\sqrt{x}$	${}^{x}2$	${}_{x}2$
Value of the flag of x	0	1	2	3	4	5	6	**7**

In this way, the mathematical expression is converted into a list, which is convenient for subsequent retrieval of expression symbols. Table 2 shows the membership functions of the five attributes [20]. According to the distribution of values in each attribute by symbols in the data set, the balance factors in the function is determined by using curve fitting. The values of each balance factor are as follows. $\alpha=0.1$, $\mu=0.2$, $\nu=0.9$, $\sigma=0.2$.

Table 2. Description of the five attribute membership functions.

Attribute	Membership function	Function description
level	$\mathrm{M}_{\mathrm{lev}}(T_D,T_Q)=e^{-\alpha\lvert level_D-level_Q\rvert}$	$level_D$ and $level_Q$ respectively represent the level of two terms. α is the balance factor
flag	$\mathrm{M}_{\mathrm{fla}}(T_D,T_Q)=\{(f_o,flag_{(D,Q)})\}$	$flag_{(D,Q)}$ refers to the spatial position relationship of the same term，if $flag_D$ = $flag_Q$, then $\mathrm{M}_{\mathrm{fla}}(T_D,T_Q)=1$, otherwise it is 0.
count	$\mathrm{M}_{\mathrm{cou}}(T_{\mathrm{D}},T_{\mathrm{Q}})=\dfrac{1}{1+\mu\lvert count_{\mathrm{D}}-count_{\mathrm{Q}}\rvert^{\upsilon}}$	μ、υ are balance factors
ratio	$\mathrm{M}_{\mathrm{rat}}(T_{\mathrm{D}},T_{\mathrm{Q}})=e^{-(\frac{rat_{\mathrm{D}}-rat_{\mathrm{Q}}}{\sigma})^2}$	σ is the balance factor
operator	$\mathrm{M}_{\mathrm{ope}}(T_D,T_Q)=\{(s_o,operator_D)\}$	$operator_D$ refers to whether the current term is an operator，If it is an operator, the value of $\mathrm{M}_{\mathrm{ope}}(T_D,T_Q)$ is 1, otherwise the value is 0.5

Table 3. Membership values of the five attributes in "$x+\sqrt{x}$" and "x^2+y".

Term	level	M_{lev}	flag	M_{fla}	count	M_{cou}	ratio	M_{rat}	operator	M_{ope}
x	(0,0)	1	(0,0)	1	(1,1)	1	(0.25,0.25)	1	(0,0)	0.5
$+$	(0,0)	1	(0,0)	1	(2,3)	0.8333	(0.25,0.25)	1	(1,1)	1
x	(1,0)	0.9048	(5,0)	0	(4,1)	0.6503	(0.5,0.25)	0.2096	(0,0)	0.5

After the membership calculation is completed, each symbol corresponds to a five-tuple membership degree vector, denoted by $list_{\mathrm{sym}}$. The structure of $list_{\mathrm{sym}}$ is shown in Eq (1).

$$list_{\mathrm{sym}}=\left(M_{\mathrm{lev}\_ex\_term},M_{\mathrm{fla}\_ex\_term},\ M_{\mathrm{cou}\_ex\_term},\ M_{\mathrm{rat}\_ex\_term},\ M_{\mathrm{ope}\_ex\_term}\right) \tag{1}$$

where the term refers to the current mathematical symbol and ex refers to the expression id corresponding to the current mathematical symbol.

Take ME_{Q} = "$x+\sqrt{x}$" and ME_{Dt} = "x^2+y" as examples. The three mathematical symbols that are the same in the two expressions are “ x ”, “ $+$ ” and “ x ”. Table 3 shows the attribute values and membership degrees after the decomposition of the three symbols.

Next, hesitant fuzzy sets [21−23] are used to calculate the membership degree of each mathematical symbol. Hesitant fuzzy sets have advantages in dealing with multi-attribute decision-making problems. The formula for calculating the similarity of expressions using hesitant fuzzy sets is shown in Eq (2).

Finally, the normalization calculation of the mathematical symbols is performed to obtain the similarity of the expressions. The specific algorithm is shown in Algorithm 1.

Algorithm 1 Similarity calculation algorithm of mathematical expressions based on mathematical symbols

```
INPUT: $ME_{\mathrm{Q}}$, $ME_{\mathrm{D}t}(t=1,2,...,T_{\mathrm{ME}})$
OUTPUT: $Sim_{\mathrm{Symbol}}$
    $term_{\mathrm{Q}}=\mathrm{FDS}(ME_{\mathrm{Q}})$
    $term_{\mathrm{D}}=\mathrm{FDS}(ME_{\mathrm{D}t})$
    Update $list_{\mathrm{D}}^{'}$ // $trem_{\mathrm{sym\_vec}}$ calculation of the five attributes of the same term in $term_{\mathrm{D}}$ and $term_{\mathrm{Q}}$,
    $trem_{\mathrm{sym\_vec}}$ is updated in the $list_{\mathrm{D}}^{'}$ together with the id and term in the $term_{\mathrm{D}}$.
    for $id$ in $(1,T_{\mathrm{ME}})$:
        for $term_{i}$ in $term_{\mathrm{Q}}$:
            $var=list_{\mathrm{D}\_id}.\mathrm{exists}(term_{i})$
            if $var=\mathrm{TRUE}$:
                $list_{\mathrm{D}\_id}.\mathrm{add}[MAX(\frac{term_{\mathrm{sym\_vec}}^{2}}{5})^{\frac{1}{2}}]$ //If the same term exists in the $list_{\mathrm{D}}$, take the one with the greatest similarity
                $list_{\mathrm{D}\_id}^{'}.\mathrm{delete}[MAX(\frac{term_{\mathrm{sym\_vec}}^{2}}{5})^{\frac{1}{2}}]$ //Delete the corresponding item in the $list_{\mathrm{D}}^{'}$
            else:
                $list_{\mathrm{D}\_id}.add(0,0,0,0,0)$ //If it does not exist, add (0,0,0,0,0).
    $Sim_{\mathrm{Symbol}}=\mathrm{SIM}(list_{\mathrm{D}},list_{\mathrm{Q}})$
END
```

Definition 2 $list_{\mathrm{D}}$ refers to a normalized list of mathematical symbols. $list_{\mathrm{D}}$ includes id, term, and five attribute membership values $trem_{\mathrm{sym\_vec}}$. $list_{\mathrm{Q}}$ refers to the $list_{\mathrm{D}}$ formed by calculating the membership degree with ME_{Q} itself, and the five-attribute membership degree of mathematical symbols in $list_{\mathrm{Q}}$ is (1,1,1,1,1).

Definition 3 The formula [20] for calculating Sim_{Symbol} in Algorithm 1 is shown in Eq (2).

$$\mathrm{SIM}(list_{\mathrm{D}},list_{\mathrm{Q}})=1-\left\{\frac{1}{5}\sum\left[\frac{1}{N_{\mathrm{RE}}}\sum_{j=1}^{N_{RE}}\left|M_{\mathrm{D}}-M_{\mathrm{Q}}\right|^{\lambda}\right]\right\}^{1/\lambda} \tag{2}$$

where M_{D} and M_{Q} are the five-tuple vector values with the same term in $list_{\mathrm{D}}$ and $list_{\mathrm{Q}}$ respectively; N_{RE} is the number of mathematical symbols of ME_{Q} after FDS decomposition; $\lambda>0$. When λ = 1, the formula degenerates to the standard Hamming distance. When λ = 2, the formula degenerates to the standard euclidean distance. In this study, $\lambda = 2$.

Take the two mathematical expressions in Table 3 as an example, we suppose that $x+\sqrt{x}$ is query and x^2+y is the mathematical expression with id = 1 in the data set. Algorithm 1 is used to calculate the similarity of these two expressions. The result of the first update of $list_{\mathrm{D}}^{'}$ is {[1, x , (1,1,1,1,0.5)], [1, +, (1,1,0.8333,1,1)], [1, x , (0.9048,0,0.6503,0.2096,0.5)]}. Next, the $list_{\mathrm{D}}$ is updated in the order of terms in the query, and the result is {[1, x , (1,1,1,1,0.5)], [1, +, (1,1,0.8333,1,1)], [1, $\sqrt{\ }$, (0,0,0,0,0)], [1, x , (0.9048,0,0.6503,0.2096,0.5)]}. Finally, formula (2) is used for similarity calculation and $N_{\mathrm{RE}} = 4$, M_{D} = [(1, 1, 1, 1, 0.5), (1, 1, 0.8333, 1, 1), (0, 0, 0, 0, 0), (0.9048, 0, 0.6503, 0.2096, 0.5)], M_{Q} = [(1, 1, 1, 1, 1), (1, 1, 1, 1, 1), (1, 1, 1, 1, 1), (1, 1, 1, 1, 1)].The final calculated SIM = 0.1425.

3.2. Mathematical expression sub-form similarity calculation (MESF)

Table 4. Three attribute descriptions and membership functions of ME_{Q} as a sub-form.

Attribute	Attribute description	Membership function
length	The length of $\mathrm{ME}_{\mathrm{D}t}$	$\mathrm{DM}_{\mathrm{len}}(\mathrm{ME}_{\mathrm{Q}},\mathrm{ME}_{\mathrm{D}t}) = \frac{length_{\mathrm{Q}}}{length_{\mathrm{D}t}}$
level	The level of ME_Q relative to $\mathrm{ME}_{\mathrm{D}t}$	$\mathrm{DM}_{\mathrm{lev}}(\mathrm{ME}_{\mathrm{Q}},\mathrm{ME}_{\mathrm{D}t}) = e^{-\alpha\cdot level_Q}$
flag	The flag of ME_Q relative to $\mathrm{ME}_{\mathrm{D}t}$	$\mathrm{DM}_{\mathrm{fla}}(\mathrm{ME}_{\mathrm{Q}},\mathrm{ME}_{\mathrm{D}t}) = \left\{\left(f_o, flag_Q\right)\right\}$

The mathematical expression sub-form similarity calculation refers to the retrieval of ME_{Q} as a whole object. ME_{Q} is equivalent to a part of $ME_{\mathrm{D}t}(t = 1,2,...,T_{\mathrm{ME}})$. The degree of membership of ME_{Q} to the three attribute values of $ME_{\mathrm{D}t}$ is calculated. The three attributes are length, level and flag. "Length" refers to the ratio of the length of the sub-formula to the length of the expression. The meaning of level and flag are the same as 2.1. Table 4 shows the membership functions corresponding to the three attributes [16].

Algorithm 2 Similarity calculation algorithm based on mathematical expression as sub-form

```
INPUT: $ME_{\mathrm{Q}}, ME_{\mathrm{D}t}(t = 1,2,...,T_{\mathrm{ME}})$
OUTPUT： $Sim_{\mathrm{Sub}}$
$ME_{\mathrm{MA}t} = ME_{\mathrm{D}t}.\mathrm{Replace}(ME_{\mathrm{Q}},"\#")$      // Use # to replace the same string as $\mathrm{ME}_Q$
Num= $ME_{MAt}$.Count( " # " )
$T_{MAt} = \mathrm{FDS}(ME_{MAt})$
for $term_{ma}$ in $T_{MAt}$ :
    if $term_{ma} == "\#"$:
        $list_{MAt}.add\left(DM_{len}, DM_{lev}, DM_{fla}\right)$
        $sim_{ma} = 1-\left\{\frac{1}{3}\sum\left[\frac{1}{Num}\sum_{j=1}^{Num}\left|list_Q - list_{MAt}\right|^{\lambda}\right]\right\}^{\frac{1}{\lambda}}$
        $list_{MA}(id, E_{sim}).add(T_{MAt}.id, sim_{ma})$ //add the mathematical expression id and similarity to $list_{MA}$ in turn
$Sim_{\mathrm{sub}} = list_{MA}.\mathrm{sort}()$      // sort () using the Built-in function in python
RETURN $Sim_{\mathrm{Sub}}$
END
```

A mathematical expression may contain multiple identical sub-expressions. After the membership of each attribute is calculated, the hesitant fuzzy set is used to calculate the similarity. It is the final similarity of the sub-form of ME_{Q} as $ME_{\mathrm{D}t}(t=1,2,...,T_{\mathrm{ME}})$. The specific algorithm is shown in Algorithm 2.

Definition 4 $DM_{len}, DM_{lev}, DM_{fla}$ represent the membership value of the three attributes length, level, and flag, respectively.

3.3. Contextual text similarity calculation of mathematical expressions (MECT)

BERT (bidirectional encoder representations from transformer) [22−24] is a pre-training language model that uses unsupervised data for pre-training and fine-tuning on the task corpus, and has excellent performance on tasks for understanding natural language. There are two tasks in the model pre-training phase: masked language mode and next sentence prediction. The joint training of these two tasks makes the word vector obtained by training more accurate and comprehensive. It can solve the polysemy problem that cannot be solved in word2vec.

This study uses mathematical expression contextual text to fine-tune BERT to achieve the similarity calculation of the contextual text. The specific algorithm is shown in Algorithm 3.

Algorithm 3 Contextual text similarity calculation algorithm

```
INPUT: $SE_{\mathrm{Q}}, SE_{\mathrm{D}t}(t = 1,2,\ldots,T_{\mathrm{SE}})$ // $SE_{\mathrm{Q}}$ is the query text. $T_{\mathrm{SE}}$ is the number of keywords in sentence
OUTPUT： $Sim_{\mathrm{CT}}$
    $V_{\mathrm{E}}$=Encode($SE_{\mathrm{Q}}$)          // Sentence embedding to $SE_{\mathrm{Q}}$
    $WOR_{\mathrm{Q}}$=jieba($SE_{\mathrm{Q}}$)      // The jieba tool (an open word segmentation tool) in python is used to segment $SE_{\mathrm{Q}}$
    for $WOR_r$ in $WOR_{\mathrm{Q}}$:
        $number$ = location($WOR_r$)          // Target each keyword
        $V_{\mathrm{w}}$ = summed_layers[$number$]          // Find word vectors in $V_{\mathrm{E}}$
        $simil = simil$ + MAX(cosine_similarity($V_{\mathrm{w}}, SE_{\mathrm{D}t}$))
    $Sim_{\mathrm{CT}}$ = $simil$ / len($WOR_{\mathrm{Q}}$)
    RETURN $Sim_{\mathrm{CT}}$
    END
```

3.4. *Similarity calculation of other attributes of scientific documents*

3.4.1. Similarity calculation of scientific document keywords (SDKY)

The Jaccard coefficient is used to calculate the similarity of two sets (G_A, G_B). It is expressed as the ratio of the intersection and union of the two sets, and can effectively calculate the degree of overlap between the two sets to obtain the similarity of the sets. Its definition is shown in Eq (3).

$$\mathrm{Jaccard}(G_A,G_B)=\frac{|G_A\cap G_B|}{|G_A\cup G_B|}=\frac{|G_A\cap G_B|}{|G_A|+|G_B|-|G_A\cap G_B|} \tag{3}$$

Each scientific document often contains a specific topic. The keywords of the documents are extracted, and similarity matching with the query words can improve the accuracy of the search results. The contents of the scientific document are divided into words. By calculating the weight of the words, the 5 words with the highest weights are selected as the keywords of the scientific document. The weight calculation method is shown in Eq (4).

$$W_{i,wor} = \frac{p_{i,wor}}{\sum_{n=0}^{N}\sum_{k=0}^{K} p_{n,k}} \cdot \lg \frac{N}{1+m_{wor}} \tag{4}$$

where $W_{i,wor}$ refers to the weight of the keyword wor in scientific document i, $p_{i,wor}$ refers to the total number of times wor appears in i. N refers to the total number of scientific documents. k refers to the number of keywords in the current scientific document. m_{wor} refers to the number of documents containing wor.

Since the difference in text length will affect the calculated keyword similarity, this study improves the Jaccard coefficient and adds the length difference part. The calculation of similarity is shown in Eq (5).

$$Sim_{\mathrm{Wor}} = \mathrm{Jaccard}(SE_{\mathrm{Q}}, WE_{\mathrm{DT}}) = \frac{\left|SE_{\mathrm{Q}} \cap WE_{\mathrm{DT}}\right|}{\left|SE_{\mathrm{Q}}\right| + \left|WE_{\mathrm{DT}}\right| - \left|SE_{\mathrm{Q}} \cap WE_{\mathrm{DT}}\right| + \phi\left|\mathrm{len}(SE_{\mathrm{Q}}) - \mathrm{len}(WE_{\mathrm{DT}})\right|} \tag{5}$$

where WE_{DT} refers to the keyword collection of scientific documents, and ϕ is the balance factor.

3.4.2. The frequency of mathematical expressions in scientific documents (FOME)

When retrieving scientific documents, the same mathematical expression appears differently in different scientific documents, and the importance and retrieval order of scientific documents are also different.

The frequency of mathematical expressions in scientific documents is the product of the frequency of mathematical expressions in the document (EF) and the inverse document frequency (EIDF), which is similar to TFIDF. The difference is that when the text frequency is calculated, the query text must be exactly the same as the text in the document before the text can be considered to appear once. In the process of searching for mathematical expressions, partially identical expressions can also be considered to appear once. For example, when ME_Q is $U=IR$, the appearance of $U=IR$ or $I=\frac{U}{R}$ in a scientific document can be counted as one occurrence of the mathematical expression.

The frequency of occurrence of mathematical expression ME_{Q} refers to Sim_{Symbol} and Sim_{Sub} to calculate.

The frequency of expressions is related to the total number of mathematical expressions in the scientific document. The more expressions contained in the scientific document, the smaller the value of EF. The calculation method of EF is shown in Eq (6).

$$EF = \frac{1}{1+e^{-\frac{\mathrm{ROUND}(Sim_{symbol}) + \mathrm{ROUND}(Sim_{sub}) - \mathrm{NUM}(Sim_{symbol}.\mathrm{exp} \cap Sim_{sub}.\mathrm{exp})}{Doc_{num}}}} \tag{6}$$

where Doc_{num} refers to the total number of mathematical expressions in the scientific document. ROUND() refers to the rounding function with 0.5 as the demarcation point, and a value greater than 0.5 is recorded as 1; otherwise. it is 0. $Sim_{\mathrm{Symbol}}.\mathrm{exp}$ refers to the expression corresponding to Sim_{Symbol}.

NUM () calculates the number of the same expressions recalled by Sim_{symbol} and Sim_{sub}.

The calculation of EIDF requires the number of occurrences of ME_Q in the dataset. If ME_Q appears multiple times in different scientific documents, its importance will decrease accordingly. The calculation method of the EIDF is shown in Eq (7).

$$EIDF = \log\left(\frac{\text{N}}{\text{INCLUDE}(exp)+1}\right) \tag{7}$$

where N refers to the total number of scientific documents in the data set, $\text{INCLUDE}(exp)$ refers to the number of scientific documents containing exp. The specific calculation of $\text{INCLUDE}(exp)$ is shown in Eq (8).

$$\text{INCLUDE(exp)} = \begin{cases} 1 & \text{ROUND}(\text{Sim}_{symbol}) \vee \text{ROUND}(\text{Sim}_{sub}) = 1 \\ 0 & \text{ROUND}(\text{Sim}_{symbol}) \wedge \text{ROUND}(\text{Sim}_{sub}) = 0 \end{cases} \tag{8}$$

Finally, the calculation method of the frequency of mathematical expressions in scientific documents is shown in Eq (9).

$$Sim_{\text{fre}} = EF \cdot EIDF \tag{9}$$

4. Multi-attribute integration of scientific documents

The LR (logistic regression) model is based on linear regression plus sigmoid function (non-linear) mapping. It is shown as Eq (10).

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \tag{10}$$

where $\theta^T x$ is the input of the sigmoid, and θ and x are both matrices. θ is the linear regression parameter. T refers to the transpose of matrix. x refers to the feature of the input.

The LR model has a simple structure and fast running speed, but the learning ability and expression ability of the LR model are very limited. A large amount of feature engineering is required for feature dispersion and feature combination to increase the learning ability of the model. Therefore, an approach is needed for automatically discovering effective features and feature combinations and shortening the LR feature experiment cycle. The GBDT model can automatically discover features and carry out effective feature combinations.

GBDT (gradient boosting decision tree) [25−28] is a boosted tree model based on the CART regression tree model. In the process of generating each tree, the residual of the previous tree is calculated. The next tree is fitted on the basis of the residuals so that the residuals obtained on the next tree decrease. It is shown in Eq (11).

$$f_{CTM}(x) = \sum_{m=1}^{CTM} \text{T}(x;\psi_m) \tag{11}$$

where $\text{T}(x;\psi_m)$ refers to the CART regression tree, ψ_m refers to the parameters of CART, CTM refers to the number of CART.

The GBDT model construction algorithm is shown in Algorithm 4.

Algorithm 4 GBDT

```
INPUT   $T=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\}$   // Training dataset
OUTPUT  $f_{CTM}(x)$
        $f_0(x)=0$   // Initialization
        $L(y,f(x))=(y-f(x))^2$   // Define loss function
        for $m$ in (1, $CTM$):
        $r_{mi}=\left[\frac{\partial L(y_i,f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$   // The gradient of the $i$-th sample on the $m$-th tree
        $R_{mj}, j=1,2,\dots,J$   // The leaf node area of the $m$-th tree
        $c_{mj}=\arg\min_c \sum_{x_i\in R_{mj}} L(y_i,f_{m-1}(x_i)+c)$   // Best fit value for leaf area $j$
        $f_m(x)=f_{m-1}(x)+\sum_{j=1}^{J} c_{mj}I$
        RETURN   $f_{CTM}(x)=\sum_{m=1}^{CTM}\sum_{j=1}^{J} c_{mj}I$
        END
```

In this study, a GBDT + LR model is used. The five attributes MESY, MESF, MECT, SDKY and FOME are selected. GBDT can automatically combine and discretize features. After the decision tree is established, the path from the root node to each leaf node is a combination of different features. Each leaf node represents a unique combination of features. After the combination is completed, it is transferred to the LR model for secondary training.

As shown in Figure 2, $Tree_1$ and $Tree_2$ are two GBDT trees. sim_{Symbol}, sim_{Sub}, sim_{CT}, $sim_{\text{math\_fre}}$ and sim_{Wor} are represented by S1, S2, S3, S4 and S5 in the figure respectively.

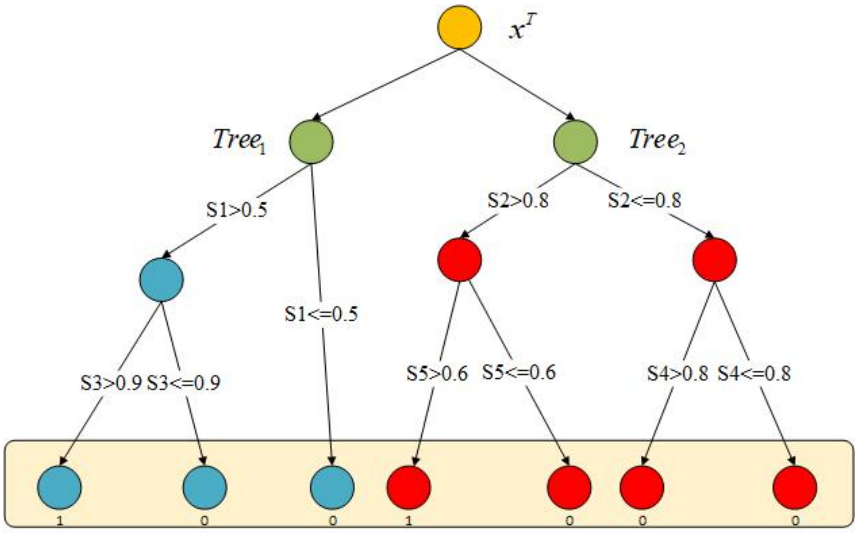

Figure 2. GBDT model diagram.

The sample x^T is judged by two tree nodes and belongs to different leaf nodes of the two trees. The leaf nodes of the two trees are coded. The leaf nodes to which sample x^T belongs are marked as 1, and the others are marked as 0. The leaf node codes of the two trees are connected in series to form a seven-dimensional sample (1, 0, 0, 1, 0, 0, 0).

Each x^T will go through multiple GBDT trees to recombine features. For GBDT trees, the path from the root node of the tree to the leaf nodes is a combination of different features. Therefore, the leaf node can uniquely represent this path. The leaf node is input into the LR model as a discrete feature for training.

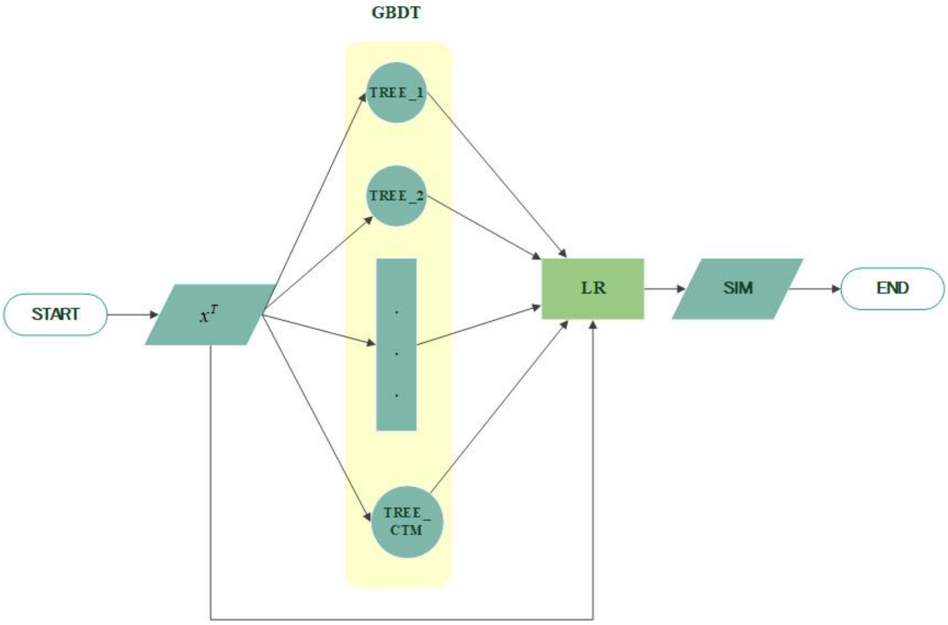

Figure 3. GBDT + LR system flow chart.

In the final prediction, the input sample will pass through each tree of GBDT to obtain a discrete feature (a set of feature combinations) corresponding to a certain leaf node. Then, the feature is passed into LR in one-hot form for linear weighted prediction. The final similarity SIM calculation result is obtained. Figure 3 shows the specific flow chart. For the LR model, the L2 penalty term is used, and the value of the inverse of regularization strength is 0.05. For the GBDT model, the metric is “binary_logloss”, num_leaves is 32, num_trees' is 60 and the learning_rate is 0.005.

5. Experimental results and analysis

5.1. *Experimental data and environment*

The dataset used in the experiment is “MathTagArticles” in NTCIR-12_MathIR_Wikipedia_Corpus, which includes 31742 scientific documents. The “MathTagArticles” includes 16 archive files (they are coded as wpmath0000001-wpmath0000016)，and each archive file contains about 2000 scientific documents. In this study, the hold-out method is used: “wpmath0000001-wpmath0000008” are used for training, “wpmath0000008-wpmath0000012” are used for verification, “wpmath0000013-wpmath0000016” are used for testing. Table 5 shows the experimental environment.

Table 5. Experimental environment.

Experimental environment	Configuration
Processor	Intel(R) Xeon(R) Silver 4215 CPU @ 2.50GHz
RAM	32GB
Operating system	Linux
Graphics card	GeForce RTX 2080
Video memory	8G
Python version	3.6
TensorFlow version	1.14.0

5.2. Evaluation protocol

5.2.1. Relevance ratings

The evaluators are five mathematics graduate students who are familiar with mathematical expressions and scientific documents. For each set of queries, the top 10 results are selected for evaluation. The evaluation indicators are relevant, partially relevant and not relevant. Among them, relevant ones are marked as 2, partially relevant ones are marked as 1, and not relevant ones are marked as 0. The results of the same query will be marked separately by five evaluators. Different evaluators should not mark the same retrieval result too differently. For example, for the same search result, when some commenters are marked as 2, other commenters can mark 1 or 2, but cannot mark 0. So, another labeling rule is set: for the same result, the difference between the scores of different evaluators should be less than or equal to 1. If it is greater than 1, the marks are invalid.

Finally, the results of the five evaluators are summarized. The reviewer's score is converted to a comprehensive score in Table 6. Based on the principle of obedience to the majority, a total score greater than 7 is considered relevant, a total score greater than 2 is considered partially relevant, otherwise it is not relevant. In the subsequent evaluation of results, if the evaluation metrics only require relevant and not relevant, the partial relevant will default to relevant.

Table 6. Relevance assessment.

Assessment	Individual	Combined
Relevant	2	8−10
Partially Relevant	1	3−7
Not Relevant	0	0−2

5.2.2. Test case

This study conducts a large number of experiments on different types of mathematical expressions and related texts. Twenty representative query expressions and texts are selected by evaluators for statistical experiments. Queries are based on the actual situation of the mathematics set, integrating the different structures of mathematical expressions and the different fields involved in scientific and technological documents. Table 7 provides the query expressions and text.

Table 7. 20 queries in system experiment.

NO.	Expression	Text	NO.	Expression	Text
1	$\sqrt{b^2-4ac}$	discriminant	11	$\frac{1}{2}mv^2$	theorem of kinetic energy
2	$\sin^2\alpha+\cos^2\alpha$	trigonometric function	12	$f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n$	Taylor function
3	$\frac{a}{b}$	proportion	13	$(x-a)^2+(y-b)^2=r^2$	round
4	$\sin\theta$	sine function	14	$U=IR$	Ohm law
5	$y^2=2px$	parabola	15	$a^2+b^2=c^2$	Pythagorean theorem
6	$S=4\pi R^2$	surface area	16	$\frac{P(B_i)P(A\mid B_i)}{\sum_{j=1}^{n}P(B_j)P(A\mid B_j)}$	Bayes
7	2^q		17	$2\sum_{i=1}^{4}\frac{1}{r_i^2}$	Descartes theorem
8	$E=mc^2$	mass energy	18	$f(b)-f(a)=f^{'}(\varepsilon)(b-a)$	Lagrange
9	$f(x+2\Pi)=f(x)$		19	$\lim_{n\to\infty}(1+\frac{1}{n})^n$	limit theorem
10	$a_n=a_1+(n-1)d$	arithmetic sequence	20	$\frac{1}{\sigma\sqrt{2\Pi}}e^{\frac{(x-\mu)^2}{2\sigma^2}}$	normal distribution

5.3. System experiment

Precision represents the accuracy rate and refers to the proportion of related documents in all the query documents in the returned results of the query. The calculation formula is shown in Eq (12).

$$\text{precision}=\frac{SR_{\text{tp}}}{SR_{\text{tp}}+SR_{\text{fp}}} \tag{12}$$

where SR_{tp} refers to the number of Query-related documents in the query result. SR_{tp} refers to the number of documents irrelevant to the Query in the query result.

Reciprocal rank (RR) is the reciprocal of the ranking of the first related document in the retrieved results. MRR is the average of the reciprocal rankings of multiple queries, and the calculation method is shown in Eq (13).

$$\text{MRR}=\frac{1}{k}\sum_{i=1}^{k}\frac{1}{\text{rank}(i)} \tag{13}$$

where $\text{rank}(i)$ refers to the ranking of the first related document for the i-th query.

Table 8 shows the values of P@3, P@5, P@10, and MRR for the 20 queries in Table 6, and Figure 4 shows the values of P@3, P@5 and P@10 for the 20 queries in Table 7.

Figure 4 shows that the P@3 of some queries can reach 100%. However, the precision of some queries is low, which is related to the fact that there are fewer scientific documents matching it in the dataset.

Table 8. Average of precision and average of MRR.

Evaluation indicator	P@3	P@5	P@10	MRR
Average	0.81	0.76	0.58	0.87

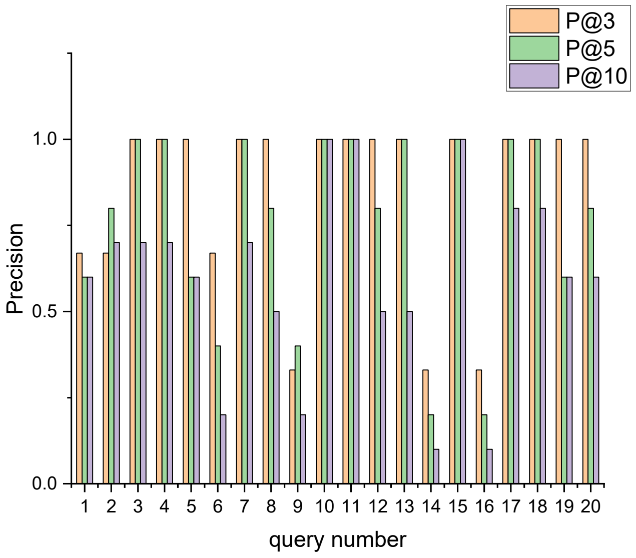

Figure 4. Precision value of different queries.

5.4. Ablation experiment

Average precision considers the position factor on the basis of precision. It is more sensitive to the position of sorting. The calculation method is shown in Eq (14).

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

where r refers to the total number of related documents, $\mathrm{pos}(i)$ refers to the position of the i-th related document in the retrieved results.

NDCG is the normalized loss cumulative gain. The calculation method of DCG (discounted cumulative gain) is shown in Eq (15).

$$\mathrm{DCG}@k = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)} \tag{15}$$

where rel_i refers to the relevance of the i-th document. There are three levels of relevance: good, fair and bad. They are assigned scores of 3, 2 and 1.

In an ideal state, according to the order of relevance from largest to smallest, the case where DCG takes the maximum value is IDCG.

$$\mathrm{IDCG}@k = \sum_{i=1}^{REL} \frac{rel_i}{\log_2(i+1)} \tag{16}$$

where REL refers to the sorting situation of the documents in the ideal state, and k refers to the collection of the first k documents.

NDCG uses IDCG to normalize the evaluation indicators.

$$\mathrm{nDCG}@k = \frac{\mathrm{DCG}@k}{\mathrm{IDCG}@k} \tag{17}$$

The similarities of the five attributes of scientific documents are calculated separately, they are MESY, MESF, MECT, SDKY and FOME. In order to verify the role of each attribute in the experiment, an ablation experiment was carried out in this study. One of the five attributes is removed in turn, and the remaining four attributes are input into GBDT and LR for training, then five models are obtained. Experiments with these five models are compared with the original model, and the results obtained are shown in the Figure 5. In Figure 5, model A represents MESF + MECT + SDKY + FOME, model B represents MESY + MECT + SDKY + FOME, model C represents MESY + MESF + SDKY + FOME, model D represents MESY + MESF + MECT + FOME, model E represents MESY + MESF + MECT + SDKY, and the model F represents MESY + MESF + MECT + SDKY + FOME.

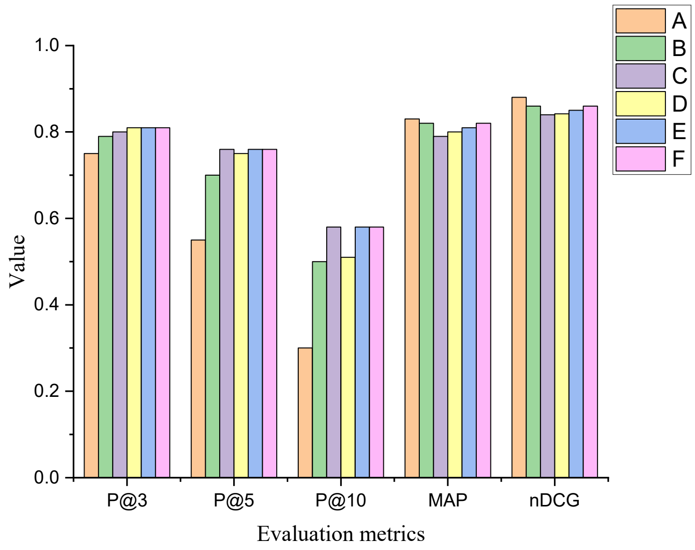

Figure 5. Results of ablation experiments.

As shown in Figure 5, the MESY attribute affects the precision of the model. There are fewer relevant results retrieved, and the less relevant results are ranked relatively higher, so the MAP and nDCG of model A will be slightly higher. MESF also affects the precision of the model, but has little effect on the ranking. The two attributes of MECT and FOME have little effect on precision, but they will affect the ranking of results. The SDKY attribute will get more relevant results and affects the ordering of the model to some extent.

5.5. Comparative experiment

Figures 6 and 7 show the comparison results of the algorithm in this study with Tangent-CFT [4] and MIaS [3], MIaS system is an open-source system. The Tangent-CFT model was reproduced experimentally. Table 9 gives the average comparisons of MAP and NDCG. Tangent-CFT [4] is a mathematical expression embedding model realized by word2Vec, that can achieve precise matching of mathematical expression structure. To locate a scientific document according to a mathematical expression, the retrieval of "mathematical expression-scientific document"(scientific document pairs corresponding to mathematical expressions) is realized. MIaS [3] is an open search engine for mathematical expressions. It can also retrieve corresponding scientific and technological documents based on the similarity of mathematical expressions. The system builds an XML tree through the structure of mathematical expressions to retrieve query expressions and expressions with query expressions as sub-expressions.

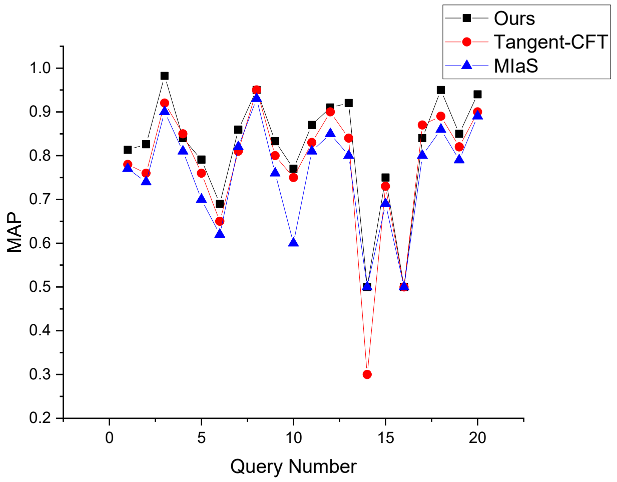

Figure 6. MAP comparison of different algorithms.

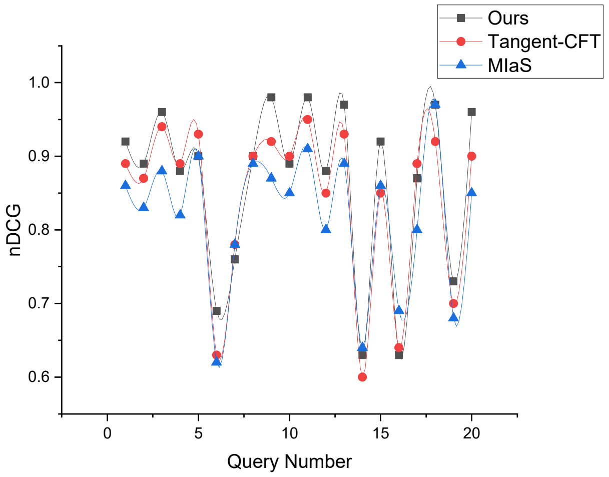

Figure 7. nDCG comparison of different algorithms.

Table 9. Comparison of the average MAP and NDCG of different algorithms.

Algorithms	Ours	Tangent-CFT	MIaS
MAP	0.8192	0.7805	0.7570
nDCG	0.8605	0.8300	0.8100

6. Conclusions

This study proposes a multi-attribute retrieval and ranking model based on GBDT + LR to solve the problem of poor integration of mathematical expressions and relevant texts in scientific document retrieval. This method combines the five attributes MESY, MESF, MECT, SDKY and FOME. GBDT is used to reorganize the features, and LR trains the reorganized features. Finally, the similarity of the final scientific documents is obtained and sorted.

Future research is expected to complete the semantic retrieval of expression symbols based on the context of expressions. Meanwhile, in terms of semantics, it is better to effectively integrate expressions and text. When sorting the final scientific documents, the attributes of the scientific documents themselves should be considered, such as, publication year and citation frequency. This can improve the rationality and effectiveness of the final scientific document sorting.

Acknowledgments

This work is supported by the Natural Science Foundation of Hebei Province of China (Grant No. F2019201329).

Conflict of interest

The authors declare no conflict of interest.

References

1. K. Yamada, H Murakami, Mathematical expression retrieval in PDFs from the web using mathematical term queries, in *33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, (2020), 155−161. https://doi.org/10.1007/978-3-030-55789-8_14
2. R. M. Oliveira, F. B. Gonzaga, V. Barbosa, G. Xexéo, A distributed system for search on math based on the microsoft bizSpark program, preprint, arXiv:1711.04189.
3. P. Sojka, M. Ruzicka, V. Novotný, MIaS: Math-aware retrieval in digital mathematical libraries, in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, (2018), 1923−1926. https://doi.org/10.1145/3269206.3269233
4. B. Mansouri, S. Rohatgi, D. Oard, J. Wu, C. L. Giles, R. Zanibbi, Tangent-CFT: An embedding model for mathematical formulas, in *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, (2019), 11−18. https://doi.org/10.1145/3341981.3344235
5. J. M. Xu, C. Y. Xu, *Computing similarity of scientific documents based on texts and formulas*, Data and Knowledge Discovery, **2** (2018), 103−109. Available from: https://wenku.baidu.com/view/3ca592af1cd9ad51f01dc281e53a580217fc500d.html?fr=income1-wk_app_search_ctr-search
6. W. Zhong, R. Zanibbi, Structural similarity search for formulas using leaf-root paths in operator subtrees, in *European Conference on Information Retrieval,* (2019), 116−129. https://doi.org/10.1007/978-3-030-15712-8_8
7. W. Zhong, S. Rohatgi, J. Wu, C. L. Giles, R. Zanibbi, Accelerating substructure similarity search for formula retrieval, *Adv. Inf. Retr.*, **12035** (2020), 714−727. https://doi.org/10.1007/978-3-030-45439-5_47
8. M. Schubotz, N. Meuschke, T. Hepp, H. Cohl, B. Gipp, VMEXT: a visualization tool for mathematical expression trees, preprint, arXiv:1707.03540v1
9. K. Davila, R. Zanibbi, Visual search engine for handwritten and typeset math in lecture videos and LATEX notes, in *16th International Conference on Frontiers in Handwriting Recognition*, (2018), 50−55. https://doi.org/10.1109/ICFHR-2018.2018.00018
10. L. Gao, Z. Jiang, Y. Yin, K. Yuan, Z. Yuan, Z. Tang, Preliminary exploration of formula embedding for mathematical information retrieval: Can mathematical formulae be embedded like a natural language?, preprint, ArXiv: 1707.05154.

11. F. Dai, L. Chen, Z. Zhang, An N-ary tree-based model for similarity evaluation on mathematical formulae, in *2020 IEEE International Conference on Systems, Man, and Cybernetics*, (2020), 2578−2584. https://doi.org/10.1109/SMC42975.2020.9283495
12. P. Dadure, P. Pakray, S. Bandyopadhyay, BERT-based embedding model for formula retrieval, in *Conference and Labs of the Evaluation Forum*, 2021.
13. A. Pathak, P. Pakray, R. Das, Context guided retrieval of math formulae from scientific documents, *J. Inf. Optim. Sci.*, **40** (2019), 1559−1574. https://doi.org/10.1080/02522667.2019.1703255
14. A. Pathak, P. Pakray, A. Gelbukh, Binary vector transformation of math formula for mathematical information retrieval, *J. Intell. Fuzzy Syst.*, **36** (2019), 4685−4695. https://doi.org/10.3233/JIFS-179018
15. A. Pathak, P. Pakray, S. Sarkar, D. Das, A. Gelbukh, MathIRs: Retrieval system for scientific documents, *Computacióny Sistemas*, **21** (2017), 253−265. https://doi.org/10.13053/CyS-21-2-2743
16. M. Schubotz, A. Grigorev, M. Leich, H. Cohl, N. Meuschke, B. Gipp, et al., Semantification of identifiers in mathematics for better math information retrieval, in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, (2016), 135−144. https://doi.org/10.1145/2911451.2911503
17. H. B. Wang, X. D. Tian, K. G. Zhang, X. J. Cui, Q. X. Shi, X. F. Li, A multi-membership evaluating method in ranking of mathematical retrieval results, *Sci. Technol. Eng.*, **19** (2019), 164−170.
18. D. Fraser, A. Kane, F. W. Tompa, Choosing math features for BM25 ranking with tangent-L, in *Proceedings of the ACM Symposium on Document Engineering*, (2018), 1−10. https://doi.org/10.1145/3209280.3209527
19. X. Tian, S. Yang, X. Li, F Yang, An indexing method of mathematical expression retrieval, in *3rd International Conference on Computer Science and Network Technology*, (2013), 574−578. https://doi.org/10.1109/ICCSNT.2013.6967179
20. X. Tian, J. Wang, Retrieval of scientific documents based on HFS and BERT, *IEEE Acccess*, **9** (2021), 8708−8717. https://doi.org/10.1109/ACCESS.2021.3049391
21. L. Zadeh, Fuzzy Sets, *Inf. Control*, **8** (1965), 338−353. https://doi.org/10.1016/S0019-9958(65)90241-X
22. V. Torra, Hesitant fuzzy sets, *Int. J. Intell. Syst.*, **25** (2010), 529−539. https://doi.org/10.1002/int.20418
23. L. N. Cai, S. W. Chen, W. Zhou, H. B. Huang, Y. Liang, Interval-valued hesitant fuzzy WOWA operator and its application in decision making, Journal of Zhengzhou University, **35** (2014), 49−53.
24. A. Reusch, M. Thiele, W. Lehner, TU_DBS in the ARQ math lab 2021, in *Conference and Labs of the Evaluation Forum*, (2021), 107−124.
25. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, preprint, arXiv: 1810.04805.
26. Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun. Q. Liu, ERNIE: Enhanced language representation with informative entities, preprint, arxiv: 1905.07129.
27. Z. Tian, R. Zhang, X. Hou, J. Liu, K. Ren, FederBoost: Private federated learning for GBDT, preprint, arXiv: 2011.02796.
28. F. Fu, J. Jiang, Y. Shao, B. Cui, An experimental evaluation of large scale GBDT systems, preprint, arxiv: 1907.01882.