Recommending Answers to Math Questions Based on KL-Divergence and Approximate XML Tree Matching

Siqi Gao and Yiu-Kai Ng
Computer Science Department, Brigham Young University
Provo, Utah, USA

ABSTRACT

Math is the science and study of quality, structure, space, and change. It seeks out patterns, formulates new conjectures, and establishes the truth by rigorous deduction from appropriately chosen axioms and definitions. The study of math makes a person better at solving problems. It gives someone skills that can use across other subjects and apply in different job roles. In the modern world, builders use math every day to do their work, since construction workers add, subtract, divide, multiply, and work with fractions. It is obvious that math is a major contributor to many areas of study. For this reason, math information retrieval (Math IR) deserves attention and recognition, since a reliable Math IR system helps users find relevant answers to math questions and benefits all math learners whenever they need help solve a math problem, regardless of the time and place. Moreover, Math IR systems enhance the learning experience of their users. In this paper, we present *MaRec*, a recommender system that retrieves and ranks math answers based on their textual content and embedded formulas in answering a math question. *MaRec* ranks a potential answer A given a math question Q by computing the (i) KL-divergence score on A and Q using their textual contents, and (ii) the subtree matching score of the math formulas in Q and A represented as XML trees. The design of *MaRec* is simple and easy to understand, since it solely relies on a probability model and an elegant tree-matching approach in ranking math answers. Conducted empirical studies show that *MaRec* significantly outperforms (i) three existing state-of-the-art MathIR systems based on an offline evaluation, and (ii) two top-of-the-line machine learning systems based on an online analysis.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

KL-divergence, content similarity, subtree matching, math questions and answers

ACM Reference Format:
Siqi Gao and Yiu-Kai Ng. 2023. Recommending Answers to Math Questions Based on KL-Divergence and Approximate XML Tree Matching. In *Annual Intl. ACM SIGIR Conf. on Research & Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP '23), Nov. 26–28, 2023, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3624918.3625337

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR-AP '23, November 26–28, 2023, Beijing, China
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0408-6/23/11…$15.00
https://doi.org/10.1145/3624918.3625337

1 INTRODUCTION

Math information retrieval (Math IR) is a relatively young field that applies information retrieval techniques to extract math answers that contain either formulaic expressions, their related contents, or both. Researchers first published works in this field in the early 21st century [29], and in the past decade, the number of scholarly articles with math content submitted to ArXiv has doubled [17]. With such rapid growth and demand to offer advanced technology in retrieving math information, it is essential to invent novel and innovative approaches for researchers and ordinary users to search for math content in the STEM area with works published in scientific community and non-technical documents.

One of the design goals of Math IR is to retrieve math answers based on their relevance to a given user question with math formulas and textual content. There are many design issues and challenges, however, that make previous IR techniques unsuitable for formulaic questions. Consider how a user might search for an article that explains how to compute the formula $\sum_i^n i^2 + 2\sqrt{i}$. Using existing IR models, such as TF-IDF [26], BM25 [35], or web search engines, the user would be presented with documents that contain the keywords "summation", i, and n, or the mathematical formula $\sum_i^n i^2 + 2\sqrt{i}$, but not necessarily a discussion on the nature of the formula or how to compute the equation. Existing IR systems in document processing, which analyze wordings in a math question with formulas alone, cannot retrieve relevant answers, since they are incapable of examining the content of math formulas.

There are also significant difficulties in ranking math answers based on math questions. Consider how a user may search for the solution to a problem using the *Rice's theorem*, which states that $L(M_1) = L(M_2) \Rightarrow <M_1> \in P$ iff $<M_2> \in P$. A web engine search may rank multiple documents that are unrelated to the specified formula straightly based on the math notations involved. With scientific documents including a combination of discussions on various notations and formulaic expressions, it is essential to rank documents with their textual content and math notations based on their degrees of similarity with respect to a particular question on the formula. Solving this problem will alleviate the difficulties of ASK [2] for Math IR, and allow researchers and ordinary users to find answers that are relevant to their needs.

To meet the user's information needs on answering math questions, we propose *MaRec*, a *M*ath answer Recommender system, for retrieving and ranking math answers based on their textual content and embedded formulas in answering the corresponding math question. *MaRec* relies on (i) KL-divergence (also called Kullback-Leibler divergence) [6] to compare content matches, and (ii) an XML multiple tree matching algorithm to compare the formulaic

matches. *MaRec* first computes the degree of similarity of the textual content of a question and its counterpart in a potential answer using KL-divergence. Hereafter, *MaRec* scores the math formula in the question and its correspondent one in the answer based on an XML representation of the formulas, which are converted into tree structures to be matched. These two different measures are combined to create a robust ranking value to establish the order of relevant answers to the math question to be recommended. Empirical study conducted to verify the performance of *MaRec* on recommending math answers has shown that it outperformed existing Math IR systems and the results are statistically significant.

2 RELATED WORK

Math IR, which is quite new, has gained attention in the last two decades and many researchers in Math IR have proposed different algorithms [14, 19, 31, 49] to extract math information. *MaRec* is a hybrid textual and formula analysis approach, which is significantly different from existing Math IR systems presented below.

Text-Based. The text-based approaches convert math formulas into text and then apply the traditional textual question-and-answer methods to answer math questions. Math Language Processing (MLP) [37] first extracts variable identifiers from math formulas and locates the definiens, which are phrases that define identifiers, from the text of math questions. Hereafter, MLP determines the matching answers to a math question by comparing the definiens of the identifiers in the math question and the potential answers. Another text-based Math IR system, the Math Indexer and Searcher (MIaS) system [32], transforms math formulas in Math Markup Language (MathML)[1] into math tokens (called M-terms), which are comma-separated bag of tokens in MathML. MIaS ranks the potential answers to a math question using the TF-IDF on M-terms extracted from the question and answer formulas.

Vector-Based. The vector-based Math IR systems convert math formulas into vectors. Two well-known vector-based Math IR systems are proposed in [11, 33]. Dadure et al. [11] and Pathak et al. [33] create a bit position information table for math entities[2] and then transform math formulas and questions into the corresponding binary vectors through the bit table. The main difference between these two methods is their ranking strategy. Pathak et al. rank the results by counting the number of matching set bits, whereas Dadure et al. order the results by computing the relevance score, which is calculated based on the number of matching bits minus the number of different bits.

Tree-Based. The tree-based approaches represent math formulas in tree structures. Existing tree structures used for capturing math formulas fall into one of the two different types: Symbol Layout Tree (SLT) [27, 45] and Operator Tree (OPT) [46]. Tangent-L [30] converts MathML into a SLT and applies $BM25^+$ to rank the potential answers, whereas Tangent-S [12] transforms MathML into an OPT and employs a linear combination of the structure similarity scores between the transformed questions and potential answers for ranking the potential answers.

Machine Learning-Based. Researchers have trained machine learning models to answer multiple-choice math questions.

[1]Math Markup Language (MathML) [10], an application of XML, describes math notations and captures their content using XML tags and elements.
[2]Math Entities are elements in MathML.

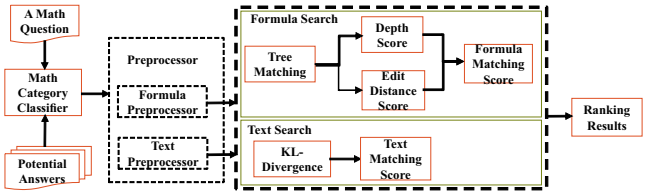


Figure 1: The overall process of *MaRec*

Sequence-to-Sequence Neural Network model [25] uses Long short-term memory (LSTM) to train a machine to understand the formula portion of mathematical questions and predict the correct answers from multiple choices. The mathematical problem solver proposed by Wang et al. [43] adapts the gated recurrent units (GRU) algorithm to convert math word questions into equations and predicts the correct answer from multiple choices. Other existing transformer models are trained by natural language [13, 23].

3 OUR MATH ANSWER RECOMMENDER

It is anticipated that a Math IR question Q contains textual information and a math formula, whereas an answer A to Q is a document that consists of the textual information (as an explanation) and a math equation, which serves as a potential answer to Q. Traditional Math IR systems [18, 39] often analyze only the textual portion of Q and A but exclude the formula portion of Q and A. We propose to retrieve and rank A to Q based on the (i) *degree of resemblance* of their math formulas, which is treated as a *tree matching* problem, and (ii) *textual similarity* of Q and A, which are treated as language models by using KL-divergence. Prior to extracting and ranking answers to a math question Q, we apply a classifier that determines the *subject area* to which Q belongs to speed up the process of searching for its answers. The overall process of our Math IR system, denoted *MaRec*, is shown in Figure 1.

3.1 A Math Question Classifier

We first train a classifier that categorizes a corpus of math questions and answers into their respective categories, such as algebra, geometry, number theory, probability, set theory, etc., which cover different subject areas in math. Hereafter, we match a question Q to a potential answer A based on the category to which Q and A belong. We have chosen the Multinomial Naïve Bayes (MNB) classifier [9], since it is simple and effective, to perform the classification by first converting user questions and answers into an event space.

3.1.1 Multinomial Event Space. An event outline is a set of possible events (or outcomes) from some process. A probability is assigned to each event in the event space, and the sum of the probabilities over all of the events in the event space must equal to one. To adapt and create this event space, we total all the word frequencies for each document, i.e., a user question or an answer in our case, in a category and create a vector where each dimension is the probability to find that word in the category.

$$P(c|d) = \frac{P(d|c)P(c)}{\sum_{c \in C} P(d|c)P(c)} = \frac{\prod_{i=1}^{n} P(w_i|c)P(c)}{\sum_{c \in C} \prod_{i=1}^{n} P(w_i|c)P(c)},$$

$$\text{where } P(c) = \frac{N_c}{N}, \text{ and } Class(d) = arg\ max_{c \in C} P(c|d) \quad (1)$$

where $P(c|d)$ is the probability of *document* d in class c, $P(d|c)$ is the probability that d is observed given c, $P(w_i|c)$ is the probability of the word w_i given c, $P(c)$ is the probability of observing c, C is the total number of classes, n is the total number of distinct words in d, N_c is the number of training documents in c, and N is the number of training documents.

3.1.2 The Multinomial Classifier. The MNB classifier is a linear classifier, suitable for classification with discrete features, such as word counts for text classification. The multinomial distribution requires integer feature counts as

$$P(d|c) = \prod_{w \in V} P(w|c)^{tf_{w,d}}, \text{ and } P(w|c) = \frac{tf_{w,c} + 1}{|c| + |V|} \quad (2)$$

where $tf_{w,d}$ is the frequency of occurrence of word w in document d, $tf_{w,c}$ is the frequency of occurrence of w in class c, $|c|$ is the number of words in the documents of c, and V is the number of distinct words in the corpus of documents.

3.2 KL-Divergence

KL-divergence, or Kullback-Leibler divergence, is a statistical distance used to measure the degree of difference between the *true* probability distribution P and probability distribution Q which is an approximation to P. A probability distribution is defined as a statistical function detailing the potential value a variable can have and the odds of each value occurring. KL-divergence is defined as

$$KL(P \,||\, Q) = \sum_{x} P(x) log_2 \frac{P(x)}{Q(x)} \quad (3)$$

One of the applications of KL-divergence is to use it as a measurement of the *textual similarity* between two language models P and Q, and in our *MaRec* system, they are the question language model and answer language model, respectively. The *question model* characterizes the usage of words in a math question to be processed by *MaRec*, whereas the *answer model* identifies the term occurrences in a potential answer to the question. An answer document can be viewed as a very small sample of text generated from the question model, which is a much larger sample of text. We apply KL-divergence in matching the textual portion of a potential answer document, denoted MA, with the textual portion of a math question, denoted MQ, by approximating how well the textual content of MA can be used for matching the content of MQ.

3.2.1 KL-divergence Based on Term Frequency. To calculate the KL-divergence score of MA and MQ simply based on the frequency of terms occurred in MA and MQ, respectively, denoted KLD_{Basis}, we first filter MQ and MA by removing stop words and then stem the remaining words. Given the non-stop, stemmed word list WQ of MQ and the non-stop, stemmed word list WA of MA, KL-divergence for the textual similarity of WQ and WA is computed as

$$KLD_{Basis}(WQ \,||\, WA) = \sum_{w \in WQ \cap WA} WQ(w) log_2 \frac{WQ(w)}{WA(w)}, \text{where}$$
$$WQ(w) = \frac{freq(w, WQ)}{|WQ|}, \text{ and } WA(w) = \frac{freq(w, WA)}{|WA|} \quad (4)$$

The KL-divergence score between a perfectly-matched WQ and WA, with the same matching frequency of each word in WQ and WA, yields the value 0. The text of WA that is closely approximated to the text of WQ yields a low KL-divergence value close to zero.

3.2.2 Enhanced Versions of Our Term-Frequent-Based KL-divergence. KLD_{Basis} can be further enhanced based on the TF-IDF [9] and Clarity approaches [3]. In KLD_{Basis}, the KL-divergence score of a potential answer A to a math question Q is simply established using the frequency of each word w in A co-occurred with the frequency of w in Q. TF-IDF offers an alternate method that evaluates a target word's weight by computing not only its frequency of occurrence in WQ (WA, respectively), but also the ratio of questions (answers, respectively) that include the word in WQ (WA, respectively). Besides using TF-IDF, we have also developed another alternative KL-divergence score, called $KLD_{Diversity}$, which evaluates words in math questions and answers by *topical reference* instead of just frequency and is derived from the *word clarity* approach [3].

KL-Divergence based on TF-IDF, denoted KLD_{TF-IDF}. TF-IDF evaluates the significance of a word or "term" based on the term's frequency (TF) and its inverse document frequency (IDF). In other words, TF measures the word's weight within a document, which is either a potential answer A or a question Q in our case, whereas IDF measures the word's weight relative to the entire dataset, which is the set of potential answers, As, to a math question. The following equation shows an alternative version of our KLD_{Basis}, with two components abstracted into parts for TF-IDF.

$$KL_{TF-IDF}(WQ||WA) = \sum_{w \in WQ \cap WA} WQ(w) \log \frac{WQ(w) \times TF\text{-}IDF_{WQ}}{WA(w) \times TF\text{-}IDF_{WA}},$$
$$\text{where } WQ(w) = \frac{freq(w,\ WQ)}{|WQ|},\ WA(w) = \frac{freq(w,\ WA)}{|WA|}$$
$$TF\text{-}IDF_{WQ}(w) = TF_{WQ} \times IDF_{WQ} = \frac{freq(w,\ WQ)}{max(freq(l, WQ) : l \in WQ)}$$
$$\times \log \frac{Number\_of\_Questions}{|\{WQ \in Q :\ w \in WQ\}|}, \text{and}$$
$$TF\text{-}IDF_{WA}(w) = TF_{WA} \times IDF_{WA} = \frac{freq(w,\ WA)}{max(freq(l,\ WA) :\ l \in WA)}$$
$$\times \log \frac{Number\_of\_Potential\_Answers}{|\{WA \in As :\ w \in WA\}|} \quad (5)$$

KL-Divergence Based on Word Diversity. Words in a question can be classified into different topics based on the degree of diversity of the words[3]. A word in a question Q with a high degree of diversity suggests that the word plays a significant role in representing different topics covered in Q. Based on this observation, we treat a potential answer with words of *high degree of diversity* that cover various topics in Q to be highly relevant to the semantics of Q. *MaRec* integrates the word diversity measure into KLD_{Basis} to further enhance the degree of accuracy in computing the textual similarity between Q and its potential answers. To determine the diversity of each word in Q, *MaRec* trained a Latent Dirichlet Allocation model (LDA) [4] to identify the *topics* of Q.

LDA, which is a topic generative probabilistic model for a corpus and a powerful unsupervised learning algorithm, classifies documents in the corpus to different latent topics based on the distribution of words in the documents. Given a set of documents S, LDA

[3]Word diversity refers to the logical/meaningful connection among various words.

generates a set of latent topics, and each topic contains a list of non-stop, stemmed words that are rated according to their probability of co-occurrence in S. *MaRec* employs LDA to classify words in a math question Q to different topics based on the number of words, i.e., WQ, in the textual portion of Q and the number of predefined topics. The larger the size of WQ is, the larger the number of topics should be generated to balance the number of words in each topic.

Given a math question Q, after the training process using LDA on Q is completed, LDA generates a number of topics, each of which is a ranked list of words, for Q. Words in each topic are ranked according to their relative probability values of the topic such that words in the topic with higher probability values are ranked higher. A word can appear in multiple topics, and the higher the number of topics it appears in, the higher the degree of *diversity* the word is. The diversity of a word is defined in Equation 6.

$$DT(w) = \frac{|t \in topics : w \in t|}{num\_topics}, \text{ where } num\_topics = (\lambda |WQ|) \quad (6)$$

After comparing the results using different λ values, λ in Equation 6 is set to be $\frac{1}{15}$ which is the most ideal value to determine the number of topics based on the experimental results (see Section 4.3 for the empirical study on choosing λ). Since lowly-ranked words in a topic have low degree of coherence to the topic, we only consider the top-k ranked words, denoted num_k, of each topic for textual content similarity measure in $KLD_{diversity}$.

$$num\_k = a + Round(\frac{num\_topic}{b}) \quad (7)$$

After considering different potential values of a and b in Equation 7, the ideal a and b are set to be 2 based on an empirical study. (See Section 4.3 for the empirical study on choosing the values of "a" and "b".) *MaRec* that combines the diversity and the KL-divergence measure to enhance the basic KL-divergence is given below.

$$KL_{Diversity}(WQ||WA) = \sum_{w \in com\_keywords} WQ(w) \log \frac{WQ(w)}{WA(w) \times DT(w)},$$

$$\text{where } com\_keywords = WQ \cap WA \cap \bigcup_{i=1}^{num\_topics} topic_i[: num\_k],$$

$$WQ(w) = \frac{freq(w, WQ)}{|WQ|}, \text{ and } WA(w) = \frac{freq(w, WA)}{|WA|} \quad (8)$$

where w are the common words among WQ, WA, and the combined list of top-num_k words in each topic, denoted $com\_keywords$. KL-divergence based on *Diversity* between a perfectly-matched WQ and WA yields the value 0.

3.3 The Approximate tree Matching Approach

One of the most important functions of a Math IR system is matching the math formula in a user question to the one in its potential answers [24, 29, 41, 49, 50]. To determine whether two math formulas are the same or similar, we develop $TreeMatch$, an unordered approximate tree matching algorithm.

In $TreeMatch$, we represent math formulas by using $MathML$, a markup language aims to facilitate the (re)use of mathematical and scientific content on the Web. Since $MathML$ is an XML structured language, the relationship of mathematical notations in a math formula is captured in a hierarchical manner. Based on the nested hierarchy structure of a math formula as captured in its corresponding MathML file, we convert the nested structure into a tree. We apply $TreeMatch$ to determine the similarity between any two math formulas represented by their respective MathML structure, which is our math formulas matching strategy.

There are four different types of tree pattern matching: *identical matching*, *sub-tree matching*, *partial matching*, and *empty matching*. Two *identically-matched* trees come with the same structures and node labels, whereas a *sub-tree* match between two trees indicates that the two trees impose a containment relationship, i.e., one tree is a subtree of the other tree, but not vice versa. A *partial match* occurs when two trees have overlapping branches, but do not completely overlap, whereas *empty matching* of two trees implies that there is no branch in common between the two trees. Moreover, there are two types of tree matching algorithms: *ordered* and *unordered*. Ordered tree matching requires that the sibling nodes (at the same hierarchical level under the same parent node) of two trees are in the same order, whereas unordered tree matching does not impose this constraint. In the representation of math formulas, if two subtrees have the same hierarchical order and their corresponding nodes are different only in their identifiers (i.e., variables) or numerical values, we treat the two trees as identical. $TreeMatch$ adapts the unordered tree matching approach which can handle all four different tree pattern matches and ignore the differences in mathematical identifiers (numerical values, respectively).

To accomplish the task of matching math formulas in MathML, we first transform the XML representation of each math formula into a tree pattern[4] and apply $TreeMatch$ to match the two tree structures. Given the *query* (also called *target*) tree QT specified in a *math question* Q and a potential *answer* (also called *candidate*) tree AT in a document D that are converted from the math formula given in Q and D, respectively, $TreeMatch$ compares the degree of similarity between the two trees based on two different scores.

3.3.1 The Depth Score of Tree Matching. $TreeMatch$ computes the *Depth_Score* of QT with respect to AT based on the depth of each branch in QT that matches a branch in AT. Assume that QB is the list of branches of QT and AB is the list of branches of AT. For each branch QB_i ($1 < i \leq |QB|$) in QB, $TreeMatch$ determines the length of the longest common sub-sequence d_i of QB_i among each branch AB_j ($1 < j \leq |AB|$) in AB, and the *depth_score* of the branch QB_i is computed as $d_i/|QB_i|$. The overall *Depth_Score* of QT with respect to AT, which is in the range between 0 and 1, is defined as

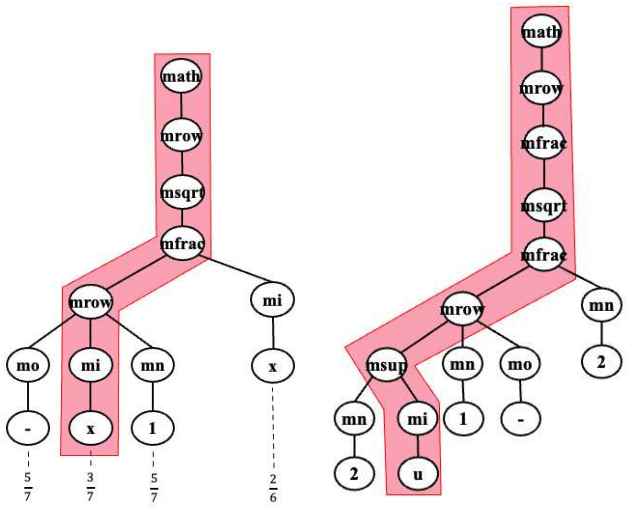
$$Depth\_Score(QT, AT) = \frac{\sum_{i=1}^{|QB|} Depth\_score(QB_i, AT)}{|QB|} \quad (9)$$

Example 1. Given the following equations:

$$\int_{-1}^{1} \frac{1}{x} \sqrt{\frac{1-x}{x}} \ln\left(\frac{2x^2+2x+1}{2x^2-2x+1}\right) dx \quad (10)$$

$$\int_{1}^{\infty} \frac{\ln\left(\frac{u^2+2u+2}{u^2-2u+2}\right)}{\sqrt{\frac{u^2-1}{2}}} du \quad (11)$$

[4]We use the ElementTree XML API [16] to parse a MathML file into its tree structure.



(a) A portion, $\frac{1-x}{x}$, of the question formula tree of Equation 10 **(b) A portion, $\frac{u^2-1}{2}$, of the answer formula tree of Equation 11**

Figure 2: The tree depth measure between two subtrees

Let Equation 10 (11, respectively) be the formula in a user question Q (potential answer A, respectively). A subtree of the query tree QT of Equation 10 is shown in Figure 2a, whereas a subtree of the potential answer tree AT of Equation 11 is depicted in Figure 2b. The depth score for each branch between the two subtrees is shown at the bottom of each branch in Figure 2a. Consider the highlighted branches in Figure 2. The longest common subsequence of the two branches is ["msqrt","mfrac", "mrow"] and its depth score is 3/7. The *Depth_Score* between the subtrees of QT and AT rooted at "msqrt" is $(5/7 + 3/7 + 5/7 + 2/6)/4 = 0.55$. □

3.3.2 The Edit-Distance Score of Tree Matching. *TreeMatch* determines the number of nodes that need to be adjusted to yield a matching between two trees by using our unordered tree edit distance algorithm. The edit distance between two trees is the *minimum* edit cost required to transform one tree into the other. There are three different edit distance operations: *insertion*, *deletion*, and *substitution*. Given a subtree A in QT and a subtree B in AT, the edit distance between A and B, denoted $d(A, B)$, is the minimum-weight series of edit operations that transforms B into A.

- Insertion. Let $B = uv$ and $A = uxv$, where u and v are sequences of nodes (including none) and x is a single node, then inserting x between u and v in B yields $uxv = A$.
- Deletion. Let $B = uxv$, and $A = uv$, then deleting x from B changes B to A.
- Substitution. Let $B = uxv$ and $A = uyv$, where $y \neq x$, then substituting x by y in B generates A.

To minimize the tree-matching cost, we apply the cost estimation algorithm in [47], which adapts the Levenshtein distance[5] that determines the edit distance of two trees. *TreeMatch* calculates the edit distance between AT and QT by setting the penalty to 1 for each insertion, deletion, or substitution operation. To keep *Edit_Distance_Score* in the same range as *Depth_Score*, *TreeMatch* normalizes the *Edit_Distance_Score* to be in the range of 0 and 1. *Edit_Distance_Score* is defined below, where $|QT|$ ($|AT|$, respectively) denotes the number of nodes in QT (AT, respectively).

$$Edit\_Distance\_Score = 1 - \frac{Edit\_Distance(QT, AT)}{|QT| + |AT|} \quad (12)$$

The two scores, i.e., *Depth_Score* and *Edit_Distance_Score*, measure the similarity of any two given trees. They are all within the range 0 and 1, and the *higher* the score, the more *similar* the *answer* tree is to the *query* tree. The final score of matching AT and QT is the average of the two scores of the two trees.

$$Matching\_Score = \frac{Depth\_Score + Edit\_Distance\_Score}{2} \quad (13)$$

The *Matching_Score* return 1 if QT and AT are matched identically. If the *Depth_Score* is 1, but the *Matching_Score* is less than 1, then QT and AT are sub-tree matching, but not identical. If the *Matching_Score* is 0, QT and AT are distinct trees, i.e., their matching is empty. If the matching is neither an identical match, sub-tree match, nor empty match, then it is a partial match.

3.4 Query-Answering

To determine the *ranking score* of a potential math answer A to a particular math query Q, we apply the *Borda Count* approach [8], which is a data fusion technique, to (i) the *matching score* (of the formulas) and (ii) one of the *KL-divergence variant scores*, i.e., the content similarity score, of A and Q, to compute the *degree of likelihood* of A being a potential answer to Q[6].

4 EXPERIMENTAL RESULTS

In this section, we (i) present the experimental results based on the empirical studies conducted on *MaRec*, (ii) evaluate the performance of the proposed recommender system on retrieving and ranking relevant answers to math questions, and (iii) compare its performance with other baseline Math IR Models[7].

To conduct our empirical studies on *MaRec*, we rely on existing datasets to provide the source data that include math questions and answers. After examining various datasets, we have chosen the Mathematics Stack Exchange[8] dataset that is robust and complete. Mathematics Stack Exchange is a math QA website for people studying math at any level to post math questions and offer answers to questions and allows professionals in related fields to communicate with one another. In addition, Stack Exchange users are allowed to rank different answers to a math question to indicate their relative degrees of relevance to the question. Stack Exchange users are also given the option to identify answers to a math question that are incorrect or inaccurate and alert other users the degree of accuracy of answers to a math question provided by others.

Besides picking the dataset, we have also chosen five baseline Math IR models and compared their performance with *MaRec*. These five baseline Math IR models cover a wide variety of Math

[5]The Levenshtein distance is an algorithm that measures the difference between two sequences.

[6]The source code of our tree matching and the implementation of the variants of our KL-divergence measures is available at https://drive.google.com/drive/folders/13gNTSsJwPsY2-URG4m0eD55HIuEcBp-M?usp=sharing.

[7]All the empirical study and experimental results are available at https://drive.google.com/drive/folders/1Xvy3U9jXTW_hyXxiCEJNfV4NPm-44Fgn? usp=sharing

[8]https://math.stackexchange.com/

IR methods, which are text-based, vector-based, tree-based, and machine learning-based, as discussed in Chapter 2. In comparing the performance of each of these baseline models with *MaRec*, we use the same Mathematics Stack Exchange (MSE) dataset that includes math questions and their corresponding answers. For the text-based, vector-based, and tree-based baseline models, we conducted an *offline* evaluation to compare their performance with *MaRec*. We treat answers extracted from MSE as ground-truth data and the performance evaluation of these models, including *MaRec*, are based on these ground-truth data. For the machine learning-based model, we performed an *online* evaluation instead, since machine learning Math IR models generate new answers on their own, whereas *MaRec* selects the answers from the MSE dataset, and it requires end users, who serve as appraisers, to evaluate the relevance of the answers extracted and ranked by the machine learning-based models and *MaRec*.

4.1 The Dataset

There are three well-known datasets used by existing Math IR systems, i.e., MSE, arXiv[9], and Wikipedia[10]. Using MSE data, ARQMath creates a dataset that includes (i) a formula file that contains all the formulas from 100 questions, and (ii) an HTML file that consists of all the questions in HTML format. NTCIR-12 task (the 3^{rd} Math Information Retrieval task at an international IR evaluation forum) also offers two datasets for its Math IR task: the arXiv dataset, which contains 105,120 scientific articles, and the English Wikipedia dataset, which consists of 319,689 documents. While the arXiv and Wikipedia datasets are math articles on different topics, MSE is an online math QA website that offers answers to questions with rankings. We decided to create a dataset from MSE, instead of using other existing Math IR datasets, since MSE provides questions and corresponding answers with rankings, instead of simply math articles. Moreover, answers extracted from MSE are shorter than their Wikipedia and arXiv counterparts, and thus are easier to process. Although ARQMath is also a MSE dataset, its size is relatively too small for our performance evaluation.

MSE database contains 1,547,227 questions, each of which is assigned three or more answers[11]. These data were extracted through the API provided by the MSE website as the dataset. All the data in the dataset are in JSON format. Each question in the dataset contains 50 fields. Since the majority of these fields are not useful for our empirical study, we only retained the question ID and the body of each question. Each answer in the dataset contains 34 fields, and we only kept the ranked score, the answer ID, the question ID the answer belongs to, and the body of each answer. In the body field of questions and answers, the textual content is represented in HTML format, while each mathematical formula is displayed in LaTeX format. To augment the processing of these data fields, we convert the textual portion of the data from HTML to plain text paragraphs and the formulas from LaTeX to MathML using existing converter tool[12]. A score of an answer is the ranking score provided by a MSE user to the answer. We use these scores as the ground truth values to evaluate the ranking strategy of *MaRec* and the baseline Math IR models.

The dataset used for *online* performance evaluation of *MaRec* was downloaded from Wolfram alpha[13], which is a QA website that offers answers to math questions to enhance users' math analytical skills. We randomly downloaded 540 questions and 7,634 different answers (see Section 4.4.2) that yield the 2^{nd} dataset for evaluation.

4.2 Performance Evaluation of Our Math Question Classification Approach

To evaluate the effectiveness of our classifier in *categorizing* math questions among different subject areas, we rely on the accuracy ratio, i.e., Accuracy = $\frac{Correctly\_classified\_instances}{Total\_number\_of\_instances}$, where $Total\_number\_of\_instances$ is the total number of questions to classify, which is 377,492 our case, extracted from the MSE dataset, which includes the subject area of each question, and $Correctly\_classified\_instances$ is the number of questions correctly assigned to their corresponding categories by the Naïve Bayes classifier.

We used the Naïve Bayes classifier (see details in Section 3.1.2) to determine the subject area *category* to which each question belongs. We tested the *accuracy* of the classifier by partitioning the training queries, training the classifier with 80% of them and testing it with the remaining 20%. The classifier determined the *correct* subject area category for 87% of the test questions. The accuracy of classification was slightly lower, in low-80 percentile, for the subject areas that are closely related, such as Geometry and Trigonometry, which is anticipated due to their common attributes. Overall, *MaRec* is still effective in retrieving relevant answers to Math questions posted by users for miscalculated categories due to the consideration of similar textual contents using KL-divergence.

4.3 An Offline Evaluation on *MaRec*

We conducted an offline performance evaluation of *MaRec* by comparing *MaRec* with three other existing Math IR systems that are well-established in the field in terms of their reputation in extracting relevant answers to math questions. We did not include ARQMath, since ARQMath is a math question answering evaluation task, not a Math IR system.

- **MIaS (Math Indexer and Searcher)** [38]. MIas, which is a math-aware, full-text based model, allows users to query math formulas and textual content in documents. The model first splits a document into textual and math portion and then indexes the text content in a conventional way. Hereafter, it detects partial matches of formulas by ordering the operands of the commutative operations, tokenizing the formula, and performing a variable and constant unification. MIaS assigns each indexed math expression a weight based on how far the actual formula is from the original representation, converts the XML nodes in each formula to a linear string form, and uses TF-IDF to calculate the matching score of an answer.
- **Tangent-CFT** [28]. Tangent-CFT is a vector-based Math IR model. It generates tuple sequence of a math formula with pairs of symbols and their relative positions and applies FastText,

[9] www.arxiv.org
[10] http://www.cs.rit.edu/~rlaz/NTCIR12_MathIR_WikiCorpus_ v2.1.0.tar.bz2
[11] The complete dataset is available at https://drive.google.com/drive/folders/1ZHhxaI8Jyw0TtHwabgsOmZyTmQ9Fwf9P?usp=sharing
[12] https://github.com/roniemartinez/latex2mathml
[13] https://www.wolframalpha.com/

which is derived from the word2vec model, to embed the formula. Hereafter, each tuple is assigned an n-dimensional vector and Tangent-CFT applies the cosine similarity measure to calculate the matching score of the formulas.

- **Tangent** [40]. Tangent is a tree-based Math IR model that converts a formula to a symbol layout tree. The model creates indices in a hash table that maps symbol pairs in formulas to a list of expressions containing them. Querying an index in a hash table requires mapping every matching expression into a list of symbol pairs it has in common with the question. For the ranking functions, Tangent applies F-Measure to determine the matching score of formulas.

4.3.1 Comparing the Performance of Math IR Systems Used for Retrieving Answers to Math Questions. Users of a ranking system tend to look at only the top few ranked results to find relevant suggestions [44]. Some search tasks have only the top-ranked suggestion, i.e., Precision at rank 1 (P@1), in mind, whereas others might consider the top-3 or top-5, i.e., Precision at rank 3 or rank 5 (P@3 or P@5), ranked suggestions. It turns out that the top-page 10 ranked results, Precision at Rank 10 (P@10), listed in a Google search has an average click-through rate of 24.7% (as of November 2022) and only 7.75% of users scroll past the first page of search engines, according to [1]. With these statistical data in mind, we have conducted an offline empirical study on the performance of *MaRec* using the three different variants of the KL-divergence measures, along with the three Math IR systems introduced in Section 4.3, based on $P@K$ ($K = 1, 3$, and 5). Besides $P@K$, we also conducted the performance measures using *Mean Reciprocal Rank (MRR)* and *normalized discounted cumulative gain* ($nDCG_K$). MRR measures the average of the reciprocal ranks of the first relevant document retrieved for a set of questions, whereas *nDCG* assesses how close the ranking result produced by a ranking system of the top-K values is to the best possible ranking performance. (K is also restricted to five for $nDCG_K$, same as the P@5 measure.) Figure 3 shows the experimental results based on the Mathematics StackExchange dataset achieved by *MaRec* using the variants of KL-divergence, in addition to MIas, Tangent-CFT, and Tangent. The overall $P@K$, MRR, and $nDCG_5$ values of $KLD_{Diversity}$ have indicated that it outperforms the remaining Math IR systems, including its variants, and the results are *statistically significant* based on the Wilcoxon Signed-Ranks Test ($p < 0.0001$) as shown in Tables 1 and 2.

4.3.2 Parameter Values Used in $KLD_{Diversity}$. As mentioned in Section 3.2.2, the λ value in Equation 6 and the "a" and "b" values in Equation 7 were determined experimentally. Table 3 shows the $P@K$ values for $KLD_{Diversity}$ using different parameter values of λ, a, and b. It reveals that when $\lambda = \frac{1}{15}$, its P@5 scores are the highest among other λ values, regardless what the values of "a" and "b" are. The same applies to "b" when it is set to be 2 and $\lambda = \frac{1}{15}$, regardless of the values of "a". To determine the ideal values of "a" is not as straightforward, since when "a" = 3, it yields the highest P@1 score; however, when "a" = 2, it yields the highest P@3 score. Since when $\lambda = \frac{1}{15}$, "a" = 2, and "b" = 2, the combination generates the highest P@3 and P@5 values and its P@1 value is only slightly smaller than the highest P@1 value, the combination is chosen as the default parameter values.

4.4 An Online Evaluation on *MaRec*

Besides conducting an offline evaluation, we also perform an online evaluation to verify the novelty of *MaRec* in recommending math answers as explained in Section 4. We first determine the ideal number of appraisers and test questions extracted from the Wolfram alpha dataset for the evaluation of *MaRec* so that they are *reliable* and *objective*.

4.4.1 The Number of Appraisers. In statistics, two types of errors, Types I and II, are defined [21]. Type I errors, also known as α errors, are the *mistakes* of *rejecting* a null hypothesis when it is true, whereas Type II errors, also known as β errors, are the *mistakes* of *accepting* a null hypothesis when it is false. We apply the formula in [21] shown below to determine the ideal number of appraisers, n, to evaluate the performance of *MaRec* online.

$$n = \frac{(Z_{\frac{\alpha}{2}} + Z_{\beta})^2 \times 2\sigma^2}{\triangle^2} + \frac{(Z_{\frac{\alpha}{2}})^2}{2} \tag{14}$$

where $\triangle$ is the *minimal expected difference* to compare answers extracted from the constructed dataset by our Math IR approach with manually-chosen answers, which is set to 1 in our study, since we expect our Math IR approach to extract relevant answers to math questions, if they exist, as good as the ones chosen manually; σ^2 is the *variance* of the extracted math answers and is set to be 2.15 in our study. It is computed by averaging the sum of the square difference between the mean and the actual number of useful math answers created for each one of the 100 test questions (downloaded from Stack Exchange chosen for verifying σ^2), which are computed on a *simple random sample* and do not change with a larger sample set of questions; α (β, respectively) denotes the probability of making a Type I (II, respectively) error, which is set to be 0.05 (0.20, respectively), and 1 - β determines the probability of a false null hypothesis that is correctly rejected, and Z is the value assigned to the standard *normal distribution* of extracted math answers. Based on the standard normal distribution, when $\alpha = 0.05$, $Z_{\frac{\alpha}{2}} = 1.96$, and when $\beta = 0.20$, $Z_{\beta} = 0.84$. When $\alpha = 0.05$ and $\beta = 0.20$, they imply that we have 95% *confidence* on the correctness of our analysis and that the probability of avoiding false negatives/positives) of our statistical study is 80%. According to [22], 0.05 is the commonly-used value for α, whereas 0.80 is a conventional value for 1 - β, and a test with $\beta = 0.20$ is considered to be statistically powerful. Based on these assigned values, the ideal number of appraisers is

$$n = \frac{(1.96 + 0.84)^2 \times 2 \times 2.15}{1^2} + \frac{1.96^2}{2} \cong 36 \tag{15}$$

The results collected from the 36 appraisers are expected to be comparable with the results that are obtained by the actual population [21], i.e., common math users.

4.4.2 The Number of Online Test Questions. To determine the ideal number of test questions to be included in the controlled experiments, we rely on two different variables: (i) the *average attention span* of an adult and (ii) the *average number of search queries* that a person often creates in one session when using a web search engine. As mentioned in [36], the average attention span of an adult is between 40 to 60 minutes. Furthermore, Jansen et al. [20], who have evaluated web users' behavior especially on (i) the amount of time web users spend on a web search engine, and (ii) the average

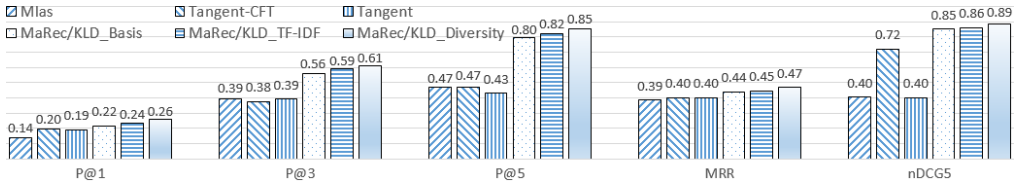


Figure 3: P@K, MRR, and $nDCG_5$ values achieved by *MaRec* with the variants of KL-divergence and the three baseline models

Table 1: Wilcoxon Signed-Ranks test among the baseline models and *MaRec* with the variants of KL-divergence using *P@K*

Models	Basis P@1	Basis P@3	Basis P@5	TF-IDF P@1	TF-IDF P@3	TF-IDF P@5	Diversity P@1	Diversity P@3	Diversity P@5
MIas	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
Tangent-CFT	0.0023	0.0001	0.0001	0.0009	0.0001	0.0001	0.0098	0.0001	0.0001
Tangent	0.1568	0.0001	0.0001	0.2656	0.0001	0.0001	0.0411	0.0001	0.0001

Table 2: Wilcoxon Signed-Ranks test for MRR and $nDCG_5$ achieved by the variants of KL-divergence and baseline models

Models	Basis MRR	Basis $nDCG_5$	TF-IDF MRR	TF-IDF $nDCG_5$	Diversity MRR	Diversity $nDCG_5$
MIas	0.0300	0.0001	0.0001	0.0001	0.0001	0.0001
Tangent-CFT	0.9480	0.0001	0.0980	0.0001	0.0001	0.0001
Tangent	0.3320	0.0001	0.3230	0.0001	0.0001	0.0001

Table 3: $P@k$ results for $MaRec/KLD_{Diversity}$ with different parameters for Equations 6 and 7

Values	$\lambda=\frac{1}{10}$ $a=4$ $b=2$	$\lambda=\frac{1}{10}$ $a=4$ $b=3$	$\lambda=\frac{1}{12}$ $a=2$ $b=2$	$\lambda=\frac{1}{15}$ $a=6$ $b=2$	$\lambda=\frac{1}{15}$ $a=4$ $b=2$	$\lambda=\frac{1}{15}$ $a=3$ $b=2$	$\lambda=\frac{1}{15}$ $a=2$ $b=2$	$\lambda=\frac{1}{15}$ $a=2$ $b=1$	$\lambda=\frac{1}{18}$ $a=2$ $b=2$
P@1	0.196	0.196	0.202	0.201	0.206	0.206	0.205	0.205	0.203
P@3	0.591	0.590	0.592	0.588	0.590	0.592	0.592	0.592	0.596
P@5	0.800	0.800	0.800	0.799	0.800	0.800	0.802	0.800	0.800

number of queries submitted by a user, estimate that the average number of queries created by each user in one session on a web search engine is about 5.1. Based on these studies, each appraiser was asked to evaluate our Math IR approach using *five* questions, since evaluating the Math IR system on the retrieved results, i.e., 5 answers, of each one of the five questions takes approximately 60 minutes, which falls into an adult time span. Moreover, each appraiser contributed three separated hours for the online evaluation. We randomly selected *540* (= 36 × 5 × 3) test questions from the Wolfram alpha dataset for the performance evaluation.

4.4.3 Online Performance Evaluation. We compared *MaRec* with two well-established machine learning-based Math IR systems for the online performance evaluation.

- **GPT-3** [5]. GPT-3 is a 3^{rd} generation generative pre-trained transformer, which is a Neural Network machine learning model that is trained using the Common Crawl dataset with nearly a trillion of words to produce any type of text using Internet data. It uses a large unsupervised corpus of tokens to train a language model. The novelty for this model is the few-shot learning for in-context learning. Unlike fine-turning, few-shot gives a few examples of the task besides the task description without gradient updates.
- **ProblemSolver** [25]. ProblemSolver consists of two different components: (i) a neural sequence-to-sequence translator that matches a question to its answer, and (ii) an application of an arithmetic tree approach to deal with Math SAT question answering. It applies a dual-pronged approach, building a Sequence-to-Sequence Neural Network pre-trained with augmented data that could answer all categories of questions and uses a system for tree matching. The model was trained using 600K questions.

4.4.4 Appraisers and Test Questions/Answers. To conduct the online performance evaluation of *MaRec* and compare its performance with GPT-3 and ProblemSolver, we recruited 36 students from our university to serve as appraisers of the study. These students are either undergraduate- or graduate-level students who are majoring in Mathematics, Computer Science, Electrical Engineering, Statistics, or Physics. Each one of these students was given fifteen different test questions and their corresponding answers to work with, which were extracted from Wolfram alpha dataset and processed by either *MaRec*, GPT-3, or ProblemSolver. The students are supposed to mark the answers to a question that are deemed relevant or correct, and rank the top-5 answers retrieved by *MaRec*, GPT-3, and ProblemSolver, respectively[14], and their rankings on the correctness of the answers serve as the *ground-truth* of the answers.

4.4.5 Comparing the Performance of MaRec and Other Math IR Systems. After the gold standard for each test case, i.e., question, provided by each one of the 36 appraisers were determined, we computed the *P*@1, *P*@5, and *MRR* values for the three Math IR systems, i.e., *MaRec*, GPT-3, and ProblemSolver, and the variants of different KL-Divergence approaches of *MaRec*, involved in our online empirical study. Figure 4 shows the performance metrics for the *P*@1 and *P*@5, in addition to *MRR*, which is the mean of

[14]Students were not aware of which answers to a particular question were retrieved by which one of the Math IR systems to avoid any bias in their relevance and rankings.

Table 4: Question processing time (in seconds) between *MaRec*, *MIaS*, *Tangent*, *Tangent-CFT*, *ProblemSolver*, and *GPT-3*

Models	MaRec KLD_{Basis}	MaRec KLD_{TF-IDF}	MaRec $KLD_{Diversity}$	MIaS	Tangent	Tangent-CFT	Problem Solver	GPT-3
Processing time	0.025s	0.025s	0.037s	0.829s	0.386s	4.000s	2.545s	15.000s

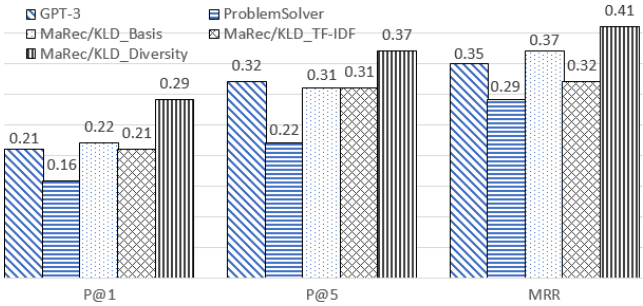


Figure 4: The P@1, P@5, and MRR values for the online evaluation of various Math IR systems and *MaRec*

the reciprocal ranks at which the *first* correct answer (among the top-5 ranked answers retrieved by a Math IR system) for each test question is made. The *P*@1, *P*@5, and MRR scores of *MaRec* based on $KLD_{Diversity}$ are higher than the corresponding ones of GPT-3, ProblemSolver, and its variants, and the results are statistically significant based on the Wilcoxon Signed-Rank test ($p < 0.03$). We did not show the $nDCG_5$ scores, since they are very similar to *P*@5.

Based on the offline and online performance evaluations on *MaRec*, it clearly shows that $KLD_{Diversity}$ outperforms its variants and its question processing time is compatible with its variants (as verified in Section 4.6). Thus, it is the core contributor and legitimate choice of *MaRec* for determining textual content similarity.

4.5 An Ablation Test on *MaRec*

To verify the effectiveness of *MaRec* in combining the tree matching and $KLD_{Diversity}$ approaches for retrieving relevant Math answers to user queries, we conducted another empirical study based on individual components of *MaRec*. Figure 5 demonstrates that when only some components of *MaRec* are applied, its performance drops compared with the results depicted in Figure 3. It further confirms the necessity of applying both tree matching and $KLD_{diversity}$ in *MaRec* for retrieving relevant Math information.

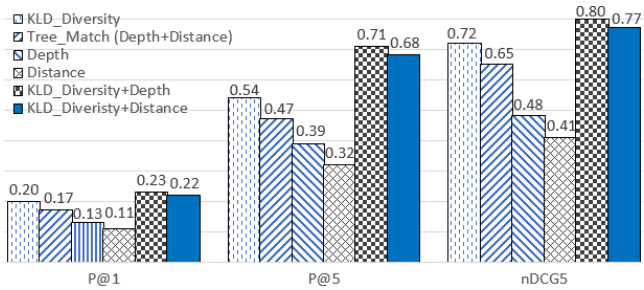


Figure 5: P@1, P@5, and nDCG values for the ablation test

4.6 Question Processing Time

One of the design goals of *MaRec* is to process user questions with processing time compatible with existing web search engines. With that in mind, we have conducted a performance evaluation to compute the average question processing time of *MaRec* and other baseline models for retrieving and ranking the answers to each math question in the MSE dataset *D*. The question processing time is evaluated using *D* and on the same laptop PC. The PC is a Macbook Pro with M1 chip, 8-core CPU with 4 performance cores and 4 efficiency cores, 8-core GPU, and 16-core Neural Engine. As shown in Table 4, all the variants of *MaRec* retrieve and rank all the answers to a math question instantly and outperform other baseline models in term of the question processing time. *Tangent-CFT* is slower than other compared models (except GPT-3), since it employs three ranking models: the SLT, OPT, and SLT-Type encoding models, which are combined to generate the final ranking result for a math question. Moreover, GPT-3 is the slowest, since it takes longer time to produce a new answer to a question, while other models retrieve existing answers.

5 CONCLUSION AND FUTURE WORK

Mathematics is widely used in daily life. There has been a sustained level of task, such as mortgage lending, budget management, stock trading, and playing music, that involves using math. Majority of the careers require some basic knowledge of math, and more in-depth knowledge of math is expected for students and workers in the STEM areas. People who are proficient in math have more opportunities and better chance for career advancement than others who are less proficiency in math [15]. Unfortunately, according to the Public School View site [42], in the year of 2023 the national math proficiency average in USA is 38%. Moreover, only 60% of 12^{th} grade students scored at or above the proficient level on the NAEP[15] Math assessment [7]. It is important to develop math information retrieval (IR) systems that lighten the burden on the users to search for desired math information with the design goal of enhancing their math proficiency skills. In this paper, we propose *MaRec*, a math IR system that supports users in searching for answers to math problems. We apply subtree matching to the formulaic portion of math questions and answers to detect similar or identical formulas. We also design variants of KL-divergence measures to best match between the semantic contents of math questions and answers. The combination of these two approaches is elegant. Furthermore, *MaRec*, which is fast and easy to implement, outperforms existing Math IR systems in terms of efficiency and effectiveness in retrieving and ranking relevant answers to math questions. Its design is a contribution to the Math IR community.

MaRec is designed for matching formulas in MathML that are in textual format. Math questions and answers, however, may be in graphical format. There are a few machine learning algorithms for recognizing image Math formulas [34, 48]; however, none of them has considered using formula recognition to process math questions. For future work, we would like to develop image formula recognition algorithms so that the further enhanced *MaRec* can handle Math questions and answers that contain images.

[15]NAEP, National Assessment of Educational Progress (The Nation's Report Card)

REFERENCES

[1] P. Ahern. 2023. 27 Mind-Bottling SEO Stats for 2023 (+ Beyond). https://intergrowth.co/seo-stats/. Intergrowth.

[2] N. Belkin, R. Oddy, and H. Brooks. 1982. ASK for Information Retrieval: Part I. Background and Theory. *Journal of Documentation* (1982).

[3] S. Bhatia, D. Majumdar, and P. Mitra. 2011. Query Suggestions in the Absence of Query Logs. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 795–804.

[4] David Blei, Andrew Ng, and Michael Jordan. 2001. Latent dirichlet allocation. *Advances in neural information processing systems* 14 (2001).

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and A. Askell. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[6] Y. Bu, S. Zou, Y. Liang, and V. Veeravalli. 2018. Estimation of KL Divergence: Optimal Minimax Rate. *IEEE Transactions on Information Theory* 64, 4 (2018), 2648–2674.

[7] The Nation's REport Card. 2019. National Achievement-Level Results.

[8] G. Cormack, C. Clarke, and S. Buettcher. 2009. Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 758–759.

[9] W. Croft, D. Metzler, and T. Strohman. 2010. *Search Engines: Information Retrieval in Practice*. Addison Wesley.

[10] D. Carlisle and P. Ion and R. Miner. 2021. Mathematical Markup Language (MathML), Version 3.0, 2^{nd} Edition. W3C. https://www.w3.org/ TR/2014/REC-MathML3-20140410/.

[11] P. Dadure, P. Pakray, and S. Bandyopadhyay. 2022. Embedding and Generalization of Formula with Context in the Retrieval of Mathematical Information. *King Saud University-Computer and Information Sciences* 34, 9 (2022), 6624–6634.

[12] K. Davila and R. Zanibbi. 2017. Layout and Semantics: Combining Representations for Mathematical Formula Search. In *Proceedings of the* 40^{th} *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1165–1168.

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] S. Dominich. 2001. *Mathematical Foundations of Information Retrieval*. Vol. 12. Springer Science & Business Media.

[15] R. Fatima. 2012. Role of Mathematics in the Development of Society. *National Meet on Celebration of National Year of Mathematics. Organized by NCERT, New Delhi* 1 (2012), 12.

[16] L. Fredrik. [n. d.]. xml.etree.ElementTree-The ElementTree XML API. https://github.com/python/cpython/tree/3.11/Lib/xml/etree/ElementTree.py.

[17] P. Ginsparg. 2021. Lessons from arXiv's 30 Years of Information Sharing. *Nature Reviews Physics* 3, 9 (2021), 602–603.

[18] P. Gupta and V. Gupta. 2012. A Survey of Text Question Answering Techniques. *International Journal of Computer Applications* 53, 4 (2012).

[19] X. Hu, L. Gao, X. Lin, Z. Tang, X. Lin, and J. Baker. 2013. Wikimirs: A Mathematical Information Retrieval System for Wikipedia. In *Proceedings of the* 13^{th} *ACM/IEEE-CS joint conference on Digital Libraries*. 11–20.

[20] B. Jansen, A. Spink, and T. Saracevic. 2000. Real Life, Real Users, and Real Needs: a Study and Analysis of User Queries on the Web. *IPM* 36, 2 (2000), 207–227.

[21] B. Jones and M. Kenward. 2003. *Design and Analysis of Cross-Over Trials, 2nd Ed*. Chapman and Hall.

[22] L. Kazmier. 2003. *Schaum's Outline of Business Statistics*. McGraw-Hill.

[23] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. 2019. Bart: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[24] M. Líška, P. Sojka, and M. Ružička. 2015. Combining Text and Formula Queries in Math Information Retrieval: Evaluation of Query Results Merging Strategies. In *Proceedings of NWSearch*. 7–9.

[25] X. Luo, A. Baranova, and J. Biegert. 2019. Problemsolver at Semeval-2019 Task 10: Sequence-to-Sequence Learning and Expression Trees. In *Proceedings of the* 13^{th} *International Workshop on Semantic Evaluation*. 1292–1296.

[26] C. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press Cambridge.

[27] B. Mansouri, V. Novotnỳ, A. Agarwal, D. Oard, and R. Zanibbi. 2022. Third CLEF Lab on Answer Retrieval for Questions on Math (Working Notes Version). *Proceedings of the CLEF 2022 (CEUR Working Notes)* (2022).

[28] B. Mansouri, S. Rohatgi, D. Oard, J. Wu, C. Giles, and R. Zanibbi. 2019. Tangent-CFT: An Embedding Model for Mathematical Formulas. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 11–18.

[29] B. Miller and A. Youssef. 2003. Technical Aspects of the Digital Library of Mathematical Functions. *Annals of Math. & AI* 38, 1 (2003), 121–136.

[30] Y. Ng, D. Fraser, B. Kassaie, and F. Tompa. 2021. Dowsing for Math Answers. In *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer, 201–212.

[31] T. Nguyen, K. Chang, and S. Hui. 2012. A Math-Aware Search Engine for Math Question Answering System. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. 724–733.

[32] V. Novotnỳ, P. Sojka, M. Stefánik, and D. Lupták. 2020. Three is Better than One: Ensembling Math Information Retrieval Systems. In *CLEF (Working Notes)*.

[33] A. Pathak, P. Pakray, and A. Gelbukh. 2018. A Formula Embedding Approach to Math Information Retrieval. *Computación y Sistemas* 22, 3 (2018), 819–833.

[34] S. Peng, K. Yuan, L. Gao, and Z. Tang. 2021. Mathbert: A Pre-Trained Model for Mathematical Formula Understanding. *arXiv preprint arXiv:2105.00377* (2021).

[35] S. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in IR* 3, 4 (2009), 333–389.

[36] L. Rozakis. 2002. *Test Taking Strategies and Study Skills for the Utterly Confused*. McGraw Hill.

[37] M. Schubotz, A. Grigorev, M. Leich, H. Cohl, N. Meuschke, B. Gipp, A. Youssef, and V.Markl. 2016. Semantification of Identifiers in Mathematics for Better Math Information Retrieval. In *Proceedings of the* 39^{th} *International ACM SIGIR conference on Research and Development in Information Retrieval*. 135–144.

[38] P. Sojka and M. Líška. 2011. The Art of Mathematics Retrieval. In *Proceedings of the 11th ACM Symposium on Document Engineering*. 57–60.

[39] R. Srihari and W. Li. 2000. A Question Answering System Supported by Information Extraction. In *Sixth Applied Natural Language Processing Conference*. 166–172.

[40] D. Stalnaker. 2013. *Math Expression Retrieval Using Symbol Pairs in Layout Trees*. Master's thesis. Rochester Institute of Technology.

[41] Y. Stathopoulos and S. Teufel. 2016. Mathematical Information Retrieval Based on Type Embeddings and Query Expansion. In *Proceedings of COLING*. 2344–2355.

[42] Public School View. 2023. Average Public School Math Proficiency. https://publicschoolreview.com/average-math-proficiency-stats/national-data.

[43] Y. Wang, X. Liu, and S. Shi. 2017. Deep Neural Solver for Math Word Problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 845–854.

[44] WebFX. 2022. 95 SEO Statistics from This Year That'll Transform Your Strategy. https://www.webfx.com/seo/statistics/.

[45] R. Zanibbi and D. Blostein. 2012. Recognition and Retrieval of Mathematical Expressions. *Document Analysis and Recognition (IJDAR)* 15, 4 (2012), 331–357.

[46] R. Zanibbi and D. Blostein. 2012. Recognition and Retrieval of Mathematical Expressions. *Document Analysis and Recognition (IJDAR)* 15, 4 (2012), 331–357.

[47] K. Zhang. 1996. A Constrained Edit Distance between Unordered Labeled Trees. *Algorithmica* 15, 3 (1996), 205–222.

[48] Z. Zhang, T. Wang, X. Song, and Y. Wang. 2022. The Design and Implementation of the Natural Handwriting Mathematical Formula Recognition System. In *Proceedings of the 6th International Conference on Advances in Image Processing*. 114–121.

[49] J. Zhao, M. Kan, and Y. Theng. 2008. Math Information Retrieval: User Requirements and Prototype Implementation. In *Proceedings of the* 8^{th} *ACM/IEEE-CS joint conference on Digital Libraries*. 187–196.

[50] W. Zhong, J. Yang, and J. Lin. 2022. Evaluating Token-Level and Passage-Level Dense Retrieval Models for Math Information Retrieval. *arXiv preprint arXiv:2203.11163* (2022).