CHEMICAL ENGINEERING TRANSACTIONS

VOL. 97, 2022

A publication of

AIDIC

The Italian Association
of Chemical Engineering
Online at www.cetjournal.it

Guest Editors: Jeng Shiun Lim, Nor Alafiza Yunus, Jiří Jaromír Klemeš
Copyright © 2022, AIDIC Servizi S.r.l.
ISBN 978-88-95608-96-9; **ISSN** 2283-9216

DOI: 10.3303/CET2297094

Design of Mathematical Formula Information Retrieval System

Yong Hou

Bengbu University, Anhui, 233030, China
aspnetcs@163.com

The mathematical formula information retrieval system -MFIRS is designed and implemented, and the architecture of the system is discussed. A similarity indexing method based on the mathematical sub-formula of representation MathML is proposed. The system has the characteristics of mathematical perception. The mathreteval dataset was created using more than 4,500,000,000 arXiv documents and 158,106,118 mathematical formulas, and on this dataset, The scalability of the system is verified. The front end of the system uses a web interface that allows users to retrieve complex queries consisting of plain text and mathematical formulas that can be written in TEX or MathML. When a user queries with TEX, the system is able to instantly convert it into a MathML tree representation and index it. The system is a mathematical formula information retrieval engine with mathematical perception characteristics, which can be retrieved by sub-formula similarity and the index of adjacent mathematical formula is realized.

1. Introduction

By searching in a digital library, people can find a lot of what their need. Mainstream search technology is mainly for plain text retrieval, text documents in the form of word bags, do not support mathematical formula processing. Scientific literature is full of indexes, indices, and complex mathematical formulas, even in the basic metadata, titles, and abstracts of papers. Research experience on Google Scholar has shown that not dealing with mathematical formulas in references can lead to serious retrieval problems.

The standard for mathematical exchange between related software tools is W3C's MathML.Few people want to write MathML directly, and people usually prefer some kind of TEX-style compact symbol, such as LATEX or AMSLATEX. As a result. Mathematical retrieval system enables users to use their favorite symbols (such as TEX package or similar (AMS) LATEX) to query, to meet the different retrieval preferences of users, so the data should be converted into a unified format. Represented MathML or content MathML is used only for the output of software systems.

In the process of scientific and technological literature retrieval, the unresolved mathematical retrieval problem becomes very prominent and arouses great interest, because the system that does not support the information retrieval of mathematical formula is not perfect. Therefore, The current popular mathematical retrieval systems are including MathDex(Chan C,2020), MathFind(Gardesten M. 2021), EgoMath(Liu H et al.,2021), Egothor(MD A et al.,2021), LATEXSearch(Perepu P K,2021), LeActiveMath(Shen Y et al.,2021), MathWebSearch(T V. Bakhteeva et al.,2021,),TUW-University of Technology(Zhai J et al.,2022,), et al.

2. Design of the system

The developed system divides the index content into mathematical formula index and ordinary text index when indexing XHTML, HTML and other documents. The indexing methods of the two types of content are different, among which ordinary text indexes are indexed in a conventional manner and using a traditional method.

In Figure 1 below, the overall architecture of the system is described in detail.

Paper Received: 24 September 2022; Revised: 10 November 2022; Accepted: 23 November 2022
Please cite this article as: Hou Y., 2022, Design of Mathematical Formula Information Retrieval System, Chemical Engineering Transactions, 97, 559-564 DOI:10.3303/CET2297094

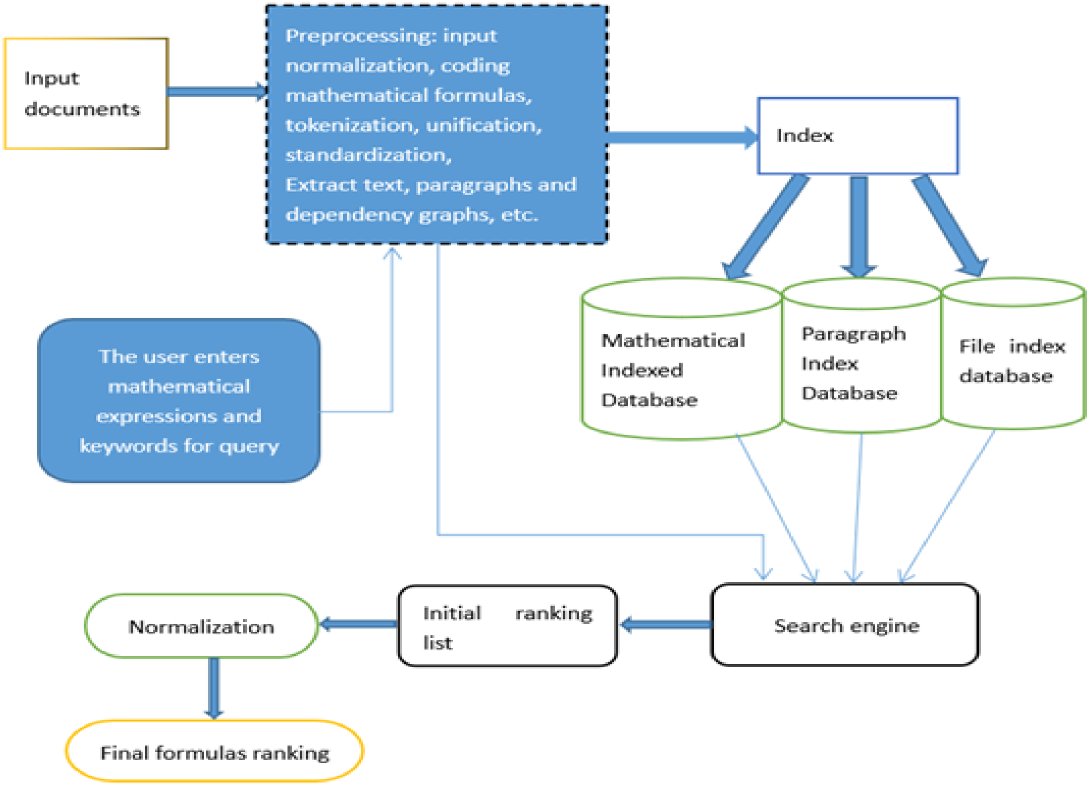
Input documents
Preprocessing: input normalization, coding mathematical formulas, tokenization, unification, standardization, Extract text, paragraphs and dependency graphs, etc.
Index
Mathematical Indexed Database
Paragraph Index Database
File index database
The user enters mathematical expressions and keywords for query
Search engine
Initial ranking list
Normalization
Final formulas ranking

Figure 1. The overall architecture of the system

The index module of the realized system is mainly to normalize the input and preprocess the mathematical formula.

2.1 Input Normalization

The MathML document is normalized using the UMCL toolset to avoid the problem of mathematical formulas with the same semantics represented by different MathML symbols. Standardize MathML in documents through the UMCL toolset.

2.2 Mathematical formula Unification

There are three different types of unified algorithms used by the system. In order to achieve multiple common representations of all formulas, the unified algorithm performs a tokenization process. The system returns match similar to user queries, while retaining formula structures and α equations.

2.3 Coding mathematical formula

The mathematical formula in math format is then used with hash coding techniques and path-based coding techniques. Path-based coding technology splits a mathematical formula into three types of information, including: brother node information for subtrees, ordered path information, and no path information.

2.4 Extract text information and mathematical formulas

Text information is extracted from four level: Body paragraph level; Document level; Math level. The system realizes the overall and partial perception recognition of mathematical formulas,which appear in multiple parts of a document, in the following ways.

Read all documents and use Python's regular expressions to extract, parse, and store the contents of MathML formulas in query documents. The specific steps are as follows.

1).Iterate through each document and find all formulas in this document with regular expression (1).

```
Pattern=re.compile('<m:annotation-xml id=“id\d+”              encoding=“MathML-Content”>(.*?)</m:annotation-xml>',re.S)                                                                                     (1)
```

2).For each formula in the document, use regular expressions (2), (3), (4), (5) to extract the formula ID number, and get the parts of the formula.

```
pattern = re.compile('<m:\w*\sid= "id(\d+)"',re.S)                    (2)
pattern1 = re.compile('^(<m:\w*)?.*?',re.S)                          (3)
pattern2 = re.compile('(.*?)</m',re.S)                               (4)
pattern3 = re.compile’(</m ： \w*)',re.S）                            (5)
```

3).Construct MDG(V,E) diagrams[xx] to identify and capture variations of mathematical formulas.

4).Write the extracted formula to a file and index it.

2.5 Normalization

The goal of normalization is to reduce MathML formula scripts with the same semantic and mathematical structure to only one representation, in order to solve the problem of possible inconsistencies and ambiguities when coding formulas in MathML.
For example, remove the appearance elements and attributes of the presentation MathML. Individual appearance elements that affect the semantics of the formula, such as mathvariants, should remain in all possible formula elements.Unify the fence elements of MathML, trying to replace the mfenced element with the mrow element.Minimize the number of Mrows in MathML and remove redundant Mrow elements.Handle MathML's subscript/superscript elements, try to replace msubsup elements with msub and msup elements, and place msubs in msup.Remove the entity symbol &#x2061 that represents the function and express the function parameters with mrow and parentheses.

2.6 Index math formulas

The index of this system consists of mathematical formulas, paragraph levels index, document levels index, and other index information. The encoded mathematical formula and the corresponding explanatory text form the mathematical index. Hash coding techniques and path coding techniques are used to encode the MathML mathematical formulas that represent format and content type, and then generate the corresponding tables and place them in the index database.

2.8 Retrieval and Ranking

A mathematically aware, full-text-based retrieval and ranking algorithm is implemented. The algorithm is able to process documents that contain mathematical symbols in the presentation MathML format and filters out all unnecessary representation elements as well as any other MathML symbols, such as content-based MathML or other markup symbols. This algorithm allows the user to retrieve mathematical formulas and the text content of the document.
During the retrieval phase, user input is divided into math and text. The formula is then preprocessed in the same way as the index stage, in addition to marketing the participle - the user may also retrieve sub-parts of the query formula, not just the formula as a whole.
Similar to natural language retrieval, an algorithm can retrieve not only the entire formula, but also the individual sub-formulas contained in the formula. For each formula in the input and its sub-formulas, the algorithm is able to create a number of different broad representations of the to allow for similarity retrieval of mathematical formulas. To calculate the relevance of a matching formula to a user query formula, the algorithm uses heuristic weighting of index words to affect the score of the matching document, thus affecting the order of the results. Weight formulas are assigned based on the complexity of the formula, as well as the levels in the input formula tree and the generalization level.
This mathematical formula, which converts the mathematical formula in xml format into the form of compressed linear string, appears at the end of the preprocessing stage.
In the retrieval stage, the final score of the document is related to the weight of the mathematical formula, and the calculation formula is shown in the following(6).

$$score(q,d) = coord(q,d) \cdot queryNorm(q) \cdot \sum_{t \in q} \left(tf(t) \cdot idf(t)^2 \cdot t_{getBoost}() \cdot norm(t,d) \right) \tag{6}$$

3. Experimental evaluations of the system

In this paper, a large-scale evaluation is achieved with the aid of a mathematical text library.

3.1 Dataset

To evaluate this system, a mathematical text library called mathreteval was built specifically. The mathreteval is used to evaluate the performance of this system. First, the arXiv document is converted into content-based XHTML format and presentation-based MathML format, Then, the resulting corpus is decompressed to a size of 260GB and a compressed size of 16 GB.

3.2 Test results

The following tests the ability of the developed system to index and retrieve relatively large real scientific literature repositories. The goal is to observe how the system's index file size, retrieval time, and ranking of retrieved documents are affected by system parameters. Check the scalability of the system. Under different configurations, the evaluation system has both textual and textless retrieval performance.
The entire document set contains 158106118 formulas, and after all pre-processing, the system indexes 2910314146 unique formulas, with an index run time of 1378 minutes (nearly 24 hours), resulting in an index size of approximately 88GB。
Computing resources and experimental parameters. 512GB of RAM, 48-core Intel Xeon CPU, Ubuntu v22.04 operating system.

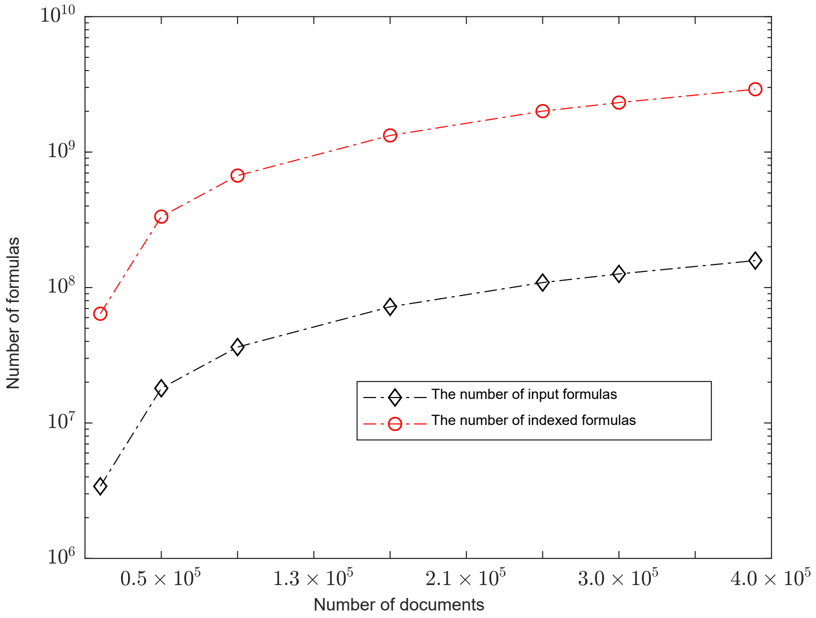

Figure 2 Scalability test results of input documents and indexed formulas

As shown in Figure 2, the scalability of the system is approximately linear with the number of documents. This provides a viable response time even for billions of index sub-formulas, even if small formulas can score matches in most documents. We use different complex queries, such as hybrid queries, non-hybrid queries, high/low complexity single/multi-formula queries, and so on. to create an index, and then measure the average query time of the system on the mathreteval dataset, resulting in an average query time 512 milliseconds.

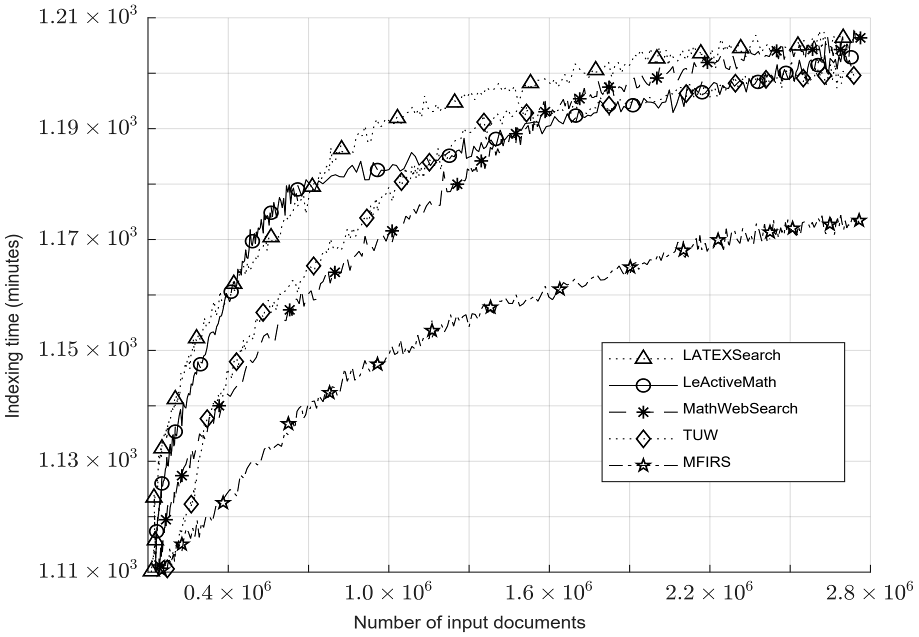

Figure 3 Comparison of index consumption time of various systems

With the increase in the number of input documents, it can be seen from Figure 3 that the index consumption time of each system shows an increasing trend. When the number of input documents is fixed, the index consumption time of the LATEXSerach system is the longest, and the index consumption time of the MFIRS system The shortest time. When the number of input documents reaches 1.46×10^6, the indexing time curves of the LeActiveMath, MathWebSearch and TUW systems appear to overlap. The indexing time of MFIRS shows an approximate linear increase.

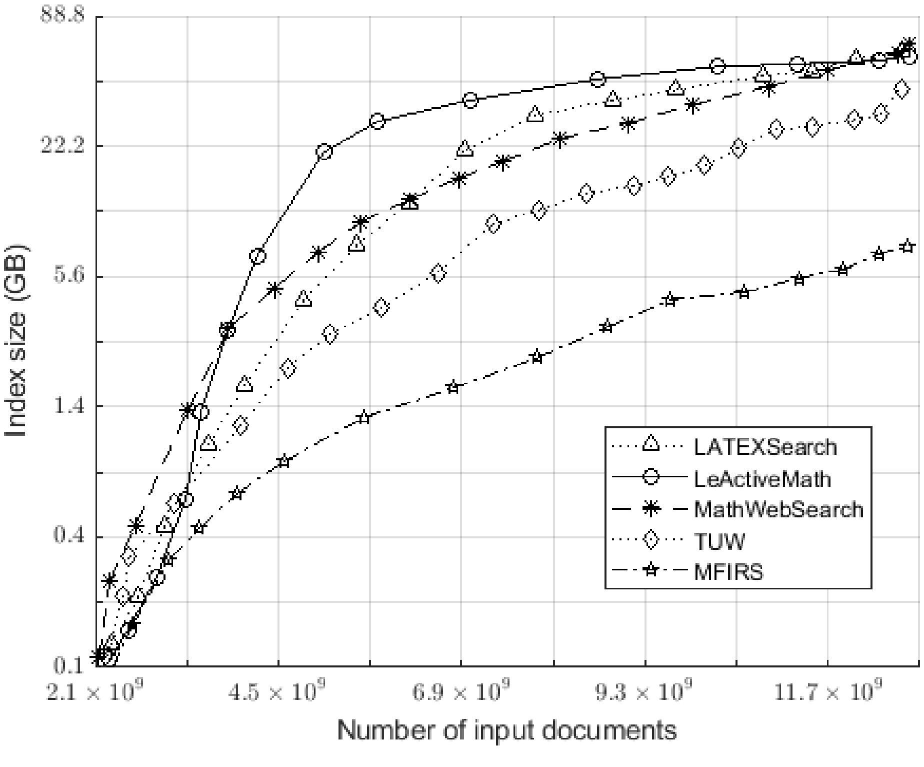

Figure 4 Comparison of index sizes of various systems

With an increase in the number of input documents, Figure 4 shows that the index file size of each system tends to increase. Among them, LeActiveMath has the largest index file, followed by MathWebsearch and LATEXSerach, TUW is in third place, and MFIRS has the smallest index file. When the number of input documents reaches about 12.9×10^9, the index file size of LATEXSerach, LeActiveMath and TUW is almost the same.

4. Conclusions

A method of mathematical retrieval, indexing and sorting is proposed, and it is integrated into the implemented system - mathematical formula information retrieval. The feasibility of this method has been verified on the mathreteval data set, Verification testing of system scalability is based on this.

Future plans are under way to use this technology in digital library projects worldwide. The system's web front end is fully functional and can convert mathematics into presentation MathML for full-text retrieval. The system is well-expanded and has the ability to be used in large digital libraries. The system can not only retrieve mathematical formulas of representational MathML, but also retrieve mathematical formulas of content-based MathML in the future.

Acknowledgments

This paper is supported by the following projects. Anhui Provincial Excellent Talents Training Project (Multi-instance Learning Algorithm Research and Application (gxyq2018107); Key Project of Anhui Provincial Department of Education (KJ2020A0744); Bengbu College High-level Talent Start-up Project (BBXY2018KYQD07).

Reference

Bax C., Sironi S., Capelli L., 2020, How Can Odors Be Measured? An Overview of Methods and Their Applications, Atmosphere, 11, 92

Cassiani M., Bertagni M.B., Marro M., Salizzoni P., 2020, Concentration Fluctuations from Localized Atmospheric Releases - Boundary-Layer Meteorology, 177, 461–510.

Chan C, 2020, stroke extraction for offline handwritten mathematical expression recognition,IEEE Access, 8:61565-61575.

Gardesten M. 2021, Investigating data collection methods for exploring mathematical and relational competencies involved in teaching mathematics, 45(2):21-25.

Liu H, Ko Y C,2021, Cross-Media Intelligent Perception and Retrieval Analysis Application Technology Based on Deep Learning Education, International Journal of Pattern Recognition and Artificial Intelligence, 35:15.

MD A, Aw B, Zg C, 2021, Efficacy, safety and tolerability of formula-based unilateral vs bilateral electroconvulsive therapy in the treatment of major depression: A randomized open label-controlled trial, Journal of Psychiatric Research, 133:52-59.

Perepu P K,2021, OpenMP Implementation of Parallel Longest Common Subsequence Algorithm for Mathematical Expression Retrieval, Parallel Processing Letters, 31:02.

Shen Y, Chen C, Dai Y, 2021, A Hybrid Model Combining Formulae with Keywords for Mathematical Information Retrieval, International Journal of Software Engineering and Knowledge Engineering 31:11n12, 1583-1602.