APTIKOM Journal on Computer Science and Information Technologies
Vol. 4, No. 1, 2019, pp. 45~56
ISSN: 2528-2417
DOI: 10.11591/APTIKOM.J.CSIT.135

Mathematical document retrieval system based on signature hashing

Sourish Dhar*, Sudipta Roy
Department of Computer Science and Engineering, Assam University, India
*Corresponding author, e-mail: dsourish80@gmail.com

Abstract

Scientific documents and magazines involve large number of mathematical expressions and formulas along with text. The continuous growth of such documents necessitates the requirement of developing specialized tools and techniques, which could handle and analyse mathematical expressions and formulas. Mathematical expressions and formulae are highly structured and quite different from traditional text. Due to which conventional text retrieval system performs poorly in retrieving scientific documents based on mathematical expression formulated as a query. Mathematical information retrieval is concerned with finding information in documents that include mathematics. To address the challenges posed by mathematical formulae as compared to text, this paper aims to construct a math aware search engine, which can retrieve relevant scientific documents based on a mathematical query. A novel signature based hashing scheme to index raw mathematical web documents is proposed in this paper, which can also take mathematical notational equivalences into account. The proposed system demonstrates better precision and stability of the ranked results when compared with other related state-of-the-art math aware search engines.

Keywords*: formula search engine, mathematical information retrieval, MathML, signature hashing, structure encoded strings*

1. Introduction

Mathematics is a very important constituent in the domain of Science, Technology, Engineering and Mathematics (STEM). Its very need is felt in different spheres of research, education and industries. There would be a seldom scientific document without a single mathematical expression (ME)/symbol. In this digital era, with more and more scientific documents being generated, information explosion indeed was inevitable. To store, manage and retrieve this vast amount of scientific documents thereby mathematical expressions novel strategies, principles and tools were developed in the last decade.

The domain of information retrieval (IR) began from early 1950; as a result many IR models are into existence now namely Boolean Model, Vector Space Model (VSM), Probabilistic model etc. However, vector representation does not consider the ordering of words in a document that is a crucial factor for MEs and exact matching may retrieve too few or too many documents [1-2]. The field of IR has been exhaustively explored for many decades but a distinct focus is required for Mathematical Information Retrieval (MIR) because conventional text retrieval systems are not suitable for retrieving mathematical expressions [3-4].

As stated in [5] "Mathematical Information Retrieval is concerned with finding information in documents that include mathematics. This is important for technical disciplines that use math frequently. (e.g. Physics and Computer Science). Mathematical Information Retrieval (MIR) systems are formula based search engine. User information needs requires careful investigation and good understanding to develop firm principles and foundations in the area of MIR systems."

The order of the terms in a mathematical expression (ME) is crucial issue which influence the semantics of a ME but presently in most of the existing text-based MIR sytems bag-of-words approach have been implemented as a result the order of the terms consequently, structure of a ME get lost. Furthermore, with the aforementioned approach most of the MIR systems have used inverted index with tf-idf ranking. Therefore, this paper proposes an alternative indexing scheme i.e. signature based hash index for mathematical information retrieval while constructing a math-aware search engine: SigMa. Moreover, we also extend the concept of structure-encoded strings (SES) for MathML documents to eliminate extraneous sysmbols like <mi>, <mo> etc. without losing the structure of a ME.

Received December 28, 2018; Revised February 20, 2019; Accepted February 26, 2019

1.1. Background

Classically information retrieval (IR) models can be classified into three broad categories namely set-theoretic, algebraic and probabilistic models [1, 6].

1.1.1. Set Theoretic Model

Documents are modeled as sets depending on the terms that it contains. Thereafter, the standard set-theoretic operations are used to derive the similarities. Based on the foundations of set theory and boolean algebra, Standard Boolean Model was derived where connectives like ^, _, ¬ etc. are used to issue the query in conjunction with the key terms [7]. Although being a very simple and efficient model to implement, it also has some limitations. Firstly, it fails to retrieve results with partial match and secondly general users find it very difficult to form complex queries. Due to these reasons, its performance results in either high precision and low recall or low precision and high recall. The strict Boolean and fuzzy-set models are preferable to other models in terms of computational requirements [8].

1.1.2. Algebraic Model

Documents are modeled as vectors, matrices or tuples. The similarity measure here is obtained as a scalar value while document and query terms are represented as vectors. The popular vector space model falls under this category. In an abstract way, the model is based on the notion that important terms convey the meaning of the document. For calculating the weight of the terms, there are two features, which are widely used namely term frequency and inverse document frequency [9].

1.1.3. Probabilistic Model

In this model, the notion of relevance is captured under probabilistic framework as described in [6, 8, 9]. In other words, this model tries to answer the probability of document d_j to be relevant, for a given query q_i. This model is based on a concrete mathematical foundation of probability and also considers term dependence, relationships, weight of the query terms etc. This model is built on a concrete mathematical foundation and also considers the feature of term dependence but the model has many variations depending on many assumptions. Another substantial problem with this model is that it is very hard to implement this model for large-scale information retrieval systems like web search.

One of the fundamental variance between text and mathematical expressions (ME) lies in their encoding schemes and formats. There are several encoding schemes available for mathematical expressions like MathML [10], LATEX [11] and Openmath [12] to name a few. Figure 1 provides the representation of mathematical expressions in different encoding schemes adapted from [2].

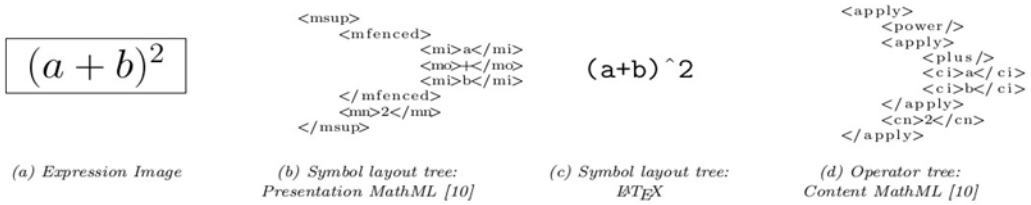

Figure 1. Different encoding schemes for the mathematical expression (a^2+b^2)

Moreover, a mathematical notation is quite inconsistent, and symbol set is limited. A notation is commonly reused, and there often exist several different ways of writing down the same core meaning [13]. For example,

$$\frac{\partial f(x)}{\partial t} \text{ and } \frac{\partial}{\partial t} f(x)$$

Like text, ME's also exhibit the property of polysemy. For instance, the Greek letter α (alpha) could be a Sommerfeld's constant in physics, dominant animal or human in zoology, the brightest star in a constellation in astronomy etc. that makes it ambiguous. Furthermore, using different variables, constants or symbols may result numerous ways to write an implicitly equivalent mathematical expression like

a^2+b^2 vs. $\alpha^2+\beta^2$ which demonstrate the property of synonymy. Normalization is a process to reduce mismatch among the expressions that are semantically similar in nature along with the reduction in index size [2, 13].

Indexing is another major concern in the field of mathematical information retrieval systems (MIR)/math search engines (MSE). Broadly, there are two breeds of MIR systems based on indexing scheme namely text-based and tree based. In text based MIR systems, the emphasis lies on constructing a plain text representation of mathematical expression/formula. Thereafter, it employs several popular information retrieval frameworks like Lucene, Solr etc. to accomplish the task of indexing in an automated way. However, the text representation of the mathematical expression results in either complete or partial loss of structure of the equation [14]. For instance, to extract the feature vectors a clustering technique combined with regular expression was proposed in [15] while [16] used finite state automata to accomplish the task. Similarly Miner et. al. proposed MathDex [17] which uses the text, based n-gram indexing but does not consider several fundamental mathematical equivalences [18].

LaTeXSearch [19] provided by Springer supports LATEX and text queries to retrieve documents from their database while SearchOnMath [20] a part of Microsoft BizSpark program now, considered five math contained datasets namely English version of Wikipedia, Wolfram Math Word, DLMF, Socratic and Planet Math for indexing and retrieval task. The indexing schemes of both the engines are not available as they are proprietary product. EgoMath [21] uses a reverse polish notation to store a mathematical formula and uses augmentation algorithm by applying transformation and generalization rules together with an ordering algorithm on the input. All these systems although presents high recall but precision level need substantial efforts.

On the other hand, in tree based systems trees and variants of trees like tries/substitution trees are employed where leaves of tree points to the expressions and the posting list. These trees are generally inspired from the automatic theorem proving data structures. The benefit of this approach is structure of the mathematical expression/ formulae and each attribute of mathematical representation is arranged in a well-structured manner and retrieval is quite fast. For e.g. MathWebSearch [22] forms a substitution tree of each substructure for semantic representation of formulae. It can work for exact and similar matching by backtracking of substitution tree. A similar approach of substitution tree was proposed by Schellenberg et. al. [23] depending on the layout of the mathematical expression for indexing and retrieval purposes. MIaS [24] also follows the same principle for indexing its documents while creating a separate tree for each substructure of a single mathematical formulae structure, which increases recall of the system but makes it more useful in a broad scale of real world applications. While WikiMirs 2.0 [25] considers only formula information but WikiMirs 3.0 [26] also added a context index. The basic system is based on LATEX markups extracted from Wikipedia dataset. Although these systems offer very high precision but system suffers from low recall.

This paper constructs a math aware search engine with an an alternative approach for indexing that is based on signature hashing along with the implementation of structure-encoded strings for mathematical expressions extended for MathML documents. The reason to use an alternative approach was motivated by the fact that most of the systems disucssed above have used a bag-of words approach along with tf-idf scores . The major bottleneck with this approach is the loss of order, thereby the whole structure which is a crucial aspect of a ME. Most of the math aware systems discussed in this section were either academic prototypes which are inactive as per their current status or propeitary products. Hence, to compare our system we have considered MIaS and WikiMirs because of their availability and are closely related to our approach.

2. RESEARCH METHOD

Typically a document $d_i \in D$ $(Document\ Collection)$ can be represented as m-dimensional feature vector. Similarly a query $q_j \in Q$ $(Query\ Collection)$ can also be represented as a vector. A similarity coefficient can be measured between the two documents using a function (d_i, q_j), which associates a score (real number) to a document. This score generally lies in the range of [0, 1] representing no similarity if 0 or exact match if 1. But searching an m-dimensional feature vector cannot better *O(D)*. However, a hash based indexing scheme can overcome this difficulty as it can easily determine whether or not d_i is a member of D in constant time [27]. The central notion of this scheme is to maximize the probability of collision for similar mathematical structures.

The workflow of the proposed system: SigMa is shown in Figure 2 can be divided into two phases namely: off-line phase for constructing the index and on-line phase for retrieval as the user issues

a query in LATEX that is processed and searched inn the index. Thereafter, results are displayed according to their score in descending order.

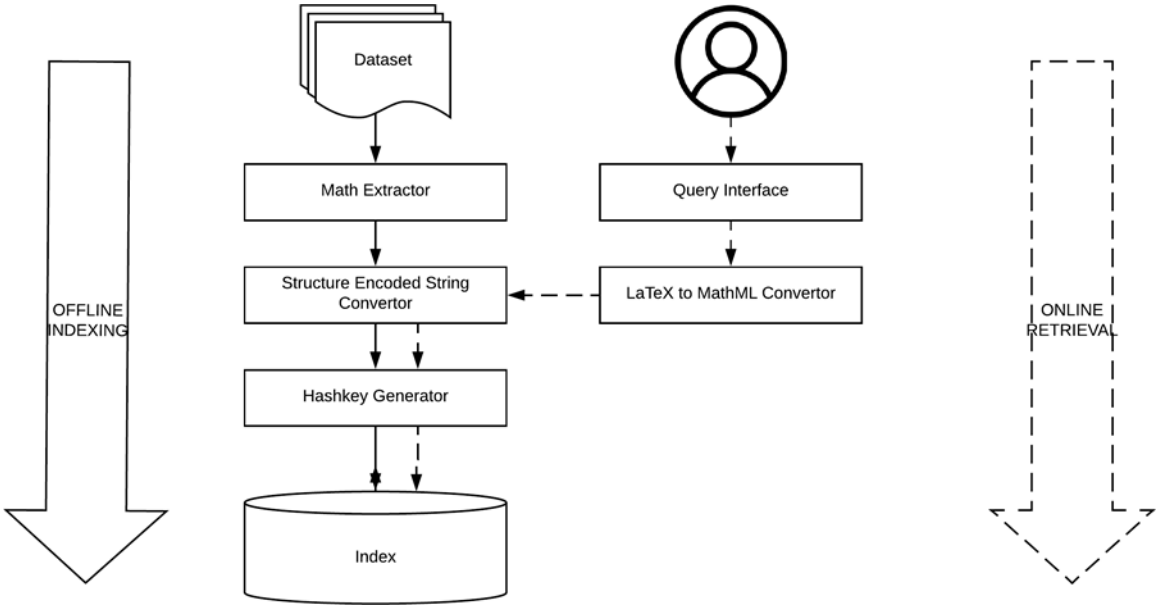

Figure 2. Workflow of the proposed system: SigMa (Solid lines represent off-line phase; dotted line represents on-line phase)

2.1. Datset Description

NTCIR-12 MathIR Task Wikipedia Corpus (version 0. 2. 1) is considered for this research work. It contains mathematical formulas written for normal users. The corpus is publicly available at http://ntcir-math. nii. ac. jp/. Wikipedia corpus contains 319,689 articles from English Wikipedia converted into simpler XHTML format with images removed. There are around 31,839 MathTag articles, which is approximately 10% of the collection approximately, and 287,850 Text articles, which contribute 90% of the collection approximately. There are around 590,000 formulas in this corpus encoded using presentation and content MathML. With the prefix *wpmath* or *wp*, the corpus has been divided into 160 parts containing around 2000 articles approximately in each of the sub-directories.

Each file is annotated with an unique identifier after translating all the formulae into MathML that appears as a <math >tag. Annotation of each file follows the convention i. e. name of the file, followed by the relative offset of the formula in the file, e.g. *id="FileName:0"* for the first formula in *FileName. html*. LaTeXML (http://dlmf. nist. gov/LaTeXML/) is used to convert each formula from LaTeX to MathML, producing three representations for each formula:

a. Presentation MathML: It is used to specify the layout and the appearance of the formula.
b. Content MathML: LaTeXML provides an operator tree representation for the semantics of an mathematical expression.
c. LATEX String: It specifies the symbol layout of the formula using LATEX representation.
d. The size of the corpus in uncompressed form is 5. 15 GB.

The query set for the purpose was downloaded along with necessary relevance judgments. The query set is presented in JSON format, which is composed of with approximately 100 queries. Each query contains a query string in LATEX along with list of labels containing the URL and its score.

2.2. Off-line Phase

In this phase the raw data goes to preprocessing stages and index is created using signature based hashing scheme without any intervention from the user. This step is necessary for fast retrieval of the documents. We have considered P-MML as our source input document format. The Off-line Phase has following modules:

a. Math Extractor

b. Structure Encoded String Generator
c. Hash-key Generator and Index creation

2.1.1. Math Extractor

This module parses and extracts all the mathematical expressions from documents of our data set. We have considered Presentation MathML (P-MML) as our primary supported format. MathML, a W3C standard, is used for representation of mathematical formulae [28].
Following assumptions are made during pre-processing stage of the document.

a. Mathematical text and space are not considered, so <mtext > along with <mspace > and <ms > elements are eliminated.
b. MathML elements which contributes mostly towards appearance or styling information with a very less or no consideration for the content and semantics are not considered. Hence, <mstyle >, <merror >, <mpadded>, <mphantom >, <mlabeledtr > and <menclose > are eliminated.
c. Tensors are not considered in this system, it may be incorporated in our subsequent version as tensors could be represented in many ways. So <mmultiscripts > are removed.
d. Similiar to pre-processing stages as described in [10, 40], Elementary Math Layout and Enlivening Expressions are completely ignored for the simple reason as these elements are generally used for grouping, binding actions or alignment purposes.

Next, the source document is segregated into two parts: math-text for mathematical content and body-text for other textual content present in the document apart from mathematics.

2.1.2. Structure Encoded String Generator

In this module, we have adopted and extended the work reported in [29]. The authors have addressed the problem of an automated performance evaluation of Mathematical Expression (ME) recognition and proposed a novel way to convert a Mathematical Expression (ME) that may be non-linear in nature into a Structure Encoded String (SES) which is linear representation without losing structure of ME's spatial relationships like superscripts, subscripts etc. Their work was based on LATEX input. According to their hypothesis, any symbol in a ME is spatially associated with six surrounding positions namely top-left(TL), above(A), top-right(TR), bottom-left(BL), below(B) and bottom-right(BR). Moreover, the entire top region constitutes single sub expression as northern region represented as N and similarly bottom region as southern region represented as S. Here, mathematical symbol M known as base of the expression. The concept is illustrated in the Figure 3(a) and 3(b).

(a) (b)

Figure 3. (a) A sample mathematical formula (b) Describes the six spatial regions for any mathematical symbol

Considering Figure 3(a) and 3(b) "a, b, c, +,=" represents base mathematical symbol (M) and superscript "2" which is in the northern region represents top right (TR). So, the Structure Encoded String (SES) of the Pythagoras formula $a^2 + b^2 = c^2$ will be

< *aNS2NE + bNS2NE=cNS2NE* >

Here, NS represents start of the northern region and NE is designated to mark the end of the northern region. After extracting the mathematical expressions from the documents, we generate equivalent SES for further processing. Scanning the Presentation MathML (P-MML) markup from <math> to <\math> generates SES. Furthermore, two special set of structure symbols i. e Ns and Ne (Ss and Se) are used to preserve structural information of ME. Here, Ns stand for North start and Ne for North end and similarly Ss and Se are designated for southern region subexpression.

Therefore, by using this approach we can convert mathematical expressions into structure encoded string, thereby making expressions linear without losing any structural information. The approach could easily be expanded for other formats like content MathML, chemical structures etc. A complete list of other structural symbols used in the algorithm is given in Table 1.

Table 1. The complete list of structural symbols

Sl. No.	Description	Symbol
1.	SOUTH START: For capturing start of the southern sub-expression	SS
2.	SOUTH END: For marking end of the southern sub-expression	SE
3.	NORTH START: For capturing start of the northern sub-expression	NS
4.	NORTH END: For marking end of the northern sub-expression	NE
5.	NEW ROW: For capturing new row in the table or fractions	@
6.	DENOMINATOR/FRACTIONS:For capturing fractions / denominators	/
7.	BOTTOM START: For capturing start of the bottom sub-expression	BS
8.	BOTTOM END: For marking end of the bottom sub-expression	BE
9.	MATRIX START: It marks the start of a matrix row	MTS
10.	MATRIX END It marks the end of a matrix	ME
11.	ROOT: For all kinds of roots	RT
12.	PARENTHESIS START: Self-explanatory	(
13.	PARENTHESIS END: Self-explanatory	)
14.	TOP START: For capturing start of the above sub-expression	TS
15.	TOP END: For marking end of the above sub-expression	TE

2.1.4. Hash-key Generator and Index creation

As reported in [30] a hash function f (x) maps a set of keywords into an integer interval from 1 to n. "A signatures is defined as a sequence of w bits created to represent the data contained in each document in a collection. The signature for a document is created by hashing each term to a w string, and OR'ing each of these bit strings together" [31].

Subsequently, query processing also takes the same route by creating a query signature first, thereby comparing the signature in the collection [32]. Document signatures are associated to a bit vector which may take value 0 when there is no match for a particular symbol and 1 when there is a match for a particular symbol. It is based on a fairly obvious representation of the "structure" of the word as a bit word, used as a hash (signature) in the hash table. In the process of search by keyword w, the system successively computes all the signatures and finds those in which the component f (w) equals 1. Only these documents may contain the keyword w, and they are sequentially scanned for matches.

As per [33] it can be formally defined as: " The signature sign (w) of the word w is an m-dimensional vector whose kth element equals 1 if the word w contains the symbol a such that f (a)=k and zero otherwise. "The signature number of a word is given by:

$$\mathrm{H(W)} = \sum_{i=0}^{m-1} \binom{n}{k} 2^i sign(w)_{i+1} \quad (1)$$

While indexing, we calculate hashes for each signature generated through the documents i. e. SES. This SES along with its doc_id is added in the corresponding hash table row, which we construct during the process. We also created an empty bit vector (size=12) and a mapping table containing 12 classes of mathematical operators and symbols to create the bit vector. For instance, the generated SES i. e. ajNSj2jNEj+jbjNSj2jNEj=jcjNSj2jNEj which represents the formula: $a^2+b^2=c^2$ is encoded into a bit vector: 100000001110.

The hash computing process for each bit of the hash, symbols from the SES is matched with the mapping table. Bit 1 at position i in the hash means that there is a true matching of the ith set from the mapping table. Finally, a complete signature hash table is generated. For handling collision problem, we have used chaining method that allows many items to exist at the same location in the hash table by holding a reference to a collection (or chain) of items. The central idea is that similar SES will yield a similar bit vector and subsequently will be hashed in the same location. The complete process is illustrated in the Figure 4.

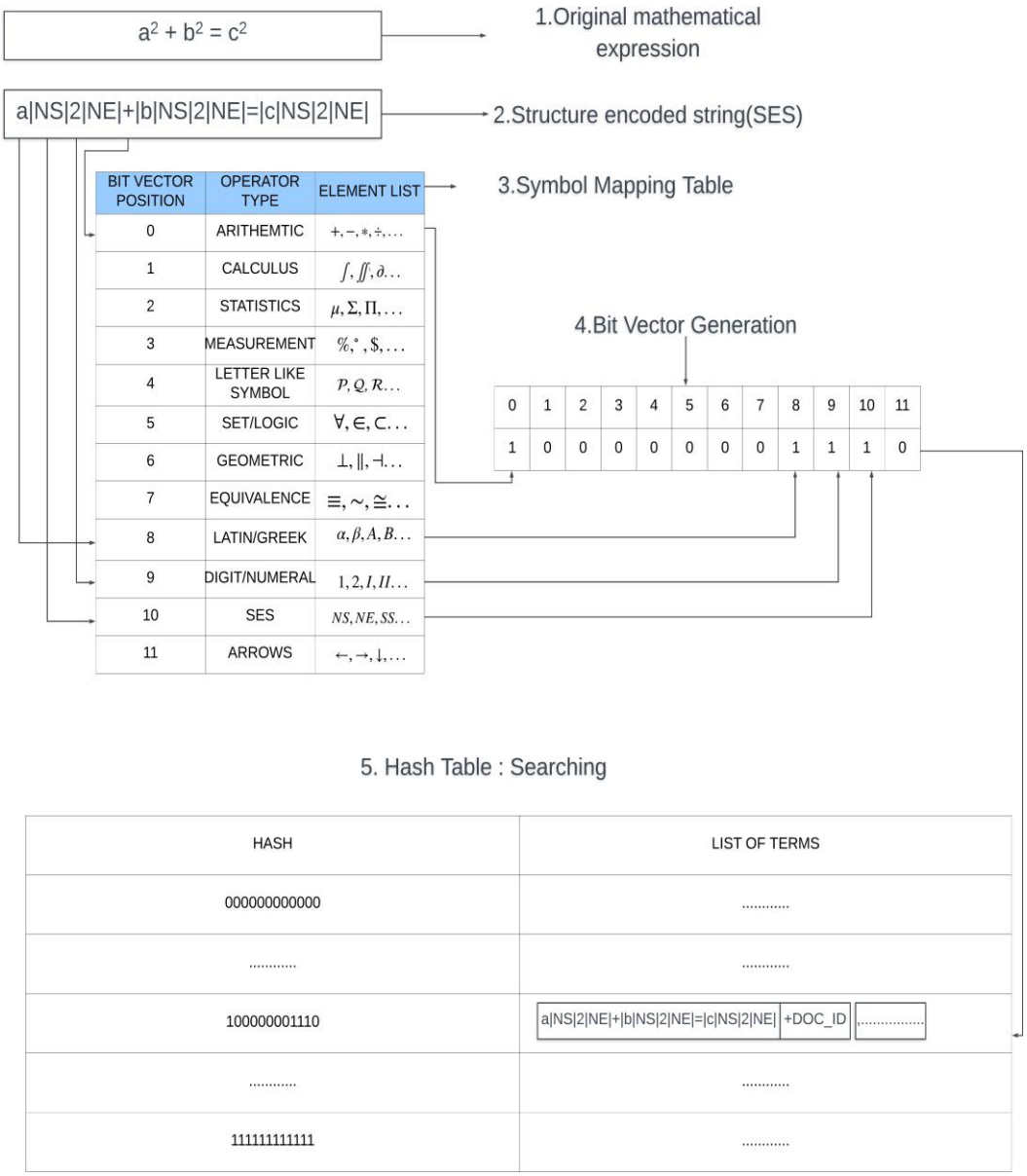

Figure 4. Process of generating bit vectors (signatures) and searching

2.2. Online Retrieval Phase

In the on-line phase, a LATEX query string is considered as input. This LATEX query string is converted to P-MML on the fly. This P-MML again goes through SES converter module and hash key generator of the index module generates a signature file for the query .

2.2.1. Matcher and Ranker

The proposed approach uses Jaccard distance [34, 35] for matching query and index database. This model is used to calculate the similarity between two sets A and B given by the following expression:

$$SCORE=|A\cap B|/|A\cup B|$$

The numerator represents the commonality between A and B, and the denominator represents the union of A and B. The Jaccard distance implementation operates at a token level, where we compare the SES by first tokenizing them and then dividing the number of tokens shared by the SES in the chain once

a match is found in our hash table. After that we retrieve top k documents in descending order based on their score. If two or more documents gets the same rank, they are ordered on first come first serve basis.

2.2.2. Pseudocode for Signature Hash Index creation and Searching

Input: < int > HashTable[size], vector < string > bitvector
Output:List: resultset(Sorted)
local List: PostingList,< int > index,< int > j
comment: Initial HashTable size=40 and load factor=0.75
Function: Insert(bitvector)
for each bitvector
do
comment: Compute the index using hash function given in equation (1)
index=hash f unction(bitvector)
if (index.exists)
then PostingList.push(DocId, SES)
else
newPostingList()
PostingList.push(DocId, SES) Function: Search(String query)
Compute the bit vector o f the query
Compute the index by using the hash function
index=hash f unction(query)
comment: Search the posting list at that specific index
for each j 2 HashTable[index].size()
do
if (HashTable[index][j].contains(query))
then
Compute similarity score using Jaccard Measure
Sort the resultset
return (resultset)
else
return (NULL)

3. RESULTS AND ANALYSIS

We evaluated our system using the following evaluation measures:

a. Precision It measures the exactness of the retrieval process [9, 36]. If I denote the actual set of relevant document and O denotes the retrieved set of document, then the precision is given by:

PRECISION=|I∩O|/|O|

b. Discounted Cumulative Gain (DCG) DCG measures the usefulness, or gain, of a document based on its position in the result list [1,37]. DCG of the top-k retrieved results can be calculated using:

$$DCG_k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2 (i + 1)}$$

Here, the list is named rel in which the i-th element (reli) denotes whether the i-th retrieved formula is relevant to the query (reli=1) or not (reli=0).

We have taken LATEX representations of mathematical equations as query with a query id 1, 2, 3… as shown in Table 2. For each query we have retrieved the top 10 results (documents) on the basis of score. We have considered three state-of-the-art MIR systems namely MIaS, WikiMirs 1 and WikiMirs 2 to compare our results. The precison@10 is calculated and a comparative analysis for 25 queries is shown in the Figure 5. We have also calculated DCG for each system based on the results returned by the query issued. The relevancy of the document is measured on a scale 1 to 5 where 1 means not relevant and 5 means highly relevant.2, 3 and 4 can be assigned as partial relevancy based on how much these retrieved

document are relevant to the query. Our experimental result on DCG@10 for 25 queries is shown in Figure 6.

Table 2. List of sample math queries

Query Id	Mathematical Notation/LaTeX Query
Q1	$(\neg q \vee \neg q \leftrightarrow (p \rightarrow \neg q)$
Q2	$(a-b)^2 = a^2 + b^2 - 2ab$
Q3	$(x \oplus y) = (x+7)(x+4) - 6k$
Q4	$\int \frac{\sin x}{x} dx$
Q5	$\neg(p \vee q) \vee (\neg p \wedge q) = \neg p$
Q6	$a \equiv b (mod\ n)$
Q7	$6.7^n - 2.3^n$
Q8	$((p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)))$
Q9	$(p-1)! \equiv -1 (mod\ p)$
Q10	$(a^2 + b^2 + c^2)^2 = 2(a^4 + b^4 + c^4)$
Q11	$(n-m) \mid (n^k - m^k)$
Q12	$(p \leftrightarrow q)$
Q13	$(p \vee q) \wedge (p \rightarrow r) \wedge (p \rightarrow r) \rightarrow r$
Q14	$(x+y) \times \frac{a}{b}$
Q15	$(x^2+1)^2$
Q16	$(z+y+x)^2$
Q17	$1 + tan^2 x$
Q18	$1 + x + x^2 + x^3 + \cdots$
Q19	$1 + x + x^2 + x^3 = y$
Q20	$1.1! + \cdots + n.n! = (n+1)!$
Q21	$10 sin^2(\theta) - 13 \sin(\theta) - 3 = 0$
Q22	$2 + 4 + \cdots + 2n = n(n+1)$
Q23	$3x^2 + 2x$
Q24	$7x + 5y = 3 (mod\ 4)$
Q25	$F_n^2 - (F_{n+1})(F_{n-1}) = (-1)^{n-1}$

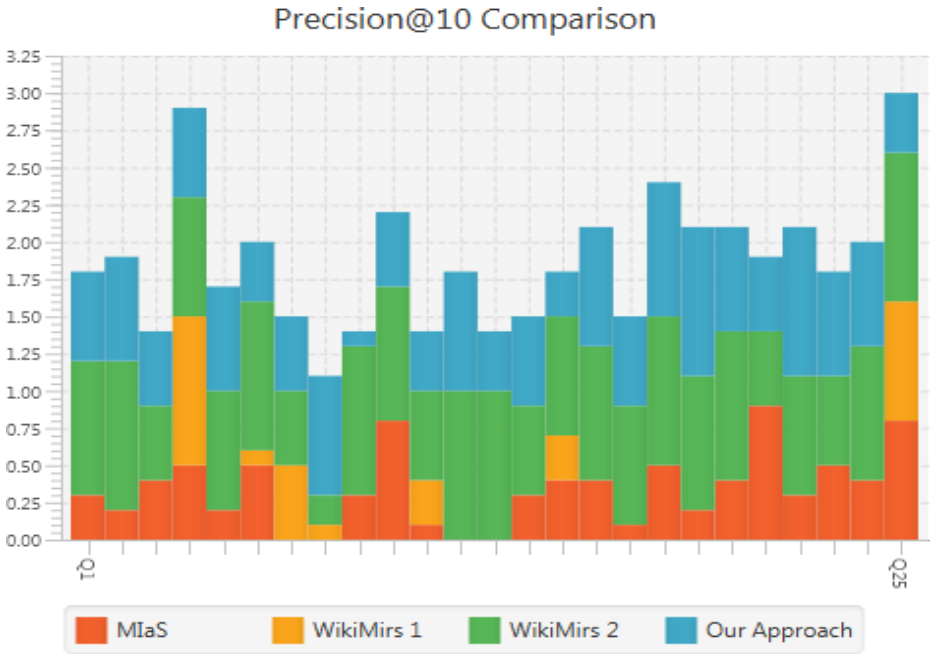

Figure 5. Precision@10 Comparision

It may be observed that our system i.e. SigMa performs better than MIaS and WikiMirs1 in terms of precision and it is comparable to WikiMirs2. As far as the usefulness of results is concerned, we observe that the SigMa yeilds much better DCG than MIaS and WikiMirs1 but WikiMirs2 achieves better

DCG than us. This may be due the fact that the improved version of WikiMirs i.e. WikiMirs2 incorporates an additional context index which improves upon the ranking of the results. Currently we are in process of indexing Mathematical Retrieval Collection 6. It contains more than 324,000 XHTML documents and having a size of 48 GB approx. (uncompressed). We are also analyzing different similarity measures and weighting scheme for mathematical expressions. We also assert that although the precision level of our system was decent but false positives are also inevitable in the signature based hash scheme. We are also examining other data structures like tries, directed acyclic graph, bloom filters to address the issues of structure preservation, ordering, normalization and false positives/negatives.

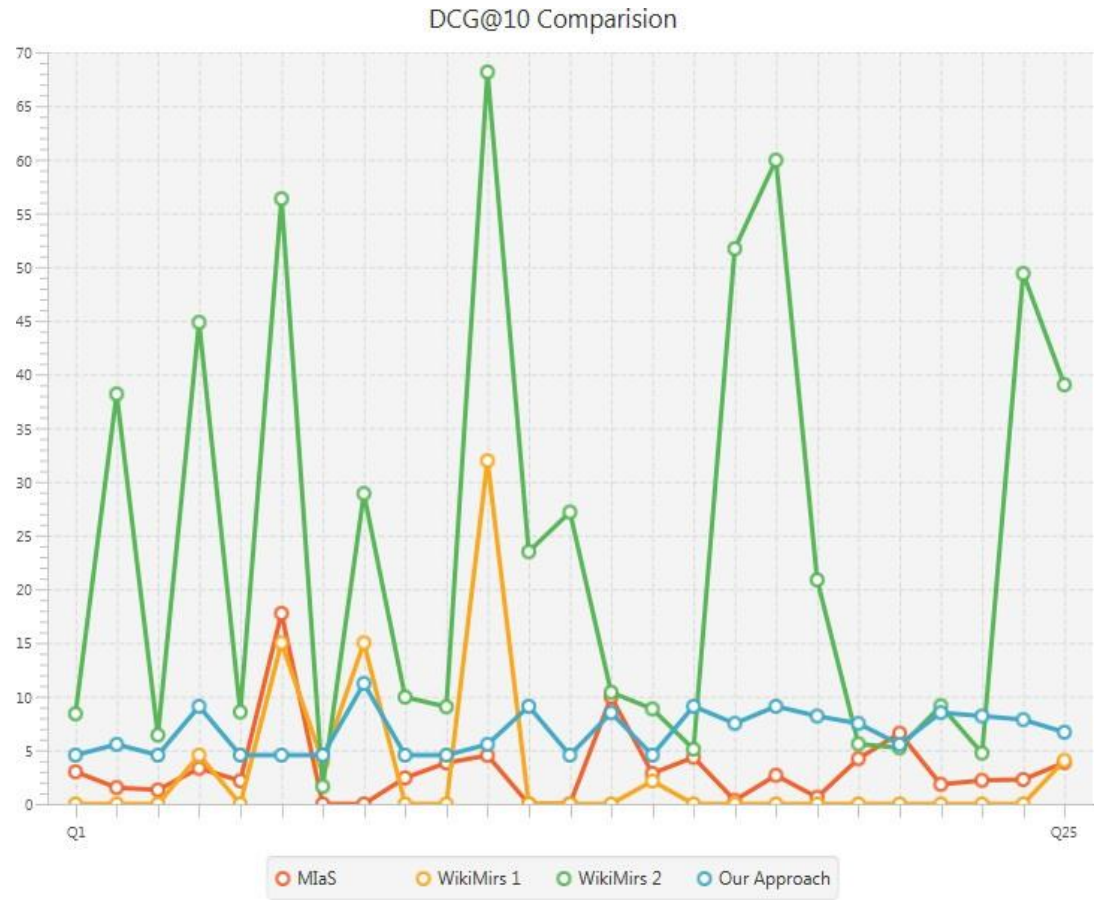

Figure 6. DCG@10 Comparision

4. CONCLUSION

In attempt of crafting a better retrieval model in the domain of MIR systems, we theorized that a signature based hashed indexing scheme would be better alternative instead of tree based or text based model. To reason with the theory we have constructed a mathematical search engine namely "SigMa" particularly for scientific documents with mathematical content.

At first mathematical information is extracted from the scientific documents and converted to structure encoded strings. These strings then are served as the input for the hash based indexing scheme, which aimed at converting these SES into a bit vector/signatures. A hash table of these signatures is created which enabled the online searching. Queries in the form of LaTeX strings are converted to P-MML on the fly and simultaneously bit vectors are generated. Finally these bitvectors are searched in the hash table of signatures and relevant results are retrieved if found a match. The system is compared with state-of-art MIR systems and we have observed that the preliminary results of this scheme are encouraging and competitive than other systems.

Although SigMa is aimed at faster retrieval and for this employs a hashing scheme based on document signatures. The limitation of this scheme is that false negative is inevitable. SigMa is also not void of false negatives. Similarity matching and weighting schemes have to dealt differently for

mathematical expression as it has to take into its consideration both the order as well as the equivalence of mathematical symbol notation. In future, other optimization techniques and weighting schemes can also be explored. Moreover for reducing false hits, Bloom Filter may be explored for its proven efficiency to eliminate false negatives. A better weighting scheme for the purpose of ranking and by exploring the semantics of mathematical expression along with meta data of the scientific documents could serve as a pointer to other research directions. Moreover, how to compute the similarity score according to the features of structures still remains an open problem, because the intent of different users of the MIR systems vary according to their context and precise needs.

REFERENCES

[1] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze. Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521865719.
[2] Richard Zanibbi, Dorothea Blostein. Recognition and Retrieval of Mathematical Expressions. In: *International Journal of Document Analysis and Recognition.* 2012; 15(4): 331–357. ISSN: 1433-2833.
[3] Ray R Larson, Chloe Reynolds, Fredric C Gey. The Abject Failure of Keyword IR for Mathematics Search: Berkeley at NTCIR-10 Math. In: *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies*, NTCIR-10, National Center of Sciences, Tokyo, Japan, 2013: 18-21.
[4] Amarnath Pathak, Partha Pakray, Sandip Sarkar, Dipankar Das, Alexander F Gelbukh. MathIRs: Retrieval System for Scientific Documents. In: *Computación y Sistemas*. 2017: 21(2).
[5] NTCIR-12 MathIR. http://ntcir-math.nii.ac.jp/introduction/. Accessed June 1, 2018.
[6] David A Grossman, Ophir Frieder. *Information Retrieval: Algorithms and Heuristics*. Norwell, MA, USA: Kluwer Academic Publishers, 1998. ISBN: 0792382714.
[7] Ricardo A Baeza Yates, Berthier Ribeiro Neto. *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN: 020139829X.
[8] Hazem; Salama Amany H Mamoon Mamoon, M El Bakry. Visualization for Information Retrieval based on Fast Search Technology. In: *IAES Indonesian Section, Indonesian Journal of Electrical Engineering and Informatics (IJEEI.)* 2013; 1(1): 27-42. ISSN: 2089-3272.
[9] Joydip Datta. An MTech Seminar Report: Ranking in Information Retrieval. Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, 2013.
[10] Mathematical Markup Language (MathML). https://en.wikipedia.org/wiki/MathML. Last accessed on June 1, 2018.
[11] LaTeX–A document preparation system. https://www.latex-project.org. Last accessed on June 15, 2018.
[12] OpenMath Home. http://www.openmath.org/. Last accessed on June 1, 2018.
[13] Dominique Archambault, Victor Moço. Canonical MathML to Simplify Conversion of MathML to Braille Mathematical Notations. In: *Computers Helping People with Special Needs*. Ed. by Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, and Arthur I. Karshmer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 1191–1198. ISBN: 978-3-540-36021-6.
[14] Ferruccio Guidi, Claudio Sacerdoti Coen. A Survey on Retrieval of Mathematical Knowledge. In: *Mathematics in Computer Science*. 2016; 10(4): 409–427. ISSN: 1661-8289.
[15] Adeel M, Cheung HS, Khiyal SH. Math Go! Prototype of a Content Based Mathematical Formula Search Engine. *Journal of Theoretical and Applied Information Technology*. 2008; 10(4): 1002–1012.
[16] Nguyen TT, Chang K, Hui SC. A Math-aware Search Engine for Math Question Answering System. Proceedings of the 21st ACM *International Conference on Information and Knowledge Management*; ACM: New York, NY, USA, CIKM '12, 2012: 724–733.
[17] Miner R, Munavalli R. An Approach to Mathematical Search Through Query Formulation and Data Normalization. *Towards Mechanized Mathematical Assistants*. 2007: 342–355.
[18] Petr Sojka, Martin Líška. The Art of Mathematics Retrieval. In: *Proceedings of the 11th ACM Symposium on Document Engineering*. DocEng'11. 2011: 57–60.
[19] Springer Innovations: LaTexSearch.com. https://www.springer.com/in/partners/society-zone-issues/springer-innovations--latexsearch-com/4516. Accessed June 1, 2018.
[20] Ricardo M Oliveira, Flavio B Gonzaga, Valmir C. Barbosa, and Geraldo Bonorino Xexéo. A distributed system for SearchOnMath based on the Microsoft BizSpark program. In: *Computing Research Repository* abs/1711.04189. 2017.
[21] Jozef Mišutka, Leo Galamboš. System Description: EgoMath2 as a Tool for Mathematical Searching on Wikipedia.Org. In: *Proceedings of the 18th Calculemus and 10th International Conference on Intelligent Computer Mathematics*. MKM'11. 2011: 307–309.
[22] Michael Kohlhase, Stefan Anca, Constantin Jucovschi, Alberto González Palomo, Sucan Ioan A. MathWebSearch 0.4, a semantic search engine for mathematics. In: *Manuscript at http://mathweb. org/projects/mws/pubs/mkm08. pdf (2008)*.
[23] Thomas Schellenberg, Bo Yuan, Richard Zanibbi. Layout-based substitution tree indexing and retrieval for mathematical expressions. In: *Proceedings of SPIE*. 2012; 8297.
[24] Petr Sojka, Michal Ruži˘cka, Vít Novotný. MIaS: Math-Aware Retrieval in Digital Mathematical Libraries".

eng. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Torino, Italy: Association for Computing Machinery, 2018. ISBN: 978-1-4503-6014-2.
[25] Xuan Hu, Liangcai Gao, Xiaoyan Lin, Zhi Tang, Xiaofan Lin, Josef B. Baker. WikiMirs: A Mathematical Information Retrieval System for Wikipedia. In: *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries.* JCDL '13. Indianapolis, Indiana, USA: ACM, 2013: 11–20.
[26] Yuehan Wang, Liangcai Gao, Simeng Wang, Zhi Tang, Xiaozhong Liu, Ke Yuan. WikiMirs 3.0: A Hybrid MIR System Based on the Context, Structure and Importance of Formulae in a Document. In: *Joint Conference on Digital Libraries*. ACM, 2015: 173–182.
[27] Benno Stein and Martin Potthast. Applying hash-based indexing in text-based information retrieval. In: *Proceedings of the 7th Dutch-Belgian Information Retrieval Workshop* (DIR 07). 2007: 29–35.
[28] W3C. MathML Fundamentals. https://www.w3.org/TR/WD-math/chapter2.html. Last accessed on June 15, 2018.
[29] P Pavan Kumar, Arun Agarwal, Chakravarthy Bhagvati. A string matching based algorithm for performance evaluation of mathematical expression recognition. In: *Sadhana.* 2014; 39(1): 63–79. ISSN: 0973-7677.
[30] Fuzzy string search. http://ntz- develop.blogspot.com/2011/03/fuzzy-string-search.html. Last accessed on June 1, 2018.
[31] Christos Faloutsos. Information Retrieval. In: ed. byWilliam B. Frakes and Ricardo Baeza-Yates. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992. Chap. Signature Files, pp. 44–65. ISBN: 0-13-463837-9.
[32] Samuel Huston. Indexing Proximity-based Dependencies for Information Retrieval. Doctoral Dissertations. University of Massachusetts-Amherst, 2014.
[33] LM Boitsov. Using Signature Hashing for Approximate String Matching. In: *Computational Mathematics and Modeling*. 2002; 13(3): 314–326. ISSN: 1573-837X.
[34] Sven Kosub. A note on the triangle inequality for the Jaccard distance. In: *CoRR* abs/1612.02696 (2016).
[35] J Naenudorn E, Wanapu S Niwattanakul S, Singthongchai. Using of Jaccard Coefficient for Keywords Similarity. In: *International Multi Conference of Engineers and Computer Scientists*. 2013; 1.
[36] *Enireddy Vamsidhar, B. Saichandana, J. Harikiran.* A Novel Approach for Feature Selection and Classifier Optimization Compressed Medical Retrieval Using Hybrid Cuckoo Search. In: *IAES Indonesian Section, Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*. 2018; 6(4): 410-417.
[37] Discounted cumulative gain. https://en.wikipedia.org/wiki/Discounted_cumulative_gain. Accessed July 1, 2018.