Acta Technica *62*, No. 3B/2017, 459–470

Complex mathematical expression retrieval based on hierarchical index[1]

Xuedong Tian[2,3], Nan Zhou[2]

Abstract. Mathematical expressions have many special attributes compared with normal text, which results in the traditional full-text searching engine could not serve the mathematical query tasks effectively. This problem is even remarkable when the searching objects are the expressions in linear algebra field because of their complex internal structures. Focusing on the issue, a retrieval model of the mathematical expressions including linear algebra formulae is proposed. Firstly, a novel feature structure was defined for describing the row-column structure of linear algebra expressions in which the sub-expressions are viewed as components with special row-column flags for recursively processing. Then, the linear algebra expressions described in LaTeX were analyzed and their features were extracted. Thirdly, a hierarchical math index was designed for math retrieval. Finally, the matching algorithms of linear algebra expressions were presented. The experiments were carried out on a mathematical expression corpus which contains 141714 formulas including 3013 linear algebra expressions. The response time of math retrieval is from 107 ms to 3336 ms. This result shows the effectiveness of the proposed method.

Key words. Mathematical expression retrieval, linear algebra expressions, row-column structure, hierarchical index, matching..

1. Introduction

The rapid growth of scientific information with large amounts of mathematical expressions requires powerful retrieval technology for people finding the required math materials quickly and conveniently. However, the complex expressing pattern of mathematical formulae brings more difficulties for realizing math retrieval than normal text, which results in the low performance of the traditional full-text searching engine in treating formulae. Therefore, it is necessary to research and develop special retrieval model for mathematical expressions.

[1]This work is supported by the National Natural Science Foundation of China (Grant No. 61375075), and the Key Project of the Science and Technology Research Program in University of Hebei Province of China (Grant No. ZD2017208).

[2]School of Computer Science and Technology, Hebei University, No. 180, Wusi East Road, Baoding, Hebei, China, 071002

[3]Corresponding author's email: xuedong_tian@126.com

Nowadays, the math retrieval technology is far from mature in theory and applications. The research work about math retrieval could be divided into two classes. The first type is to build the math retrieval function on the basis of traditional full-text searching engine through converting mathematical expressions into linear strings. DLMF (Digital Library of Mathematical Functions) [1]–[2] converted mathematical expressions into a serialized text equivalent, a text searching engine was employed to realize math search. MathDex [3] realized math retrieval by employing a traditional full-text searching engine through converting formulae into linear text strings with a multi-pass normalization process. In Literature [4], the retrieval method of mathematical content employed in LeActiveMath was described. The mathematical information was descripted in OMDoc format. The math query modes include simple formulae and formulae with wildcards. The math retrieval was realized through converting query expressions in OpenMath into text strings with the help of expression trees and matching with Lucene library. EgoMath [5]–[6] is also a math retrieval system based on full text searching engine. Its object is to realize mathematical retrieval in Wikipedia. Mias (Math Indexer and Searcher) system [7] is a math retrieval system based on Apache Lucene search engine. It realized math searching by analyzing the mathematical expressions in Presentation MathML and extracting the sub-formulae features for math indexing and retrieving. WikiMirs [8] targeted at providing math retrieval function in Wikipedia. A special indexing and matching model was proposed for realizing sub-structure matching and similarity matching of mathematical expressions based on layout structures. It is composed of four parts called preprocessor, tokenizer, indexer and ranker. Lin et al. [9] proposed a mathematics retrieval method. The user query expressions could be input not only with math markup language but also through clipping formula on PDF layouts. Its characteristics is the conversion from structured formulae to linear terms through semi-operator tree which ensures the realization of math retrieval based on the traditional full text index structure.

The second class is to design special math indexing and matching model which fully considers the characteristics of formulae through employing corresponding data structures, such as tree structure. MathWebSearch [10]–[11] could collect, index and search math documents with full-text searching function. MathSearch [12]–[13] designed a special mathematics query language called MQL (Math Query Language) for users to input their math query requirements exactly and conveniently. An N-gram formula division method was designed for formula analyzing and indexing. LaTeXSearch [14] is provided by Springer with three query modes: Latex Code, Article Title and Article DOI. The system returns exact or similar result respectively.

Although some achievements have been gained in math retrieval, there are many questions to be researched. In this paper, aiming at the situation that the linear algebra has not been fully noticed in math retrieval research, a retrieval model is proposed including the defining and extracting of retrieval feature, the constructing of math index and the matching algorithm of linear algebra expressions. And the experimental result and analysis are given for illustrating the effectiveness of the proposed method.

2. Index of linear algebra expressions

The main expressing modes of linear algebra such as matrices, determinants and systems of linear equations are all called complex expressions here (simply called LAEs). They could be viewed as a kind of special compound components in math field.

Definition 1. S_{I} is a set of simple components which are relatively independence in mathematical expression F, including function names, variable names, constant names and operators. For example, "sin()", "x", "C" and "+" all belong to S_{I}.

Definition 2. S_{C} is a set of LAEs in mathematical expression F including the typical modes of linear algebra, such as matrixes, determinants and systems of linear equations, in which the elements are arranged in row-column structure. For example, "$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$", "$\begin{vmatrix} 1 & 4 \\ 2 & 5 \end{vmatrix}$" and "$\begin{cases} x_1 - x_2 = a_1 \\ x_2 - x_3 = a_2 \end{cases}$" belong to S_{C}.

Definition 3. The attributes of a component E_j in mathematical expression F could be represented as a four-tuple E_j (*No*, *Level*, *Operator*, *Flag*) called FDS[15], Where *No* is the identifier of E_j; *Level* is the level number of E_j; *Operator* is the flag to indicate whether E_j be an operator, *Operator*$= 1$ indicates be an operator, *Operator*$= 0$ means E_j not be an operator; *Flag* is the flag of the relationship between E_j and its control symbol E_i. The control symbol E_i should satisfy the following condition

$$\begin{array}{l}(E_i.No < E_jNo) \wedge (E_i.Level = E_j.Level - 1) \\ \wedge (E_j.No - E_i.No = \min(E_k.No - E_i.No), \forall k < i)\end{array} . \tag{1}$$

When $E_j \in S_{\mathrm{I}}$, $Flag = 1, 2, 3, 4, 5, 6, 7$ indicate the relationships of upper, superscript, right, subscript and down between E_j and E_i. When $E_j \in S_{\mathrm{C}}$, Flag defined in [15] is to be extended. The value of the successor symbols of LAEs' delimiters equals to "$6v$", where v is the row number of E_j in the corresponding LAEs.

The retrieval tags of a mathematical expression of matrix $y = \begin{bmatrix} a-b & 2 \\ 1 & \frac{n}{m} \end{bmatrix}$ are shown in Fig. 1.

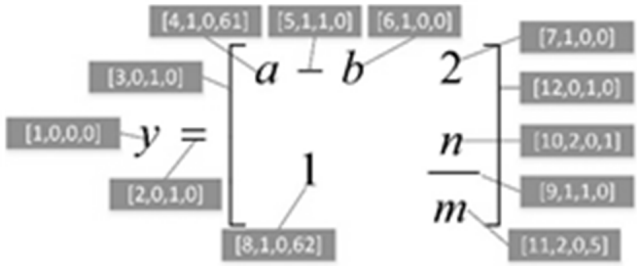

Fig. 1. Retrieval attributes of a matrix

All of the expressions within the matrix are considered to be located at the lower

layer than the delimiter "[" and "]" of matrix. The *Level* values of the symbols in these expressions are assigned according to the same rules as normal formulae. The *Flag* value of the first symbol in the sub expression "$a-b$" is "61" in which "6" indicates the including relation between "$a-b$" and "[", and "1" express "$a-b$" is located at the row 1 of the matrix. While the column attributes of a sub expression in a matrix is decided by the occurring order of value "61" in the matrix implicitly with the help of the separator "&" in LaTeX description. For example, the sub expression "$\frac{n}{m}$" has the column number "2" because the "\frac" in it is the second occurrence following the symbol "1". Once the elements are tagged, they could be indexed and matched as a normal mathematical expression through employing recursion strategy. The expressions of determinants and the systems of linear equations are tagged with the same way as matrixes.

The expressions of the systems of linear equations could be viewed as the matrix structure with multiple rows as well as single column. For example, "$\begin{cases} y=x^2-a \\ y=x^2+b \end{cases}$" is tagged as a matrix which has a structure of two rows and one column as shown in Fig. 2.

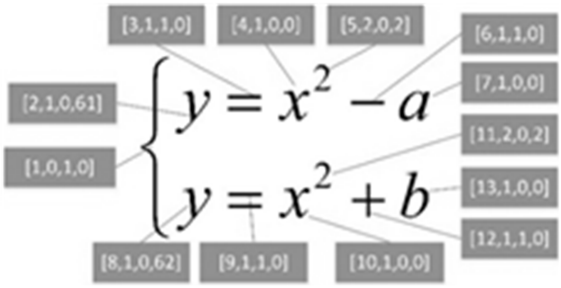

Fig. 2. The retrieval attributes of a systems of linear equations

2.1. The tagging algorithm of LAEs

The tagging process of LAEs is to analyze their symbol strings in LaTeX description recursively.

Algorithm 1. Tagging algorithm of complex expressions

Input: The LaTeX description of mathematical expression
Output: The FDS data of the mathematical expression

---

(1) Read a symbol.

(2) If the symbol is "/", goto Step 3; else, process it according to the FDS rules of normal symbols such as digits and operators, and goto Step 1.

(3) Get the current keyword. If it is an expression of linear algebra such as matrix, determinants, the systems of linear equations or special delimiters, goto Step 4; else, process it according to the FDS rules of normal symbols and move the pointer to

the next keyword, and goto Step 1.

(4) Extract the content included within the delimiters.

(5) Take "\\" as the line break symbol to segment and extract the line expressions.

(6) Define the identifier of line number: Line=0.

(7) For each line expression, looping executing (8).

(8) Line++

(9) Define the identifier of row number: Row=0

(10) Take "&" as the column break symbol to segment and extract the sub expressions

(11) For each sub expression

(12) Row++

(13) If Row==1

(14) Let Flag=61// the current sub expression

(15) Else

(16) Tagging the relations between elements according FDS

(17) Process each sub expression according to the FDS rules of normal symbols.

(18) Algorithm terminals.

---

3. The hierarchical index of mathematical expressions with LAEs

The hierarchical index [16] of mathematical expressions consists of two layers called Hash index layer and inverted index layer.

Hash index layer is arranged into Treap [17] structure, in which the key value is the Hash value of all symbols at the main baseline of formulae (*ExpCode*, *EC* for short), and the Priority value is the Hash value of the operators at the main baseline of formulae (*ExpStructureCode*, *ESC* for short).

Inverted index layer is constructed as a formula structure tree [18] with the pointers to the documents which have the corresponding math information.

Definition 4. The operators of mathematical expressions are called the two-dimensional operators (*TDO*) which expresses the math meanings through two dimensional structures explicitly or implicitly. For example, the operator "—" in a fraction structure is called a two-dimensional operator. And the symbol "∧" which exists in formula "a^2" implicitly indicates a superscript relations between "a" and "2", so "∧" is also called a two-dimensional operator.

In a formula structure tree, the child node in the lower layer belong to the father node which is the corresponding two-dimensional operator employing the child node as an operand in the higher layer. The hierarchical index of mathematical expressions is shown in Fig. 3.

In linear algebra expressions, the delimiters of matrixes, determinants and systems of linear equations, such as "[", "]", "|", "{" are viewed as operators on the main basic lines.

Take the matrix in Fig. 1 as an example. The symbols on the main line of the formula are "y", "=", "[" and "]" in which "=", "[" and "]" are operators. Therefore,

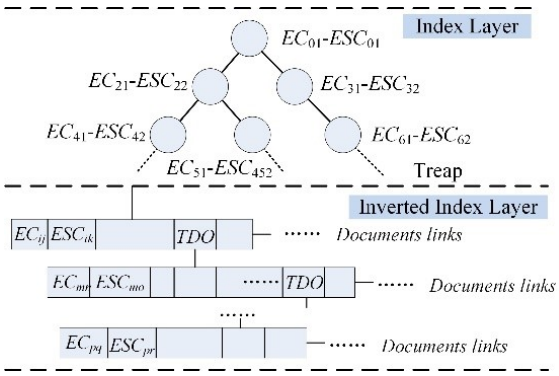

Fig. 3. Hierarchical index of math expressions

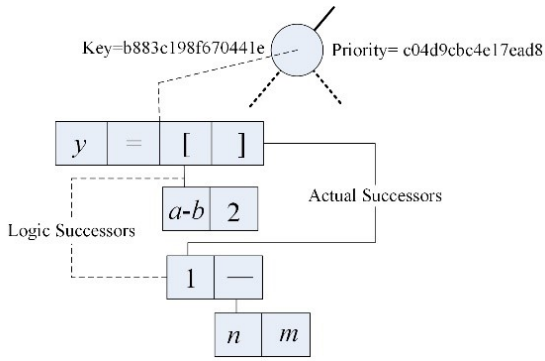

Fig. 4. Hierarchical index of a formula

EC = "y = [ ]" and ESC = " = [ ]". Take MD5 [19] to calculate the Hash value of EC and ESC as the key and Priority of the Treap index. Here, Hash (EC)= b883c198f670441e, Hash (ESC)= c04d9cbc4e17ead8. Its index structure is shown in Fig. 4.

4. The matching algorithm of LAEs

The matching operations of LAEs must fully consider the row-column flags of sub expressions within LAEs. Once the sub expressions are extracted from LAEs with the help of Flags in FDS, they could be treated as the normal math query expressions and searched with recursion strategy.

Algorithm 2. Matching algorithm of LAEs

Input: Input: The query expression of LAEs
Output: Output: The matching results

---

(1) Read the query expression of linear algebra.
(2) Extract the FDS data of query expression.
(3) Calculate the values of Key and Priority and locate the corresponding nodes in the Trap index layer.
(4) Match the normal components in the expression.
(5) Match the expression of linear algebra in the inverted index layer.
1) Define the row identifier of the linear algebra expression: Line=0.
2) Define the retrieval flag: IsSuccess=TRUE.
3) For each line expression, looping executing (8)
Line++
Define the identifier of row number: Row=0
For each sub expression in row
Match the FDS data of the current element.
If (! The corresponding element was matched)
IsSuccess=False
Row++
(6) If(IsSuccess==True)
The matching is success
Else The matching is failed
(7) Return the matching results.
(8) Algorithm terminals.

5. Experimental results and analysis

A mathematical retrieval system based on the proposed method is realized with the searching function of linear algebra expressions in B/S mode with Microsoft Windows server 2003, Microsoft SQL Server 2005 and C#, ASP.net.

5.1. The introduction of the math database

As there is no benchmark dataset for testing the performance of math retrieval [8], we establish an experimental dataset of mathematical expressions with 141714 formulae which contains most of the typical mathematical structures frequently used in science field. There are 3013 expressions of linear algebra structure in the database as shown in Table 1.

Table 1. The linear algebra structure in math database

Type	Matrices	Determinants	The systems of linear equations
Number	2078	762	173

Some mathematical samples in the math database are shown in Table 2.

Table 2. Some samples in math database

Type	Matrixs	LaTeX descriptions	Rows	Columns
Matrixs	$A = \left[\begin{array}{ccc} 2 & -1 & 2 \\ 5 & a & 3 \\ -1 & b & -2 \end{array}\right]$	A = \left[\{\begin{array}{*{20}{c}} 2&{ - 1}&2\\5&a&3\\{ - 1}&b&{ - 2} \end{array}} \right]	3	3
Determinants	$D_3 = \left\|\begin{array}{cccc} 2 & 1 & 8 & 1 \\ 1 & -3 & 9 & -6 \\ 0 & 2 & -5 & 2 \\ 1 & 4 & 0 & 6 \end{array}\right\|$	\[{D_3} = \left| {\begin{array}{*{20}{c}} 2&1&8&1\\1&{ - 3}&9&{ - 6}\\ 0&2&{ - 5}&2\\1&4&0&6 \end{array}} \right|\]	4	4
The systems of linear equations	$\left\{\begin{array}{c} x_1 - 2x_2 + 4x_3 = 0 \\ 2x_1 + x_2 + x_3 = 0 \\ x_1 + x_2 + x_3 = 0 \end{array}\right.$	\[\left\{ {\begin{array}{*{20}{c}} {{x_1} - 2{x_2} + 4{x_3} = 0}\\{2{x_1} + {x_2} + {x_3} = 0}\\{{x_1} + {x_2} + {x_3} = 0} \end{array}} \right.\]	3	1

The efficiency of math indexing and matching is closely related to the scale of the linear algebra expressions. For illustrating the situation of the linear algebra expressions in our experimental database, some statistical data such as the row and column number of every linear algebra expression are shown in Fig. 5. The histogram of the symbol numbers of the maximum component in linear algebra expressions is shown in Fig. 6.

From Fig. 5 we can see that the linear algebra expressions with three rows or three columns take a relative large proportion in our database. Followed by the linear algebra expressions with four or two rows as well as columns.

The symbol number of the components in linear algebra expressions is another factor that effects the performance of math indexing and matching. We count the maximum component from each linear algebra expression which has the most number of symbols including not only operands but also operands. The histogram of the symbol numbers of the maximum component in linear algebra expressions is shown in Fig. 6.

From Fig. 6 we can see that the components which have one or two symbols are the major pattern in linear algebra expressions. Besides, the components with three or four symbols also frequently appear.

The above data of linear algebra expressions shows that the linear algebra expressions have not only complex row-column structure but also relative large scale of symbols, which naturally result in the difficulties of linear algebra expression

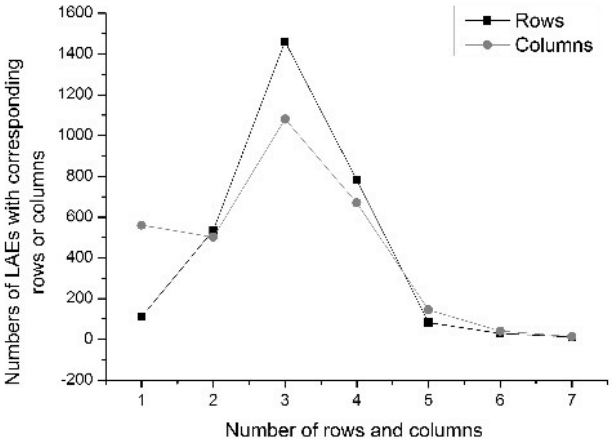

Fig. 5. The histogram of the row and column number of linear algebra expressions

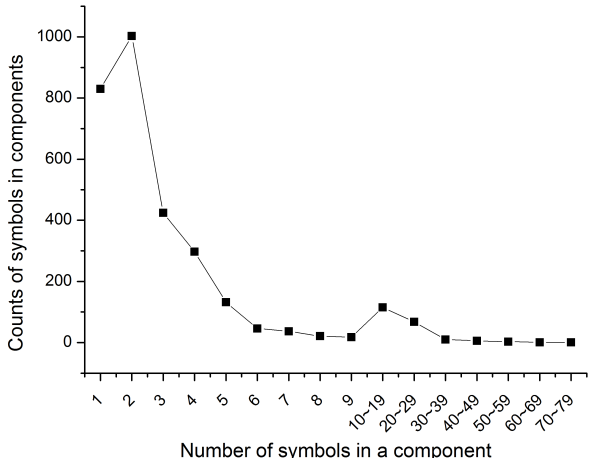

Fig. 6. The symbol number histogram of the maximum component in linear algebra expressions

retrieval.

5.2. Index experiment

The size of index is related to the number of linear algebra expressions. As a linear algebra expression may contain several sub expressions which consist of many symbols, the index size of linear algebra expressions is commonly larger than normal

Table 3. The size of math index

Formula number	10^3	10^4	10^5
Literature [9] (MB)	1.39	13.3	130
Ours (MB)	0.86	7.27	49.6

Table 4. The retrieval time of mathematical expressions

Formula number		10^3			10^4			10^5		
Literature [9] (ms)		8∼147			10∼431			13∼532		
Ours	Query expression size	10	40	70	10	40	70	10	40	70
	Response time(ms)	107	329	470	814	834	850	3061	3184	3336

math indexes. The index size of our method and the method proposed in literature [9] is shown in Table 3.

As a linear algebra expression may contain several sub expressions which consist of many symbols, its index size is commonly larger than normal math indexes.

5.3. Math matching experiment

To test the response time of math query of our system, we select three kinds of linear algebra expressions with different lengths as experimental samples. The retrieval time of mathematical expressions is shown in Table 4. Considering the linear algebra expressions generally have more lengths than normal mathematical formulae as they consist of several sub expressions arranged in row-column structure, the sizes of query expressions are also listed in the table.

When the symbol number of math database and query expressions increases, the response time of our method raises steadily. Because of the special structure and massive symbols of linear algebra expressions, the parameter is acceptable compared with other system.

As the method belongs to the precise matching in symbol level, all expressions according with the query expressions could be obtained through matching operations and no expressions would be wrongly extracted. It is not necessary to calculate the precision rate and recall rate like the traditional full text searching engines.

6. Conclusion

In this paper, a retrieval model of linear algebra expressions is proposed for realizing math information retrieval in scientific field. Focusing on the special structure of linear algebra expressions, a novel feature structure is defined for describing the

row-column structure of expressions. The corresponding indexing and matching algorithms are also designed which could searching the query expressions of linear algebra. The experimental result shows the feasibility of our proposed method. Our future work is to apply the retrieval model into scientific document retrieval through extracting the linear algebra expressions in documents as the math retrieval features with other mathematical expressions.

References

[1] B R. Miller: *Three years of DLMF: Web, math and search.* International Conference on Intelligent Computer Mathematics, 8–12 July 2013, Bath, UK, Intelligent Computer Mathematics, Springer-Verlag Berlin Heidelberg, LNCS *7961* (2013), 288–295.

[2] B R. Miller, A. Youssef: *Technical aspects of the digital library of mathematical functions.* Annals of Mathematics and Artificial Intelligence *38* (2003), Nos. 1–3, 121 to 136.

[3] R. Miner, R. Munavalli: *An approach to mathematical search through query formulation and data normalization.* International Conference Towards Mechanized Mathematical Assistants (MKM), 27–30 June 2007, Hagenberg, Austria, MKM/Calculemus, Springer-Verlag Berlin Heidelberg, LNCS *4573* (2007), 342–355.

[4] P. Libbrecht, E. Melis: *Methods to access and retrieve mathematical content in ActiveMath.* International Congress on Mathematical Software (ICMS), 1–3 September 2006, Castro Urdiales, Spain, Mathematical Software - ICMS 2006, Springer-Verlag Berlin Heidelberg, LNCS *4151* (2006), 331–342.

[5] J. Mišutka, L. Galamboš: *Extending full text search engine for mathematical content.* Masaryk University, Brno (2008), 55–67, Towards Digital Mathematics Library (DML), Birmingham, United Kingdom (2008).

[6] J. Mišutka, L. Galamboš: *System Description: EgoMath2 as a tool for mathematical searching on Wikipedia.org.* International Conference on Intelligent Computer Mathematics, 18–23 July 2011, Bertinoro, Italy, Intelligent Computer Mathematics, Springer-Verlag Berlin Heidelberg, LNCS *6824* (2011), 307–309.

[7] P. Sojka, M. Líška: *Technical aspects of the digital library of mathematical functions.* International Conference on Intelligent Computer Mathematics, 18–23 July 2011, Bertinoro, Italy, Intelligent Computer Mathematics, Springer-Verlag Berlin Heidelberg, LNCS *6824* (2011), 228–243.

[8] X. Hu, L. C. Gao, X. Y. Lin, Z. Tang, X. F. Lin, J. B. Baker: *WikiMirs: A mathematical information retrieval system for wikipedia.* ACM/IEEE-CS joint conference on Digital libraries, 22–26 July 2013, Indianapolis, Indiana, USA, JCDL'13, ACM New York (2013), 11–20.

[9] X. Y. Lin, L. C. Gao, X. Hu, Z. Tang, Y. N. Xiao, X. Z. Liu: *A mathematics retrieval system for formulae in layout presentations.* International ACM SIGIR conference on Research & development in information retrieval, 6–11 July 2014, Gold Coast, Queensland, Australia, SIGIR'14, ACM New York (2014), 697–706.

[10] R. Hambasan, M. Kohlhase, C. C. Prodescu: *MathWebSearch at NTCIR-11.* NTCIR Conference, 9–12 December 2014, Tokyo, Japan (2014), 114–119.

[11] A. Kohlhase: *Math Web Search interfaces and the generation gap of mathematicians.* International Congress on Mathematical Software, 5–9 August 2014, Seoul, South Korea, Mathematical Software, Springer-Verlag Berlin Heidelberg, LNCS *8592* (2014), 586–593.

[12] K. Jing: *Research on math query language and index in web-based math search.* Master's Thesis, Lanzhou University, Lanzhou, China (2009).

[13] Y. X. Xu, W. Su, M. Cheng, Z. Qu, H. Li: *N-gram index structure study for semantic based mathematical formula.* International Conference on Computational Intel-

ligence and Security (CIS), 15–16 November 2014, Kunming, China, IEEE Conference Publications (2014), 293–298.

[14] Springer. http://latexsearch.com/.

[15] X. D. Tian, S. Q. Yang, X. F. Li, F. Yang: *An indexing method of mathematical expression retrieval*. MInternational Conference on Computer Science and Network Technology (ICCSNT), 12–13 October 2014, Dalian, China, IEEE Conference Publications (2013), 574–578.

[16] X. D. Tian, N. Zhou: *A formula retrieval method based on hierarchical structure clusters*. International Conference on Computer Science and Network Technology (ICCSNT), 19–20 December 2015, Harbin, China, IEEE Conference Publications (2015), 173–177.

[17] Y. Liu: *The research about treap data structure*. Computer Application and Software *22* (2005), No. 8, 36–38.

[18] R. Zanibbi: *: Recognition of mathematics notation via computer using baseline structure*. Computing and Information Science, Queen's University Kingston, Ontario, Canada (2000).

[19] R. Rivest: *The MD5 message-digest algorithms*. MIT Laboratory for Computer Science, Request for Comments: 1320, pp. 1–19, 1992.

Received April 30, 2017