Journal of Information Science and Engineering 36, 75-89 (2020)
DOI: 10.6688/JISE.202001_36(1).0005

Retrieval of Mathematical Information with Syntactic and Semantic Structure over Web

Sharaf Hussain and Shakeel Khoja
Department of Computer Science
Institute of Business Administration
Karachi, 74400 Pakistan
E-mail: {shussain; skhoja}@iba.edu.pk

Efficient retrieval of mathematical expressions over web is a complex process as compared to simple text search. This is only possible when the syntactic (*e.g.* Textual) and semantic (*e.g.* Structural) information of a mathematical expression is retrieved properly and analyzed methodically. In this paper, we are proposing a technique that indexes expressions along with their syntactic and semantic information. These expressions are represented in Content-MathML(CMML). To improve the memory efficiency in index, an encoding technique is introduced which encode CMML mathematical expressions in Braille Unicode characters. In order to improve ranking of retrieved documents, a weighting function is introduced which assign a weight to each indexing term. The weighting score of each term contributes in ranking function that improves the rank of a document which contains query terms. The proposed technique is evaluated on NTCIR-12 Wikipedia and Arxiv corpora. Performance is also measured using NTCIR-MathIR evaluation criteria. The precision for Wikipedia-formula-queries is achieved 47% and for Arxiv is achieved 44% at top 5 documents.

Keywords: information retrieval, formula retrieval, term ranking, structure matching, term encoding, formula indexing

1. INTRODUCTION

The scientific information available on the World Wide Web (WWW) mostly comprises mathematical equations and expressions. Searching these mathematical expressions using typical web search engines is not only inefficient but also quite difficult. Inefficient, since the web search engines ignore the inherent tree structure of mathematical expressions and render them as text only. Difficult, because no proper interface is provided to input queries involving mathematical expressions. Though some search engines provide the facility to write simple math queries in the text format (*e.g.* `x^2+2x+1=0`) but for the queries with complex structures such as equation $\int_0^{2\pi} e^{sin\theta} d\theta$, a plain text box is merely insufficient [1].

The systems which deal with the searching mathematical expressions from the collection of scientific documents are called Mathematical Search Engines or more formally Mathematical Information Retrieval (MIR) systems. Unlike traditional text retrieval tasks, the retrieval of mathematical expressions is

Received August 18, 2018; revised October 21 & December 15, 2018; accepted January 2, 2019.
Communicated by Chao-Lin Liu.

complex, as the search problem is not just syntactical matching of terms but more a semantic reasoning task. Hence, the retrieval system is required to implement a proper data structure for preserving both textual and contextual information of mathematical expressions so that the more relevant information could be retrieved. The majority of MIR systems use document collections that contain mathematical expressions in LATEX or Mathematics Markup Language (MathML) formats. The LATEX/MathML formats are highly structured and are used in different MIR systems such as Math Web Search [2], EgoMath [3], and Math Indexing and Searching [4] for retrieving mathematical information. A small comparison of these systems is shown in Table 1.

This paper focuses on semantic retrieval of math expressions, memory-efficient indexing, and ranking of retrieved documents. The proposed indexing structure helps in retrieving the true syntactic and semantic (*e.g.* meaning of mathematical expression) results for the query of mathematical expressions. For memory-efficient indexing, we are introducing an encoding technique that reduces the size of indexing terms in order to make smaller index in memory. This paper also discusses a scoring method which assign a weight to each indexing term for improving the rank of retrieved documents.

Section 2 of the paper discusses the theories and approaches in the field of mathematical retrieval and concludes with the discussion of limitations of existing approaches and the need for the proposed solution. Section 3 presents the technical details and the framework of our proposed approach, we will also discuss the data corpus used in the experiments, scoring method of the documents, and crafting a mathematical query. Section 4 is dedicated to technical details and experimental setup, Section 5 discusses the acquired results, and Section 6 concludes the presented work by listing the advantages and limitations of the proposed approach.

2. EXISTING MIR SYSTEMS

In the last few years, many successful attempts have been made to develop efficient MIR systems. As a result, a sizable number of papers has been published in the field of MIR which elaborates the techniques of indexing and retrieval of math formulae. Indexing methods and its data structures are the key elements for retrieving information efficiently, as suggested in Section 1 that researchers use different approaches for indexing including text-based and tree-based for retrieving mathematical information, Table 1 shows the various text-based and tree-based systems.

2.1 Text-Based Indexing

In text-based indexing, mathematical expressions are converted into a sequence of strings and store into the inverted index. Inverted index can be used in any MIR system in which mathematical expressions are converted into text strings using linearization process and tokenized them into sub-strings, these tokens are stored into index along with their location and frequency of their occurrence in the documents. It also provides a fast access to large datasets. Fast query processing can be obtained with an inverted index as it makes use of specific similarity functions *e.g.*, the cosine similarity, which only considers the words and their weights

are present in the query. It is a simple and a fast process for formulae retrieval but ignores the contextual meaning of formulae.

Table 1. Comparison of existing MIR systems.

MIR System	Data Sets	Formula Representation	Retrieval Type	Term Encoding	Ranking Function
Text-Based Indexing					
DLMF	DLMF	MathML	Syntactic	×	×
MathDex	Wikipedia, ArXiv	PMML	Syntactic	×	×
EgoMath	Wikipedia	PMML	Syntactic	×	×
MCAT	ArXiv	CMML	Syntactic + Semantic	×	×
MIaS	ArXiv	PMML	Syntactic	×	✓
OPMES	ArXive,Math StackExchange	LaTeX	Syntactic	×	×
Tree-Based Indexing					
MWS	ArXiv	CMML	Semantic	✓	×
Tangent	ArXiv	CMML	Syntactic	×	×
WikiMir	DLMF, ArXiv, Wikipedia	LaTeX	Syntactic + Semantic	×	✓

2.1.1 DLMF

DLMF uses LaTeX documents for indexing. The collection of documents is converted into parallel MathML for generating a semantic representation of mathematical formulae. DLMF uses three pre-processing steps before indexing known as textualization, flattening, and normalization. DLMF introduces a new data structure for normalization of an equation, called sorted parse tree. It is a tree whose branches of any node are sorted from left to right and it does not alter the mathematical meaning [5].

2.1.2 MathDex

MathDex is math-aware full text search engine [6], which converts retrieved documents into XHTML+MathML. It linearizes math equations into a sequence of text tokens and indexes them into Lucene[1] indexing format. Complex math equations are broken down into simpler sub-expressions and weights are assigned to each sub-expression as per its complexity, length, and nesting depth. The system uses similarity search for query matching.

2.1.3 EgoMath

EgoMath converts various formats (*e.g.* LATEX,TEX, PDF, HTML, *etc.*) of documents into Content MathML(CMML) for making mathematical formulae index. It stores formula and its sub-formulae into an index in post fix notation, each formula is stored in variety of different synonyms, called augmentation. A formula

[1] `http://lucene.apache.org/`

can be found in various format, therefore generalization is applied to all structures and single structure stores into index where commutativity, associativity, and distributivity hold.

2.1.4 MCAT

MCAT is another math-aware full-text search engine. It indexes mathematical formulae written in PMML. Initially unnecessary tags are removed from the mathematical equations, then three types of encoding are used for indexing called ordered path (opath), unordered path (upath), and sisters [7]. It uses Lucene similarity search algorithm for matching query results.

2.1.5 MIaS

MIaS is a Lucene based math-aware full-text search engine [4]. It accepts XHTML documents for making an index. In the pre-processing phase, mathematical formulae are extracted from the documents in PMML format then they are tokenized into a sub-formulae. Canonicalization is used to provide an appropriate order to formulae and sub-formulae [8]. A Unification algorithms is also applied on formulae and sub-formulae. In the indexing process, all formulae are indexed with their original and unified forms. The weight is assigned to formula and its derivatives. The inverted index is used for indexing.

2.1.6 OPMES

OPMES uses symbolic similarity search method for searching documents from an index and uses operator tree for storing LaTeX math expressions. OPMES stores index in two parts; the first part of index stores leaf-root path labels and the second part of index stores mapping of formula ID to additional information of that formula. OPMES can search documents by using similarity-search and structural-search methods [9].

2.2 Tree-Based Indexing

In tree-based indexing method, the index is stored in a tree structure format. A tree structure contains nodes and edges, data items are stored in nodes and edges provide a link between data items. The tree based MIR systems usually store operators and functions in nodes while variables and constants are stored in the leaves of a tree. One of the advantages of tree-based indexing is that it allows the context based search. Following is a description of some Tree-Based MIR systems.

2.2.1 MatWebSearch (MWS)

The MWS uses Substitution-Tree(ST) to perform indexing [10]. In pre-processing phase, XHTML+MathML documents are transformed into a harvest, which contains document ID, formula ID, and formulae in CMML. MWS converts all variables of an equation into generic variables, this process is called unification, these generic variables are inserted into substitution tree [2]. The major advantage of making an index into substitution tree is that the actual terms are replaced with auto-generated substitution symbols, and they are stored at each node, helping in decreasing tree storage space and improving query processing time.

2.2.2 Tangent Math

Tangent indexes math expressions into Symbol Layout Tree (SLT) [11, 12], where symbols are mapped with their layout in order to the occurrence of constants, operands, variables and relationship represented in an expression [13]. Tangent uses Maximum Subtree Similarity (MSS) ranking metric for query-by-expression that produces intuitive rankings of formulae based on their appearance, as represented by the types and relative positions of symbols.

2.2.3 WikiMir

WikiMir is another MIR system which retrieves math formulae from the collection of Wikipedia documents. It extracts math formulae from LaTeX and PDF documents and converts them into PMML format before indexing. It indexes math formulae into a presentation tree [14] and employs inverted index data structure for indexing. WikiMir3.0 uses semantic tree method to store math formulae [15].

2.3 Observations

As discussed earlier, Mathdex works as a normal text search engine, converting MathML expressions into a sequence of text encoded math fragments. It does not consider the context of math expressions and return results on the basis of matching text encoded math fragments. Unlike MathDex, EgoMath represents math expressions into a flatten term and they are stored as text terms in the index. Additionally, in EgoMath2, augmentation and ordering process are implemented on each expression for semantic enrichment, which improves the recall of the system but precision is not up to the mark. Another system DLMF also performs text-based indexing by converting all symbols and operators into text and stores mathematical expressions in a sequence of text tokens. DLMF only indexes TeX / LaTeX documents and is not able to capture the hierarchical structure of mathematical expressions [5]. However, MIaS is a text-based search engine which not only indexes formula and sub-formulae in text format but also ranks the document, OPMES creates an index in two parts occupying huge space in memory. It was also observed that the first-time querying takes longer time than the time of subsequent same queries [9].

In tree-based indexing methods, MWS uses substitution tree for making an index of math expressions. The query retrieval time increases if the tree grows. Tangent Math is an extension of MWS. It segregates expressions by their size, due to which it is difficult to determine how relevant an expression is to a search query simply based on its size. Therefore, the retrieval algorithm can overlook expressions that should be returned as relevant search result [16]. Contrary, WikiMir3.0 creates two indices (*e.g.* formula and context) in the memory, thus occupying large space. The authors also claim that the performance of WikiMir3.0 in terms of context and DCG score is higher than MIaS [15].

3. PROPOSED APPROACH FOR RETRIEVING MATHEMATICAL INFORMATION

It has been observed from the above discussion that MIR systems require further improvements in the following three areas,

1. Preserving semantics (*e.g.* structural information) of math equations

2. Memory management for storing a math index
3. Ranking of retrieval documents

Table 2. Braille equivalent Unicode symbols for CMML elements.

S.No:	CMML Element	Symbol	S.No:	CMML Element	Symbol
1	`<apply>`	⠸⠐⠣	6	`<times>`	⠸⠩⠣
2	`<ci>`	⠸⠉⠣	7	`<divide>`	⠸⠌⠣
3	`<cn>`	⠸⠝⠣	8	`<csymbol>`	⠸⠒⠣
4	`<plus>`	⠸⠮⠣	9	`<subscript>`	⠰⠣
5	`<minus>`	⠸⠤⠣	10	`<superscript>`	⠈⠣

In the proposed approach, we have focused on above mentioned areas and developed new strategies for handling these issues. The Content-MathML(CMML) is only available language that can store the structure of mathematical expression. Therefore, CMML encoded expressions are used for indexing with necessary modification. The objective of using CMML is to provide a precise encoding for the basic meaning of a mathematical expression, rather than any specific rendering of the expression. Standardization, formation of Mathematical Expression Tree(MET), and Generalization are the major processes that are applied on mathematical expressions before indexing.

3.1 Standardization

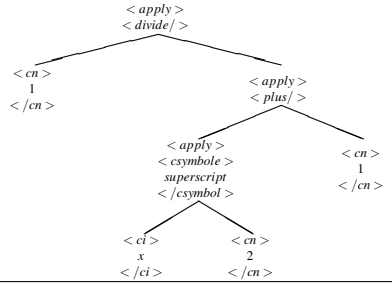
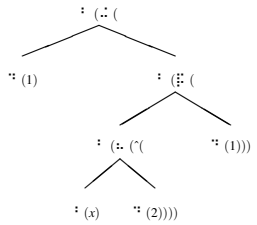
Standardization is a process that converts mathematical expressions into a standard format. Mathematical expressions are made up of numerical constants (numbers), variables, and arithmetical operators. In addition, functions are also very important part of mathematical expressions such as $\sqrt{.}, \sum, and \int$, which are used for performing some specific tasks. A mathematical expression can be written in different formats by using the symbols, operators, functions, and numbers such as $\frac{x}{y}$ can be written in x/y, or xy^{-1} format. In standardization process, multiple forms of an equation are converted into a single standard form that enables searching to retrieve more relevant results.

3.2 Formation of Math Expression Tree (MET)

Mathematical Expression Tree (MET) is a tree structure of mathematical expressions that are encoded in CMML. Before the tree is developed, CMML elements including tags, attributes, and content dictionaries are replaced by Unicode characters preceded with opening elliptical bracket '⠸⠐⠣' and end with closing elliptical bracket '⠸⠐⠜'. The sequence of Unicode character is selected from U+2800 to U+28FF, which are Braille patterns. The Braille patterns are chosen for encoding because they are only used for printing Braille documents and are not used in scientific documents for formulae representation. The result of this conversion produces a concise sequence of Unicode characters as index terms that occupies small space in the memory. As as example, some equivalent Unicode characters for CMML elements are shown in Table 2.

The comparison of CMML and Braille encoded tree (*e.g.* $\frac{1}{1+x^2}$) is shown in Table 3, which clearly shows that Braille encoded tree is more concise and memory efficient.

Table 3. Standardized and generalized trees for $\frac{1}{x^2+1}$.

CMML Encoded Tree	Braille Encoded Tree
	

CMML Encoding: `<apply><divide/><cn>1</cn><apply><plus/><cn>1</cn>`
`<apply><csymbol>superscript</csymbol><ci>x</ci><cn>2</cn></apply>`
`</apply></apply>`

Braille Encoding: ⠸ (⠠⠄ (⠨ (1)⠸ (⠿ (⠸ (⠲ (⠈(⠸ (*x*)⠨ (2))))⠨ (1)))

3.3 Generalization

Generalization of a mathematical expression is a process in which we replace variables, constants, and operators with unified symbols. The generalized tree describes the general structure of an expression which is useful for searching math expressions those have same contextual meaning. The math expressions are called semantically similar, When they are contextually equal. Therefore, generalization helps to search mathematical expressions semantically. The general structure of an equation is called generalized form.

Generalization is divided into two types, simple and extended generalization. In simple generalization, variables, constants, and operators are replaced with Unicode symbols U+25ED (◭), U+25D1 (◑), and U+25CC (◌) respectively. In extended generalization, different variables, constants, and operators are replaced with ◭ $_i$, ◑ $_i$, and ◌ $_i$ symbols respectively, where $i = 1, 2, 3, \ldots, n$. The variables, constants, or operators share the same identification, if they are repeating in an equation.

There are various levels of generalization, in first level only constants are replaced with ◑ symbol, In the second level, variables are replaced with ◭ symbol. In third level both constants and variables are replaced with ◑ and ◭ . In fourth and last level, operators are replaced with ◌ along with ◑ and ◭ . All generalized forms are stored in the inverted index for searching. Simple and extended generalization forms of $\frac{(x+y)^3}{(x^2-y^2)}$ are shown in Table 4.

Table 4. Simple and generalized forms of an equation $\frac{(x+y)^3}{(x^2-y^2)}$.

Form	Simple Generalization	Extended Generalization
1	$\frac{(x+y)^◑}{(x^◑-y^◑)}$	$\frac{(x+y)^◑}{(x^◑-y^◑)}$
2	$\frac{(◭+◭)^3}{(◭^2-◭^2)}$	$\frac{(◭_1+◭_2)^3}{(◭_1^2-◭_2^2)}$
3	$\frac{(◭+◭)^◑}{(◭^◑-◭^◑)}$	$\frac{(◭_1+◭_2)^{◑_1}}{(◭_1^{◑_2}-◭_2^{◑_2})}$
4	$(◭ ◌ ◭)^◑ ◌ (◭^◑ ◌ ◭^◑)$	$(◭_1 ◌_1 ◭_2)^{◑_1} ◌_2 (◭_1^{◑_2} ◌_3 ◭_2^{◑_2})$

3.4 Scoring of MET Terms and Documents Ranking

In order to improve the ranking of retrieval documents, different weights are assigned to MET and its subtrees. These weights are computed on the basis of their level of appearance in the tree structure. The weights are also assigned to each generalized form of a tree and its subtrees. In the proposed system, expression and its generalized forms are stored in the index along with their weights. The weight of an expression is computed by the following formula:

$$W_R = \frac{l^{t_l}\{(1 + w_v|v| + w_c|c| + w_o|o|) \times (|v| + |c| + |o|)\}}{n} \tag{1}$$

A mathematical expression is a phrase that groups together numbers (constant), letters (variables) or their combination joined by operators (+, -, ×, /,^), to represent the value of an entity. In Eq. (1), arbitrary weights are assigned to variables(*e.g.* $w_v = 0.8$), operators(*e.g.* $w_o = 0.6$), and constants (*e.g.* $w_c = 0.5$) as per the importance of these elements in a mathematical expression. The weight for tree level constant (*e.g.* $l = 0.7$) is selected at the higher side because the level of tree or subtree t_l value will reduce it significantly when the tree level increase. The l^{t_l} is a discount factor which decreases the weight of a expression according to its level of appearance in the tree. The product of weights (*e.g.* w_v, w_c, and w_o) with the cardinalities of variables, constants, and operators (*e.g.* $|v|, |c|$, and $|o|$) are called weighting component (*e.g.* $w_v|v| + w_c|c| + w_o|o|$) which determines the weight of an expression in the MET. The sum of cardinalities of terms (*e.g.* $|v| + |c| + |o|$) determines the combined weight of an mathematical expression, this sum produces a gain in the total weight of an expression. The cardinality of a term is unimportant when generalization is performed, therefore the term's cardinality can be eliminated on the basis of generalization scheme. For example, if a mathematical expression is generalizing with respect to variables and constants then W_R formula is rewritten as,

$$W_{vc} = \frac{l^{t_l}\{(1 + w_o|o|) \times (|v| + |c| + |o|)\}}{n} \tag{2}$$

In Eq. (2), cardinalities of variables and constants are eliminated because they have been generalized and at this level are not considered as important entities in the mathematical expression tree (MET). However, these cardinalities of variables and constants are not removed from the combined weight of the term (*e.g.* $(|v| + |c| + |o|)$) because it determines the actual weight of mathematical expression. As per example, the Real (W_R) and Generalized (W_{vc}) weighting scores of mathematical expression $\left(\frac{1}{x^2+1}\right)$ are given in Table 5.

The query is crafted in such a way that it can be retrieved fully and partially matched math expression from the index. It is written in CMML format and is converted it into text by applying pre-processing steps as defined in Section 4. The query is divided into sub-parts of the given mathematical expression. Whole part and subparts of the query are also represented in various generalized forms. The complete query is the composition of whole part, subparts, and generalized forms. For example, if the query is $\frac{1}{1+x^2}$, then after the pre-processing, the query will be represented as;

Query:

: (.: (": (1): (⁝: (:. (^(: (x)": (2))))": (1))) OR

Table 5. Weighting scores of $\left(\frac{1}{x^2+1}\right)$.

Term#	Expression	Generalization	t_l	$\|v\|$	$\|c\|$	$\|o\|$	n	W_R	W_{vc}
1	$\frac{1}{x^2+1}$	$\frac{\blacksquare}{\blacksquare+\blacksquare^{\blacksquare}}$	0	1	2	3	7	3.2285	1.6857
2	x^2+1	$\blacksquare^{\blacksquare}+\blacksquare$	1	1	2	2	5	2.2400	0.9800
3	x^2	$\blacksquare^{\blacksquare}$	2	1	1	1	3	1.0943	0.4573

: (⁞ (: (:. (^(: (x)"; (2))))"; (1))) OR
: (:. (^(: (x)"; (2)))) OR
: (.: ("; (◑): (⁞ (: (:. (^(: (◭)"; (◑))))"; (◑))) OR
: (⁞ (: (:. (^(: (◭)"; (◑))))"; (◑))) OR
: (:. (^(: (◭)"; (◑))))

In above query only variables and constants are generalized.

The Lucene's practical function is used for computing the score of retrieved documents as described in Eq. (3).

$$Score(q,d)=queryNorm(q)\times Coord(q,d)\times\left[\sum_{i=0}^{n}\left(t_f(t_i\in d)\times idf(t_i)^2\times t_i.getBoost()\times norm(t_i,d)\right)\right] \tag{3}$$

$Score(q,d)$ is the relevance score of document d for query q. The $queryNorm(q)$ is the query normalization factor that is used to normalize a query so that results from one query may be compared with the results of other. The Coordination factor $Coord(q,d)$ is a fraction of math terms found in the document and the total number of math terms found in the query. The term $t_f(t\in d)$, computes the frequency of each query term in the document. The $idf(t)$ computes the inverse document frequency of query terms in the retrieved document.

The $t_i.getBoost()$ is a boosting function of a math term that appears in the document. The boosting values are assigned to math terms during the indexing process. In Table 5, weights are assigned to each math term by using Eqs. (1) and (2). The $q.getBoost()$ is a boosting function for math terms which appears in the query. The boosting value is set during the query formation process. Generally the default value of $g.getBoost()$ is set to 1.

The $norm(t_i,d)$ is a function that combines the boost and length factors. The value of field boost set when one field is more important to others. Where as length normalization is computed during indexing on the basis of number of math terms appeared in the document. Finally, the retrieved documents are arranged in descending order according to the score they get during the retrieval process.

The following section discuses the overall architecture and development of our proposed MIR system.

4. SYSTEM DESIGN AND DEVELOPMENT

The suggested system is divided into four phases, data-collection and pre-processing, indexing, querying, and the front-end design & development. Each phase is developed as a separate module. The architecture of a proposed system is shown in Figure 1, also suggesting the flow and storage of data in various modules.

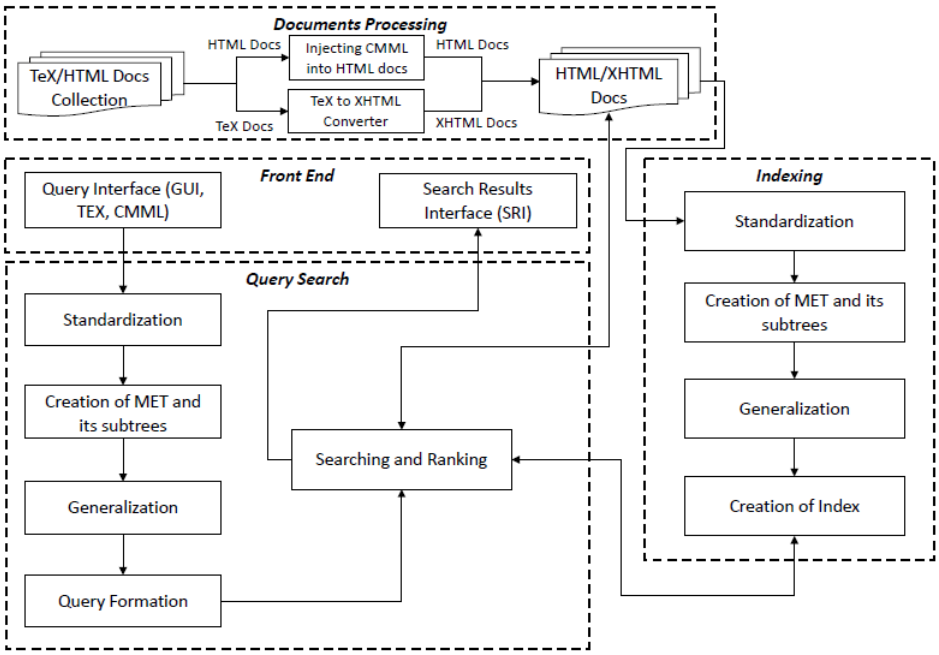


Fig. 1. Architecture of the proposed system.

4.1 Data Collection and Pre-Processing

To perform the preliminary experiments, two data sets, NTCIR-12-MathIR-Wikipedia[2] and ArXiv are used for the creation of mathematical expressions index. The Wikipedia collection contains 64,8414 math expressions in 20,185 documents and occupies 2.2GB of space for an index. The ArXiv's collection contains 154.577 million math expressions in 440,000 documents which occupy 692GB of space for its index in the memory.

In the Pre-processing phase, LATEX and Web documents are converted into Extensible Hypertext Markup Language (XHTML) using *LATEXNLMATH* tool. The Wikipedia HTML documents contains math formulae in PMML and LATEX format but CMML encoding was missing. During the pre-processing of documents, we inject CMML encoding into Wikipedia documents once PMML encoding was carried out, without affecting the originality of documents, the process of CMML injection into Wikipedia documents is shown in Algorithm 1. Once this process was executed, the documents were ready for entering into the indexing phase.

4.2 Indexing

A goal of the proposed indexing technique is to preserve the syntactic and semantic information of math expressions. The indexing involves four sub-modules, standardization, the creation of MET, generalization, and creation of an index.

In the standardization process, multiple forms of a mathematical expression are converted into a single standard form, as discussed in section 3.1. The second

[2]http://ntcir-math.nii.ac.jp/data

Algorithm CMML_Fussion(doc)

Input: Wikipedia document
Output: WikiPedia document augmented with CMML encoding of LaTeX equations
LinePos ← 0;
Regex ← (? <= *displaystyle*).∗(? =} < /*annotation* >);
while *((line=doc.readLine()) !null)* **do**
  if *search(Regex,line)=True* **then**
    LatexEq ← getEq(Regex,line);
    CmmlEq ← LatexToCMML(LatexEq); ▷ Conversion via *latxmlmath*[a]
     AddEq(doc,CmmlEq,LinePos+1);
  end
  LinPos ←LinePos+1;
end

Algorithm 1: Inject CMML encoding into WikiPedia document.

[a] `https://dlmf.nist.gov/LaTeXML/manual/commands/latexmlmath.html`

phase of indexing process is a MET development module. It forms a tree and then extracts math expressions from nodes to leaves, where each path of a node to a leaf represents part of an equation (*i.e.* sub-expression) but root to leaf defines the complete equation. The level of each path from a node to leaf determines the score of expression, which will be used to rank the document during query search. The third module of indexing is generalization which is used to generate generalized forms of mathematical expressions. It is divided into four levels as suggested in Section 3.3.

The mathematical expression tree, sub-trees, and their generalized forms are considered as strings. These strings are stored into an inverted index along with their weights. The inverted index also stores document ID, the title of a document, path of a document, and the location of mathematical expressions in the document.

4.3 Query Processing

Querying is an important component of MIR system which is responsible to get a query from the users and produce results. In the proposed system, users can input queries in CMML, LATEX, or in GUI environment [1]. Internally, the system can accept queries only in CMML, therefore, the LATEX or GUI queries are translates into CMML format before processing. Similarly, the query written in graphical format is first translated into LATEX and then converted into CMML. The input query is converted into a standard format by using the standardization process. The MET and its subtrees are created from a standardized query using the MET module, the generalization procedure is applied on MET and its subtrees. The final query is the logical combination of MET and its subtrees along with all their generalized forms as discussed in Section 3.4.

4.4 Document Ranking

The retrieved documents are ranked according to the similarity matching among mathematical expression trees found in a query and the document. The weight of both generalized and non-generalized trees are contributed to assigning a rank to a document. The rank of a document is computed by the Eq. (4), as discussed in Section 3.4.

5. SYSTEM EVALUATIONS AND RESULTS

In this section, a detailed description of system evaluations and obtained results are provided. The evaluation of MIR system provides an insight to the performance of the different internal component and therefore certain benchmarks evaluation criteria are required. NTCIR provides the evaluation matrices for measuring the performance of IR systems. NTCIREVAL is a toolkit for evaluating various types of IR system including ranked retrieval, diversified ranked retrieval, ranked retrieval evaluation based on equivalence classes, and NTCIR 1CLICK task. In the proposed system, documents are ranked on the basis of their scores in the corpus. Therefore, ranked retrieval is used to measure the performance of the proposed system.

5.1 Results

In order to investigate the performance of the proposed system, we have developed an evaluation framework based on the techniques suggested in NTCIR-12-MathIR Ranked Retrieval Task. Since NTCIR-12 is mainly composed of two corpora *i.e.* The Wikipedia and ArXiv collections, therefore, we have evaluated our system using both Wikipedia and Arxive Math IR collections. The NTCIR-12 provided formula queries to participants to evaluate the system, the Wikipedia Formula Browsing (NTCIR12-MathWikiFormula) subtask contains total 40 queries, out of which 20 queries are without wildcards and remaining queries are with wildcards. The execution of 20 queries without a wildcard were performed. Similarly, the Arxiv Formula Browsing subtask (NTCIR12-Math-queries- participants) containing 29 queries which are the combination of text and formulae were also performed. Since proposed system is based on formula search only, so we only selected Arxiv formula queries for the evaluation.

The evaluations are performed using the software provided by the NTCIR12 (NTCIREVAL[3]) for the evaluations of math IR systems. We have selected different measurements for the evaluations as suggested by the NTCIR12 including Precision (P), nDCG, MSnDCG, Q-measure and Expected Reciprocal Rank (nERR).

Table 6. Evaluations - NTCIR-12 WikiPedia formula queries.

@k	AP	Q-measure	AP	Q	nDCG	MSnDCG	P	nERR
5	0.4932	0.7549	0.4550	0.4171	0.4902	0.4966	0.4700	0.6569
10	0.4932	0.7549	0.3648	0.3390	0.4585	0.4613	0.4000	0.6585
15	0.4932	0.7549	0.3214	0.3071	0.4571	0.4595	0.3733	0.6616
20	0.4932	0.7549	0.2956	0.2904	0.4592	0.4615	0.3600	0.6625
30	0.4932	0.7549	0.2770	0.2920	0.4671	0.4708	0.3317	0.6632
100	0.4932	0.7549	0.3532	0.4274	0.5614	0.5758	0.2050	0.6649
200	0.4932	0.7549	0.4291	0.5877	0.6517	0.6759	0.1445	0.6652
500	0.4932	0.7549	0.4713	0.6941	0.7324	0.7651	0.0771	0.6652
1000	0.4932	0.7549	0.4932	0.7549	0.8016	0.8416	0.0474	0.6652

The Wikipedia queries without wildcards resulted in the nDCG value of 49.02%, the MSnDCG value of 49.66%, Precision values of 47%, the Average Precision value of 45.50%, and nERR value of 65.69% at top 5 documents. All Wikipedia results are shown in Table 6.

[3] `http://research.nii.ac.jp/ntcir/tools/tool-en.html`

The Arxiv queries without text resulted in the nDCG value of 48.38%, the MSnDCG value of 47.88%, Precision values of 44%, the Average Precision value of 44%, and nERR value of 56.20% at top 5 documents. All Arxiv results are shown in Table 7.

Table 7. Evaluations - NTCIR-12 ArXiv formula queries.

@k	AP	Q-measure	AP	Q	nDCG	MSnDCG	P	nERR
5	0.3483	0.4058	0.4400	0.4276	0.4838	0.4788	0.4400	0.5620
10	0.3483	0.4058	0.3099	0.3035	0.4390	0.4300	0.3400	0.5664
15	0.3483	0.4058	0.2557	0.2597	0.4433	0.4360	0.3067	0.5668
20	0.3483	0.4058	0.2690	0.2822	0.4619	0.4573	0.3400	0.5671
30	0.3483	0.4058	0.3171	0.3535	0.5203	0.5235	0.3267	0.5672
100	0.3483	0.4058	0.3261	0.3669	0.5809	0.5903	0.1300	0.5796
200	0.3483	0.4058	0.3434	0.3959	0.6447	0.6614	0.0830	0.5813
500	0.3483	0.4058	0.3477	0.4045	0.6640	0.6826	0.0364	0.5813
1000	0.3483	0.4058	0.3483	0.4058	0.6673	0.6862	0.0186	0.5813

The above results show the performance of the proposed system. Having examined the result of both Wikipedia and Arxiv tasks, it has been observed that the performance of Precision, Average Precision, and Q-measure is maximum at top 5 documents, while the performance of nDCG, MSnDCG, and nERR are maximum at top 1000 documents.

The various MathIR systems participated in the NTCIR12-MathIR competition, their evaluation results are reported in literature [17]. We have also compared our performance with the NTCIR12 systems and have observed that our proposed system is equally good with NTCIR-MathIR system. The comparison of our system with the systems which participated in NTCIR-12 is given in Table 8.

Table 8. Comparison of proposed system with other systems [16].

ArXiv Main Task

MIR System	Precision			
	@5	@10	@15	@20
WikiMir	0.2207	0.1828	0.1609	0.1379
MCAT	0.2552	0.2379	0.2092	0.1845
MIaS	0.1241	0.1345	0.1218	0.1069
Tangent3	0.2552	0.2000	0.1586	0.1345
Proposed System	0.4400	0.3400	0.3067	0.3400

MathWikiFormula Task

MIR System	Precision			
	@5	@10	@15	@20
MCAT	0.4250	0.3350	0.2850	0.2450
Tangent3	0.4300	0.3400	0.2933	0.2450
Proposed System	0.4700	0.4000	0.3733	0.3600

6. CONCLUSION

In the proposed approach, we have worked on semantic retrieval of mathematical information, efficient memory indexing of math equations, and improved ranking of retrieved documents which contain math terms. The semantic retrieval of math information is made possible by introducing a new generalization scheme for math formulae. The mathematical formulae written in CMML are encoded in Braille Unicode characters that occupy small space in memory and make an efficient index. A new scoring technique is developed for assigning a weight to each indexing term, which contributes to a ranking of the retrieved documents.

The performance of the proposed system is satisfactory but it needs further improvements. Currently, the developed system indexes mathematical formulae and do not store text terms from the HTML/XHTML documents. Therefore, a user can search for math information from a formula query. In the future, text tokens will be indexed along with mathematical formulae so that user can write query either in math or text format.

REFERENCES

1. S. Hussain, S. Bai, and S. Khoja, "Efficient applications for mathematical resources on web," in *Proceedings of the 3rd International Conference on Computing for Sustainable Global Development*, 2016, pp. 968-973.
2. M. Kohlhase, B. A. Matican, and C. C. Prodescu, *MathWebSearch 0.5: Scaling an Open Formula Search Engine*, Springer, 2012, pp. 342-357.
3. J. Mišutka and L. Galamboš, "Extending full text search engine for mathematical content," *Towards Digital Mathematics Library*, Birmingham, UK, 2008, pp. 55-67.
4. P. Sojka and M. Liska, "The art of mathematics retrieval," in *Proceedings of ACM Symposium on Document Engineering, Mountain View*, 2011, pp. 57-60.
5. B. R. Miller and A. Youssef, "Technical aspects of the digital library of mathematical functions," *Annals of Mathematics and Artificial Intelligence*, Vol. 38, 2003, pp. 121-136.
6. R. Munavalli and R. Miner, "Mathfind: A math-aware search engine," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 735-735.
7. G. Y. Kristianto, G. Topic, F. Ho, and A. Aizawa, "The MCAT math retrieval system for NTCIR-11 math track," in *Proceedings of the 11th Conference on Evaluation of Information Access Technologies*, 2014, pp. 102-126.
8. D. Formánek, M. Líška, M. Růžička, and P. Sojka, "Normalization of digital mathematics library content mathml canonicalization," in *Joint Proceedings of the 24th Workshop on OpenMath and the 7th Workshop on Mathematical User Interfaces*, 2012, pp. 91-103.
9. W. Zhong and H. Fang, "OPMES: A similarity search engine for mathematical content," in *Proceedings of the 38th European Conference on Information Retrieval Research*, 2016, pp. 849-852.
10. P. Graf, "Substitution tree indexing," in *Proceedings of the 6th International Conference on Rewriting Techniques and Applications*, 1995, pp. 117-131.
11. N. Pattaniyil and R. Zanibbi, "Combining TF-IDF text retrieval with an inverted index over symbol pairs in math expressions: The tangent math search

engine at NTCIR 2014," in *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*, 2014.
12. K. Davila and R. Zanibbi, "Layout and Semantics: Combining representations for mathematical formula search," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 1165-1168.
13. R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions," *IJDAR*, Vol. 15, 2012, pp. 331-357.
14. X. Hu, L. Gao, X. Lin, Z. Tang, X. Lin, and J. B. Baker, "Wikimirs: A mathematical information retrieval system for wikipedia," in *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2013, pp. 11-20.
15. Y. Wang, L. Gao, S. Wang, Z. Tang, X. Liu, and K. Yuan, "Wikimirs 3.0: A hybrid MIR system based on the context, structure and importance of formulae in a document," in *Proceedings of the 15th ACM/IEEE-CE Joint Conference on Digital Libraries*, 2015, pp. 173-182.
16. R. Zanibbi, K. Davila, A. Kane, and F. W. Tompa, "The tangent search engine: Improved similarity metrics and scalability for math formula search," *CoRR*, Vol. abs/1507.06235, 2015.
17. R. Zanibbi, A. Aizawa, M. Kohlhase, I. Ounis, G. Topic, and K. Davila, "NTCIR-12 mathir task overview," in *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, 2016.

Sharaf Hussain is pursuing his Ph.D. in the field Computer Science from the Institute of Business Administration (IBA), Karachi, Pakistan. His core research area includes information retrieval in general and mathematical information retrieval on web in particular.

Dr Shakeel Khoja is a Professor at Faculty of Computer Science, IBA and a Commonwealth Academic Fellow. He read for his Ph.D. and Postdoc at School of Electronics and Computer Science, University of Southampton, UK. His research work includes the development of E-learning frameworks, digital libraries, content and concept based browsing, and application of multimedia tools over the web.