Dowsing for Answers to Math Questions: Ongoing Viability of Traditional MathIR

Yin Ki Ng[1], Dallas J. Fraser[2], Besat Kassaie[1] and Frank Wm. Tompa[1]

[1]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, N2L 3G1
[2]Knowledgehook Inc, 151 Charles St W, Kitchener, ON, Canada, N2G 1H6

Abstract

We present our application of the math-aware search engine Tangent-L to the 2021 ARQMath Lab. This is a continuation of our MathDowsers submissions to last year's Lab, where we produced the best Task 1 participant run. Since then, we have improved the search engine's formula retrieval power by considering additional math features in the ranking function. This year, we also explore two approaches to incorporate proximity in evaluating the suitability of a document to be considered a match to a query.

For the 2021 ARQMath Lab, our primary run in Task 1 produces an nDCG′ value of 0.434, which is nearly five points higher than that produced by the second-best participant run. An unsubmitted run, which corrects the setup of the primary run and preserves duplicate keyword terms during query term extraction, produces an even higher nDCG′ of 0.462. Meanwhile, our primary run in Task 2 produces an nDCG′ value of 0.552, which is the best automatic run and is comparable to the best participant run, a manual run from the Approach0 team.

The success of our runs continue to demonstrate that a traditional math information retrieval system remains a viable option for Community Question Answering specialized in the mathematical domain and for in-context formula retrieval.

Keywords

Community Question Answering (CQA), Mathematical Information Retrieval (MathIR), Symbol Layout Tree (SLT), Mathematics Stack Exchange (MSE), ARQMath Lab, Tangent-L, formula matching, proximity

CEUR-WS.org/Vol-2936/paper-05.pdf

1. Introduction

The growing popularity of Community Question Answering (CQA) sites such as Math Stack Exchange[1] (MSE) and Math Overflow[2] demonstrates the need to find answers to mathematical questions, especially for questions posed in mathematical natural language. An effective question answering system capable of handling mathematical formulas and terminology would be of great interest to help serve this need.

The ARQMath Lab at CLEF 2021 [1], hereafter referenced as *ARQMath-2*, continues the previous year's Lab [2] (*ARQMath-1*) by sponsoring an evaluation exercise centering around

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania
kiking0501@gmail.com (Y. K. Ng); dallas.fraser.waterloo@gmail.com (D. J. Fraser); bkassie@uwaterloo.ca (B. Kassaie); fwtompa@uwaterloo.ca (F. Wm. Tompa)
https://www.linkedin.com/in/kiking0501/ (Y. K. Ng); https://uwaterloo.ca/scholar/bkassaie/home (B. Kassaie); http://www.cs.uwaterloo.ca/~fwtompa/ (F. Wm. Tompa)
0000-0002-1907-9535 (F. Wm. Tompa)
© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

[1]https://math.stackexchange.com
[2]https://mathoverflow.net

Figure 1: Researcher dowsing for answers to math queries.

a CQA Task with questions involving math data. The Labs use a collection of questions and answers from MSE between 2010 and 2018 consisting of approximately 1.1 million question-posts and 1.4 million answer-posts. In this Lab series, Task 1 is the CQA Task in which participants are asked to return potential answers to unseen mathematical questions among existing answer-posts in the collection. The closely related Task 2 considers formula retrieval in-context, in which formulas within questions serve as queries for matching relevant formulas from question-posts and answer-posts in the same collection.

In ARQMath-1, the Waterloo team of MathDowsers (Figure 1) participated in Task 1, and our best run achieved an nDCG' value of 0.345 [3], which outperformed other participating systems [4, 5, 6, 7]. Our approach was a three-stage Mathematics Information Retrieval (MathIR) system centered around the use of a math-aware search engine, Tangent-L [8]: first, topics of mathematical questions were automatically transformed into formal queries consisting of keywords and formulas; then the formal queries were executed against a corpus of MSE question-answer pairs by Tangent-L; finally, results were re-ranked based on a linear regression model trained on CQA metadata using mock relevance assessments. Submissions were made based on different configurations in each stage of the system, and the best run was produced without re-ranking, demonstrating success of a traditional math-aware query system in addressing a CQA task specialized in the mathematical domain.

For ARQMath-2, we participate again as the MathDowsers team for Task 1 and (for the first time) Task 2, with the goal to continue exploring the potential of a traditional math-aware query system in tackling both tasks. In particular, we are interested in further developing the formula matching capability of our core math-aware search engine Tangent-L, given that a satisfactory performance has been observed over formula-dependent questions in ARQMath-1 [9]. With the empowered Tangent-L, we then refine our system for Task 1 and develop two baseline approaches for Task 2.

Our refinement is successful, with our primary run for Task 1 continuing to be the best participant run with respect to the primary measure nDCG'[3]. Our primary run for Task 2 turns out to be the most effective automatic run, essentially indistinguishable from the best participant run, a manual run from the Approach0 team. In this paper, we present:

- an updated Tangent-L with several avenues that improve its formula matching capability,

[3]Normalized Discounted Cumulative Gain (nDCG) with unjudged documents removed

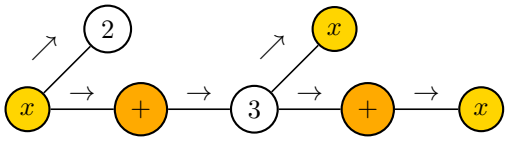

Figure 2: Symbol Layout Tree for $x^2 + 3^x + x$ with repetitions highlighted.

Table 1
Generated repetition tokens for the formula in Figure 2.

Token Type	*Tokens Generated*		*Remark*
Repeated symbols	$\{x, \rightarrow\rightarrow\nearrow\}$	$\{x, \rightarrow\rightarrow\rightarrow\rightarrow\}$	The first occurrence of x resides on the same path as each of the second and third occurrences.
	$\{+, \rightarrow\rightarrow\}$		Similarly for $+$.
	$\{x, \nearrow, \rightarrow\rightarrow\}$		The second and third occurrences of x lie on difference root-to-leaf paths and share a closest common ancestor.
Augmented locations	$\{x, \rightarrow\rightarrow\nearrow, \emptyset\}$	$\{x, \rightarrow\rightarrow\rightarrow\rightarrow, \emptyset\}$	Augmented with the path from the root to the first occurrence.
	$\{+, \rightarrow\rightarrow, \rightarrow\}$		Similarly for $+$.
	$\{x, \nearrow, \rightarrow\rightarrow, \rightarrow\rightarrow\}$		Augmented with the path from the root to the closest common ancestor.

- a refinement of our system for mathematical answer retrieval with respect to query conversion and searching with Tangent-L,
- two related approaches that are motivated by proximity,
- for in-context formula retrieval, two simple baselines based on our developed system,
- performance results for both Task 1 and Task 2 in ARQMath-2

2. Improving Formula Matching with Tangent-L

Tangent-L is the cornerstone of our system for the tasks. It is a traditional math-aware query system built on the popular Lucene text search platform [10]. During both index time and search time, it converts a formula into a bag of math tokens that each capture local characteristics of the *Symbol Layout Tree* (SLT) representation of a formula [11], so that mathematical documents can be matched against a query through text tokens and converted math tokens using a weighted $\mathrm{BM25}^+$ ranking [12].

The basic math tokens used by Tangent-L and the approach to weighting text against math tokens are described elsewhere [9]. In this section, we describe improvements tested in this year's Lab.

2.1. Repeated Symbols

Repetitions of symbols are commonplace in a formula; for instance, x repeats in the formula $x^2 + 3^x + x$, as does the operator $+$. Ideally, a search for either $y^x - x$ or $6x^3 - y + x$ could match that formula because of the pattern of repetitions for x, and a search for $2y^3 + y + 5$ could also match because of the repeated symbol $+$.

With this motivation, a new type of token—*repetition tokens*—is introduced into Tangent-L's formula representation to capture this characteristic. Repetition tokens are generated based on the relative positions of the repeated symbols in the formula's SLT representation. For every pair of repeated symbols:

1. if the pair of repeated symbols reside on the same root-to-leaf path of the SLT (that is, one is an ancestor of the other), then a repetition token {*symbol*, p} is generated, where p represents the path between the repeated symbols;
2. otherwise, a repetition token {*symbol*, p_1, p_2} is generated where p_1 and p_2 represent the paths from the closest common ancestor in the SLT to each repeated symbol.

If a symbol repeats k times where $k > 1$, $\binom{k}{2}$ repetition tokens are generated for that symbol following the above procedure. For each of these tokens, an additional "location" token is generated with the augmentation of the path traversing from the root to the closest common ancestor of the pair. As such, a total of $2 \cdot \binom{k}{2}$ repetition tokens are generated and indexed. Table 1 shows the repetition tokens that would be indexed for the formula $x^2 + 3^x + x$ in Figure 2.

2.2. Revised Ranking Formula

With the introduction of repetition tokens, Tangent-L now generates three token types: text tokens, regular math tokens, and repetition tokens from documents or queries containing mathematical expressions. During a search, Tangent-L applies $\mathrm{BM25}^+$ ranking to the query terms and the document terms, using custom weights for each class of token as described here.

Let q_t be the set of text tokens, q_m be the set of regular math tokens, and q_r be the set of repetition tokens generated for the query terms. Let d be a document represented by the set of all its indexed tokens. Then the revised ranking formula with the repetition tokens is:

$$\begin{aligned}&\mathrm{BM25}^+_{\mathrm{w}}(q_t \cup q_m \cup q_r, d) = \\ &\qquad \alpha \cdot \frac{\gamma \cdot \mathrm{BM25}^+(q_r, d) + (1-\gamma) \cdot \mathrm{BM25}^+(q_m, d)}{\max(\gamma, 1-\gamma)} + (1-\alpha) \cdot \mathrm{BM25}^+(q_t, d)\end{aligned} \tag{1}$$

where α and γ are parameters ranging from 0 to 1. The value of α balances the weight of math features against keyword features, while the value of γ balances the weight of repetitions within math formulas against other math features. Both parameters can be tuned based on the target dataset.

2.3. Formula Normalization

Mathematical expressions can be rewritten in numerous ways without altering their meaning. For example, $A + B$ matches $B + A$ semantically because of the commutative law. To accommodate such variability and increase recall, we equip Tangent-L with the ability to generate similar math features for two formulas with the same semantics.

We consider the following five classes of semantic matches:

1. *Commutativity*: $A + B$ should match $B + A$
2. *Symmetry*: $A = B$ should match $B = A$
3. *Alternative Notation*: $A \times B$ should match $A\ B$, and $A \not\succ B$ should match $A \leq B$
4. *Operator Unification*: $A \prec B$ should match $A < B$
5. *Inequality Equivalence*: $A \geq B$ should match $B \leq A$

and simple adjustments are applied to Tangent L's regular math tokens to support these semantic matches.

The adjustment to handle the first two classes, *Commutativity* and *Symmetry*, are similar. Recall that originally Tangent-L generates a math token for each pair of adjacent symbols with their orders preserved. For example, two math tokens (A, $+$, $\rightarrow$) and ($+$, B, $\rightarrow$) are generated for the expression $A + B$, and two different math tokens (B, $+$, $\rightarrow$) and ($+$, A, $\rightarrow$) are generated for the expression $B + A$. In order for an exact match to take place for the two expressions, a simple adjustment to the math tokens is to ignore the order of a pair of adjacent symbols whenever commutative operators or symmetric relations are involved. With this approach, both expressions $A + B$ and $B + A$ generate the same pair of math tokens, ($+$, A, $\rightarrow$) and ($+$, B, $\rightarrow$), so that an exact match is made possible.[4]

The next two classes, *Alternative Notation* and *Operator Unification*, can be easily accommodated by choosing a canonical symbol for each equivalence class of operators and consistently using only the canonical symbols in any math tokens generated as features.

The final class, *Inequality Equivalence*, can be handled by choosing a canonical symbol (for instance, choosing the symbol "$\leq$" in preference to "$\geq$") and then reversing the operands whenever necessary during math tokens generation.[5]

For each of these five classes of semantic matches, Tangent-L provides a separate flag to control whether or not the class is to be supported, so that only those deemed to be advantageous are applied when math tokens are generated.

2.4. Data Cleansing

For the ARQMath dataset, the original LaTeX formulas from the Math Stack Exchange collections are wrapped within an identifiable block (a `span` tag with `class="math-container"` and

[4]Our simple implementation suffers from the fact that math tokens handle only a pair of adjacent symbols at a time. For a longer expression, such as $A + B \times 5$, the overly simplistic approach generates the same set of math tokens as the expression $B + A \times 5$, failing to consider the priority of operators. nevertheless, we have chosen to take this approach because correct treatment requires that the math formulas be parsed properly, which is difficult to achieve when the input of Tangent-L—Presentation MathML—captures layout only.

[5]Similar to commutative operations and symmetric relations, the reversion of operands is implemented simplistically over a pair of adjacent symbols at a time. Thus the generated set of math tokens might equally well represent a semantically distinct formula.

Table 2
Erroneous Presentation MathML for the formula "$0.999... < 1$" (formula id 382).

Expected Presentation MathML	*Erroneous Presentation MathML Provided*

`<mrow>`
`  <mrow>`
`  <mn>0.9999</mn>`
`  <mi mathvariant="normal">...</mi>`
`  <mo>&lt;</mo>`
`  <mn>1</mn>`
`  </mrow>`
`<mrow>`
	`<mrow>`
	`  <mrow>`
	`  <mn>0.9999</mn>`
	`  <mo></mo>`
	`  <mi mathvariant="normal">...</mi>`
	`  <mo></mo>`
	`  <mi mathvariant="normal">&amp;lt;</mi>`
	`  <mo></mo>`
	`  <mi>1</mi>`
	`  <mo></mo>`
	`  <mi>t</mi>`
	`  </mrow>`
	`  <mo>;</mo>`
	`  <mn>1</mn>`
	`</mrow>`

an `id` identifier), and the corresponding Presentation MathML representations are provided as separate files. Since the input to Tangent-L includes formulas encoded in Presentation MathML, its formula matching ability will be hindered when the quality of the MathML representation is poor or conversions from LaTeX are missing.

Thanks to the effort from the Lab organizers, coverage of the Presentation MathML for detected formulas has been increased from 92% for ARQMath-1 to over 99% for ARQMath-2 [13]. However, further cleansing is still beneficial in preparation for search. We further improve the data cleansing in preparation for search as follows.

Correcting Conversion Errors: The provided Presentation MathML, generated from LaTeX representation using LaTeXML[6], contains conversion errors for formulas including either less-than "<" or greater-than ">"operators. In particular, when a LaTeX formula contains the operator "<", it is first encoded as "`<`", but then erroneously escaped again to form"`&lt;`". This results in an erroneous encoding in Presentation MathML, as shown in Table 2.

As part of our data preparation, Presentation MathML encodings with doubly-escaped representations for "<" and ">" are recognized with regular expression matching and replaced by our own converted representations, improving 869,074 ($\sim 3\%$) formulas.

Providing Missing Formula Identifiers: Approximately 10% of the annotated formulas in the postings are not correctly and completely captured, many missing their unique formula identifiers, as shown in Figure 3. In this case, our program is unable to locate their Presentation MathML representation in the file provided by the Lab organizers.

Formulas such as those from Figure 3 are recognized as much as possible through regular expression matching for text within $ and $$ blocks. These are then checked against the formula file provided by the lab organizers to reverse-trace their formula-ids. As a

[6]https://dlmf.nist.gov/LaTeXML

```
<p><span class="math-container">$\rm(2)\quad  1+J \subseteq U,\quad\ \ $</span>  i.e. <span class="math-
container">$\rm\: 1 + j\:$</span>  is a unit for every <span class="math-container">$\rm\: j \in J$</span></p>
<p><span class="math-container">$\rm(3)\quad  I\neq 1\ \Rightarrow\  I+J \neq 1,\qquad\ $</span>  i.e.  proper
ideals survive in <span class="math-container">$\rm\:R/J$</span></p>
<p><span class="math-container">$\rm(4)\quad M\:$</span> max <span class="math-container">$\rm\:\Rightarrow  M+J
\ne 1,\quad $</span>  i.e. max ideals survive in <span class="math-container">$\rm\:R/J$</span></p>
<p><strong>Proof</strong> <span class="math-container">$\:$</span> (sketch) <span class="math-container"
id="2276">\ </span>  With  <span class="math-container">$\rm\:i \in I,\ j \in J,\:$</span> and max ideal <span
class="math-container">$\rm\:M,$</span></p>
```

Figure 3: Partial text from an answer post (post-id 2653) including "math-container" blocks without "id" attributes, even though the corresponding formulas are included in the formula file with formula-ids from 2285 to 2296.

result, our program is able to capture over 99% of the formulas, including the 10% that are improperly represented in math-container blocks without ids.

3. Task 1: Finding Answers to Math Questions

In Task 1, participants are given mathematical questions selected from MSE posts from either year 2019 (for ARQMath-1) or year 2020 (for ARQMath-2). Each question is formatted as a *topic* that contains a unique identifier, the title, the question body text, and the tags. Participant systems are asked to return the top-1000 potential answer-posts for each of the topics from the MSE collection.

For ARQMath-2, we continue to use the three-stage system adopted for ARQMath-1 [9]:

Stage 1 *Conversion*: Transform the input (a mathematical question posed on MSE) into a well-formulated query consisting of a *bag* of formulas and keywords.

Stage 2 *Searching*: Use Tangent-L, the math-aware search engine, to execute the formal query to find the best matches against an indexed document corpus created from the collection.

Stage 3 *Re-ranking*: Re-order the best matches with a run-specific re-ranking model.

In this section, we describe various modifications we wished to explore. We first validate the benefits of each modification using the ARQMath-1 benchmark, and then we test them using the ARQMath-2 benchmark.

3.1. Conversion: Fine-tuning Keyword Extraction from Formulas

For ARQMath-1, our designed automated mechanism used to extract query keywords and formulas from the task *topics* was shown to be competitive with the human ability to select search terms [9], as it produces an result that is comparable to the manual set of query terms selected by the Lab organizers. For ARQMath-2, we fine-tune this automated mechanism using the ARQMath-1 benchmark for validation as follows:

1. Keywords *within a formula representation* are intentionally[7] retained and extracted, as a drop in nDCG′ occurs if they are removed. For example, "mod" is a crucial keyword for topic $A.7$—*Finding out the remainder of* $\frac{11^{10}-1}{100}$ *using modulus* – but this word is present within a formula representation only and not anywhere else in the text. Similarly, "sin", "cos", "tan" can be extracted from `\sin`, `\cos`, `\tan` in formula representations after punctuation is removed.
2. Every term extracted by the automated mechanism should become part of the query, and their weight should be boosted naturally if they repeat.[8] On the other hand, restricting the number of keywords and formulas extracted from the mechanism (as we had hypothesized to be a possible improvement last year) does *not* show an improved result.

After fine-tuning the automated mechanism, results obtained for the ARQMath-1 benchmark can be observed to consistently outperform those obtained with the manual set of query terms, validating the potential of this mechanism.

3.2. Searching: Enriching the Document Corpus

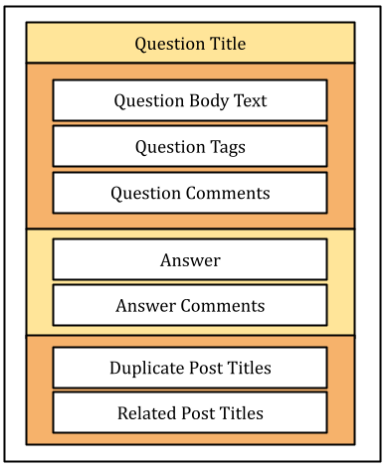

Figure 4: An illustration of the revised indexing unit to create the document corpus. Each document is an HTML file containing a question-answer pair and its associated information.

For ARQMath-2, we continue to use question-answer pairs as indexing units for the document corpus, as worse performance results for the ARQMath-1 benchmark if the content of the associated question is dropped and only text from each answer is indexed. In addition to the fields included for ARQMath-1, comments[9] associated with answers are also included. As a

[7] Keywords were not intended to be extracted from within formula representations in the original design for ARQMath-1, but turned out to be a valuable "mistake" that helped boost performance.

[8] In the submission for ARQMath-1, duplicate terms were extracted, but their weights were not boosted accordingly because of an oversight in our implementation.

[9] When extracting the comments, the file *Comments.V.1.0.xml* is used instead of the more recently released *Comments.V.1.2.xml* because the former contains approximately three times as many comments as the latter. Note, however, that the former file contains more "noise" that requires cleansing as discussed in Section 2.4.

Table 3
Various proximity measures [14], each of which can also be normalized by document length.

Span:	length of the shortest document segment that covers all query term occurrences in a document, including repeated occurrences
Normalized-Span:	length of the shortest document segment that covers all query term occurrences in a document, including repeated occurrences, divided by the number of matched instances
Min-Span:	length of the shortest document segment that covers each matched query term at least once in a document
Normalized-Min-Span:	length of the shortest document segment that covers each matched query term at least once in a document, divided by the number of matched query terms
Min-Distance:	smallest distance value of all pairs of unique matched query terms
Ave-Distance:	average distance value of all pairs of unique matched query terms
Max-Distance:	largest value of all pairs of unique matched query terms

Table 4
Comparison of proximity measures on the ARQMath-1 benchmark for highly relevant (HR), relevant (R), partially relevant (PR), and non-relevant (NR) math answers, where $\Delta(a,b) = \frac{\text{prox}(a)-\text{prox}(b)}{0.5(\text{prox}(a)+\text{prox}(b))}$.

	Δ(HR,R)	Δ(R,PR)	Δ(PR,NR)	Δ(R,NR)	Δ(HR,NR)
Span	7%	8%	3%	10%	18%
Span-NormByDocLen	0%	1%	5%	5%	5%
Normalized-Span	-5%	-6%	-62%	-67%	-72%
Normalized-Span-NormByDocLen	-20%	-13%	-64%	-76%	-92%
Min-Span	9%	7%	6%	13%	21%
Min-Span-NormByDocLen	-1%	2%	8%	11%	10%
Normalized-Min-Span	2%	1%	-39%	-38%	-36%
Normalized-Min-Span-NormByDocLen	-11%	-3%	-40%	-43%	-53%
Min-Distance	1%	-2%	-89%	-90%	-89%
Min-Distance-NormByDocLen	-10%	-9%	-104%	-111%	-117%
Ave-Distance	4%	3%	-16%	-14%	-10%
Ave-Distance-NormByDocLen	-7%	-2%	-15%	-17%	-24%
Max-Distance	9%	7%	6%	13%	21%
Max-Distance-NormByDocLen	-1%	2%	9%	11%	10%

result, more formulas and more text words are available for matching. Figure 4 illustrates the fields indexed as part of each question-answer pair.

3.3. Re-ranking: Proximity

Whereas in ARQMath-1 we attempted re-ranking the retrieved answers from Tangent-L based on CQA metadata, for ARQMath-2 we investigate the possibility of re-ranking based on proximity. Proximity is a measure of distance between matched query terms as detailed in Table 3, which can be a strong signal for document relevancy.

Following the experimental design used by Tao and Zhai [14], we measure the average

proximity of search terms for highly relevant, relevant, partially relevant, and non-relevant documents in the ARQMath-1 benchmark. The experimental result is shown in Table 4. We observe strong signals from several measures that distinguish relevance with the correct order (marked in gradient orange), particularly for normalized-span which correctly orders all four levels of relevancy (a smaller normalized-span indicating a higher level of relevancy) without the need to be normalized by document length.

Motivated by this finding, for ARQMath-2 we attempt re-ranking of the retrieved answers by Tangent-L in increasing order of normalized-span, breaking ties by a decreasing $\mathrm{BM25}^{+}$ score returned from Tangent-L.

3.4. Matching Formulas Holistically

Formula matching within Tangent-L is based on comparing a set of math tokens from the query to those from each document (Equation 1). If we index a document that has multiple formulas, math tokens generated from all the formulas within the document are considered as a single unordered bag of terms. However, given the strong signal of proximity playing a role in document relevancy (Table 4), we hypothesize that matching each formula as a whole within a document, instead of matching math tokens irrespective of formulas that might scatter across a document, could produce a better result.[10] As such, as a post-experiment we design a *holistic* formula search as follows:

At preparation time, we first pre-build a *formula corpus* for Tangent-L that indexes all visually distinct formulas in the MSE dataset, each as a separate document with the formula's *visual-id* serving as a key. We define the *formula similarity* between two formulas to be the *normalized* $\mathrm{BM25}^{+}$ score for one formula when the other formula acts as a query. When indexing the question-answer corpus, rather than replacing each formula within the document by the set of math tokens generated for that formula, we represent each formula by a single *holistic formula token* that contains the formula's visual-id (that is, its key from the formula corpus). At query time, we first search for each query formula in the formula corpus and then replace the formula text in the query by the keys of the top-k most similar formulas, thus changing the query to search for those visual-ids (as well as whatever keywords are also part of the query, of course). Finally, the *ranking formula* for documents is revised to weight each match of a formula id by its formula similarity with respect to the original query formula.

In the following subsections, we describe these ideas in greater detail.

3.4.1. Formula Corpus

The formula corpus is built by extracting all *visually distinct* formulas from the document corpus described in Section 3.2—including formulas found within questions, answers, and comment posts. Each formula in this corpus is associated with the formula's *visual-id*, which serves as a key. The resulting corpus contains 8,595,899 out of 9,329,274 ($\sim 92\%$) visually distinct formulas and is indexed by Tangent-L under the setup described in Section 2, each formula being considered as a document.

[10] Note, however, that this ignores proximity among keywords and between keywords and formulas.

3.4.2. Normalized Formula Similarity

We define "formula similarity" as follows: Let f_q be an arbitrary formula used as a query, F be the set of formulas in the formula corpus, and $f \in F$. Let $\mathrm{RawScore}(f_q, f)$ represents the score obtained for formula f when the query is f_q, using the following definition:

$$\mathrm{RawScore}(f_q, f) = (1-\gamma) \cdot \mathrm{BM25}^{+}(q_m, f) + \gamma \cdot \mathrm{BM25}^{+}(q_r, f) \tag{2}$$

where q_m is the set of regular math tokens and q_r is the set of repetition tokens in a query formula f_q. As in Equation 1, $0 \leq \gamma \leq 1$ balances the weight of repetition tokens against regular math tokens.

The *Normalized Formula Similarity* of f with respect to f_q is:

$$N(f, f_q) = \frac{\mathrm{RawScore}(f_q, f)}{\max_{\varphi \in F} \mathrm{RawScore}(f_q, \varphi)} \tag{3}$$

The value of $N(f, f_q)$ is in the range [0,1] and represents how well the query formula f_q is matched by f relative to other formulas within the formula corpus.

3.4.3. Holistic formula token

A holistic formula token is a placeholder token that incorporates the formula's *visual-id*. Formulas in a question-answer document are replaced by their holistic formula tokens only, so that when searching the question-answer corpus, formulas can only be matched as a whole.

3.4.4. Ranking for Holistic Search

Let q_t be the set of keyword tokens and q_f be the set of query formulas. Let $f_q \in q_f$ be a query formula and let $S_k(f_q)$ be the set of keys for the top-κ most similar formulas with respect to f_q, determined by Normalized Formula Similarity. Let d be a document represented by the set of all its indexed tokens.

When searching the document corpus, we adopt the following variant of $\mathrm{BM25}^{+}$:

$$\mathrm{BM25}^{+}_{\mathrm{w}}(q_t \cup q_f, d) = (1-\alpha) \cdot \mathrm{BM25}^{+}(q_t, d) + \alpha \cdot \mathrm{BM25}^{+}(q_f, d) \tag{4}$$

and

$$\mathrm{BM25}^{+}(q_f, d) = \sum_{f_q \in q_f} \sum_{f \in (d \cap S_k(f_q))} \left(N(f, f_q) \cdot \frac{(k+1) tf_f}{k\left(1.0 - b + b\frac{|d|}{\overline{d}}\right) + tf_f} + \delta \right) \log\left(\frac{|D|+1}{|D_f|}\right) \tag{5}$$

where, as in Equation 1, $0 \leq \alpha \leq 1$ balances math features against keyword features.[11]

3.5. Task 1: Runs and Result

Parameter settings are chosen based on testing with the ARQMath-1 benchmark. For ARQMath-2, we prepared four automatic runs:

[11] As usual for $\mathrm{BM25}^{+}$ [15], k, b, and δ are constants (following common practice, chosen to be 1.2, 0.75, and 1, respectively); tf_f is the number of occurrences of formula f in d; $|d|$ is the total number of terms in d; $\overline{d} = \sum_{d \in D} \frac{|d|}{|D|}$ is the average document length; and $|D_f|$ is the number of documents in D containing formula f.

Table 5
The setup for the primary run for ARQMath-2.

Repeated Symbols	(Sect. 2.1)	Repetition tokens are adopted.
Revised Ranking Formula	(Sect. 2.2)	In Equation 1, $\alpha = 0.25$ and $\gamma = 0.1$.
Formula Normalization	(Sect. 2.3)	Only semantic matches of *Commutativity* is supported.
Data Cleansing	(Sect. 2.4)	Recognition of Presentation MathML is improved.
Document Corpus	(Sect. 3.2)	Comments from answers are added to the indexing unit.
Query Keyword Extraction	(Sect. 3.1)	Keywords within a formula representation are retained. Query terms are (*unintentionally*) de-duplicated.

primary: A submitted run with *most of* the presumably best setup, based on tests on the ARQMath-1 benchmark, as described in Table 5.

proximityReRank: A submitted run based on Section 3.3. This uses the same setup as the primary run, but the top-1000 matches are subsequently re-ranked by proximity, using normalized span as the proximity measure.

holisticSearch: A post-experiment run that matches formulas holistically based on Section 3.4. When searching in the formula corpus, γ is set to 0.1 in Equation 2 and when searching in the document corpus, α is set to 0.5 in Equation 4 and κ is set to be 300.

duplicateTerms: A post-experiment run sharing the same setup as the primary run, except that duplicate query terms are *preserved* as described in Section 3.1.

The results of these runs for ARQMath-2 are shown in Table 6, together with the baseline runs and our submissions from last year over the ARQMath-1 benchmark. In general, after parameter selection based on the ARQMath-1 benchmark, our updated system produces results that have a significant improvement compared with those from last year's system over the ARQMAth-1 topics. For instance, our *primary* setup evaluated over the ARQMath-1 benchmark achieves an nDCG' score of 0.433, which is nearly a 10-point gain over the nDCG' score of 0.345 produced by our best participant run (*alpha05-noR*) last year.

This parameter selection based on the ARQMath-1 benchmark helps our updated system to achieve equally good results for the new set of math topics in ARQMath-2. Our *primary* run produces an nDCG' of 0.434, which remains the best run among all participants[13]. The unsubmitted run *duplicateTerms*, which corrects an oversight in the *primary* run and therefore reflects our intended "best" setup, scores even higher, with an nDCG' of 0.462.

The *duplicateTerms* run also has the highest values for the ARQMath-2 benchmark in all other evaluation measures, with the exception of $\mathrm{P}'@10$ for the baseline run *Linked MSE posts* (which uses human-built links that were not available to participating teams[13]). With a closer look to the effectiveness breakdown by topic category in Table 7, we observe that this run has a strong performance for Formula-dependent topics, Proof-like topics, and topics of Low-level difficulty. In spite of a different set of math topics being evaluated, these observed strengths are similar to the observed strengths of our best participant run last year [3].

On the other hand, our submitted alternative run *proximityReRank*, which tries to re-rank the results using the proximity signal *Normalized-Span*, does not perform well. For the ARQMath-1

Table 6
Task 1: Evaluation of the MathDowsers runs and the baseline runs in ARQMath-2, compared with that over the ARQMath-1 benchmark. Parentheses indicate a result from an approach using privately held data not available to participants.

		ARQMath-1 (77 Topics)				*ARQMath-2 (71 Topics)*			
		nDCG′	*MAP′*†	*P′@10*†	*bpref*†	*nDCG′*	*MAP′*†	*P′@10*†	*bpref*†
Baselines									
Linked MSE posts		(0.279)	(0.194)	(0.386)	(0.214)	(0.203)	(0.120)	(0.282)	(0.131)
TF-IDF + Tangent-S		0.248	0.047	0.073	0.044	0.201	0.045	0.086	0.048
TF-IDF		0.204	0.049	0.074	0.043	0.185	0.046	0.063	0.046
Tangent-S		0.158	0.033	0.051	0.033	0.111	0.027	0.052	0.039
MathDowsers									
duplicateTerms		**0.457**	**0.207**	**0.267**	**0.190**	**0.462**	**0.187**	**0.241**	**0.163**
primary	¶	0.433	0.191	0.249	0.178	0.434	0.169	0.211	0.145
holisticSearch		0.405	0.192	0.266	0.181	0.414	0.167	0.225	0.150
proximityReRank	∗	0.373	0.117	0.131	0.095	0.335	0.081	0.049	0.052
MathDowsers (year 2020)									
alpha05-noR	∗	0.345	0.139	0.162	0.126	-	-	-	-
alpha02	∗	0.301	0.069	0.075	0.044	-	-	-	-
alpha05-trans	∗𝕄	0.298	0.074	0.079	0.050	-	-	-	-
alpha05	¶	0.278	0.063	0.073	0.041	-	-	-	-
alpha10	∗	0.267	0.063	0.079	0.042	-	-	-	-

¶ submitted primary run ∗ submitted alternate run 𝕄 manual run † using H+M binarization

benchmark, this run shows a 6-point loss compared to the *primary* run (0.373 vs 0.433) and the loss is enlarged to nearly 10 points in ARQMath-2 (0.335 vs 0.434), indicating an unsatisfactory re-ranking. It seems that even for a measure that shows a strong signal for proximity in Table 4, the separation among documents based on proximity might be inadequate to reflect relevance.

Finally, our unsubmitted run *holisticSearch*, which is an approach also motivated by proximity, performs fairly well. Compared to the *primary* run, the nDCG' score for the *holisticSearch* run shows a 3-point loss over the ARQMath-1 benchmark (0.405 vs 0.433) and similarly a 2-point loss in ARQMath-2 (0.414 vs 0.434). Notably, this run outperforms all other runs submitted by participants in ARQMath-2 and outperforms our primary run in the $P'@10$ and *bpref* measures. However, this run is outperformed by the unsubmitted *duplicateTerms* run in all evaluation measures (with nearly a 5-point loss (0.414 vs 0.462 for nDCG'), suggesting room for improvement for this approach.

4. Task 2: In-context Formula Retrieval

For Task 2, participants are asked to retrieve the top matching formulas, together with their associated posts, for each *topic formula* chosen from the set of topics used for Task 1. Relevancy of a retrieved formula is evaluated in context: both the associated post of a retrieved formula and

Table 7
Category performance of the *duplicateTerms* run in ARQMath-2. The better performance measure for each sub-category and each evaluation measure is highlighted in bold.

	Topic Count	*duplicateTerms* *nDCG′*	*MAP′*	*P′@10*	*bpref*
Overall	71	0.462	0.187	0.241	0.163
Dependency					
Text	10	0.423	0.158	0.260	0.142
Formula	21	**0.516**	**0.235**	**0.319**	**0.204**
Both	40	0.443	0.169	0.195	0.146
Topic Type					
Calculation	25	0.455	0.189	0.200	0.165
Concept	19	0.429	0.160	0.232	0.137
Proof	27	**0.492**	**0.204**	**0.285**	**0.178**
Difficulty					
Low	32	**0.509**	**0.216**	**0.300**	**0.199**
Medium	20	0.383	0.116	0.150	0.098
Hard	19	0.466	0.213	0.237	0.169

the associated topic content of the topic formula are presented to the assessors for evaluation. Assessments are then aggregated so that each *visually distinct* formula is judged to be relevant if any of the corresponding formula occurrences are deemed to be relevant. The performance of a system is then determined by its performance with respect to visually distinct formulas only.

For ARQMath-2, we propose two simple approaches that re-use two major components created for Task 1:

1. the Formula Corpus of all visually distinct formulas, as described in Section 3.4.1;
2. the results from Task 1 Answer-Ranking of the top 10,000 answer-posts for each topic, run with the primary setup as detailed in Table 5.

The rest of this section describes our two approaches built on these components.

4.1. Formula-centric: Selecting Visually Matching Formulas

The first straightforward approach is formula-centric, relying on Tangent-L's internal formula matching capability to find the matching formulas. To create a list of matching formulas for a topic, we first search for matches to the topic formula in the formula corpus of all visually distinct formulas. This gives us a ranking R of visually distinct formulas. We then expand each element of R with its set of formula occurrences: formulas that have the same visual-id but appearing in different posts.[12] We refer to a set of formula occurrences having the same

[12]Only question-posts and answer-posts are of concern in the task, so any returned formulas from comment-posts are ignored.

visual-id as a *visual group*. The selection of formula occurrences to return is then governed by the rank of their associated posts in the answer retrieval task. In particular,

1. Formulas within the same visual group are ranked in the same order as the ranking of their associated posts in Task 1 for the corresponding topic. If the associated posts of formulas are question-posts that are not associated with any answer from Task 1, the formulas are assigned the lowest ranking. Finally, the lexical order of formula-ids is used to break ties.
2. For each of the top-20 visually distinct formulas in R, we select the top five formulas from its visual group (or all formulas in the visual group if there are fewer than five); for the remainder, we select the top formula only (if any have associated question or answer posts).
3. Sequentially considering the formulas in R in order, selected formula occurrences from each visual group are appended to the final list of matching formulas until 1000 formula occurrences are selected in total.

4.2. Document-centric: Screening Formulas from Matched Documents

The second straightforward approach is document-centric, relying more on the results from the answer retrieval task. Based on the answer-ranking from Task 1, the final list of matching formula occurrences is selected from the answers as follows:

1. For each matched answer-post for the corresponding topic in Task 1, we retrieve its question-answer document from the document corpus. If the document contains only one formula, that formula is selected. Otherwise, each formula from the document is mapped to its visual group, and its *Normalized Similarity Score* (Equation 3) with respect to the topic formula is computed using $\gamma = 0.1$ in Equation 2 (but see below). Formulas having a score less than a threshold of 0.8 are screened out, and the rest are preserved and ranked accordingly.
2. Following the original answer-ranking, preserved formulas from each question-answer pair are appended to the final list until 1000 formulas are selected in total.

Formulas in an answer-post might correspond to visually distinct formulas any where in the formula corpus, but it is highly inefficient to compute the Normalized Similarity Score for every formula in the formula corpus, which requires retrieving over 8.5 million RawScores using Tangent-L. Therefore, for each topic, formulas in answer-posts that are not within the top 10,000 most similar formulas to the query formula are assigned a score of 0 and therefore screened out.

4.3. Task 2: Runs and Result

For ARQMath-2, we include two automatic runs:

formulaBase: A submitted run selecting visually matching formulas as in Section 4.1;

docBase: A submitted run selecting formulas from matched documents as in Section 4.2.

Table 8
Task 2: Evaluation of MathDowsers runs, the best participant runs, and baseline runs in ARQMath-2.

		ARQMath-1				*ARQMath-2*			
		nDCG′	*MAP′*†	*P′@10*†	*bpref*†	*nDCG′*	*MAP′*†	*P′@10*†	*bpref*†
Baselines									
Tangent-S		0.691	0.446	0.453	0.412	0.492	0.272	0.419	0.290
MathDowsers									
formulaBase	¶	0.562	0.370	0.447	0.374	0.552	0.333	0.450	0.348
docBase	∗	0.404	0.251	0.386	0.275	0.433	0.257	0.359	0.291
Best Participant Run									
Approach0-P300	∗𝕄	0.507	0.342	0.441	0.343	**0.555**	**0.361**	**0.488**	**0.362**
DPRL-ltrall	¶	**0.738**	**0.525**	**0.542**	**0.495**	0.445	0.216	0.333	0.228
Best Participant Run (year 2020)									
DPRL-Tangent-CFTED *		0.563	0.388	0.436	0.372	-	-	-	-

¶ submitted primary run ∗ submitted alternate run 𝕄 manual run † using H+M binarization

The result of both runs in ARQMath-2 are shown in Table 8, together with the baseline run and the best participant runs for the ARQMath-1 and ARQMath-2 benchmarks. Our primary run *formulaBase*, with parameter selection based on the ARQMath-1 benchmark, achieves a very close performance to the best participant run *Tangent-CFTED* produced from the DPRL team last year (0.562 vs 0.563). However, on the ARQMath-1 benchmark, it does not perform as well as the *ltrall* run submitted this year by the DPRL team, having a 17-point loss on nDCG' over the same set of math topics (0.562 vs 0.735).

On the ARQMath-2 benchmark, however, with a new set of math topics, our primary run *formulaBase* performs approximately as well, with an nDCG' score of 0.552. This score is the best among all automatic runs, and it is almost indistinguishable from the best participant run *P300* from the Approach0 team, which is a manual run. Notably, on the ARQMath-2 benchmark, it outperforms the *ltrall* run from the DPRL team by over 10 points (0.552 vs 0.445).

On the other hand, our alternative run *docBase* does not perform as well as expected. For the ARQMath-1 benchmark, this run shows nearly a 16-point loss with respect to our primary run (0.404 vs 0.562) and nearly a 12-point loss (0.433 vs 0.552) for the ARQMath-2 in terms of nDCG'. This run also achieves lower scores in all other evaluation measures, suggesting that simply selecting formulas from matching documents does not work well.

5. Efficiency

The machines used for our experiments have the following specifications:

Machine A	A Ubuntu 20.04.1 LTS Server with an AMD EPYC™ 7502P Processor (32 Cores 64 Threads, 2.50GHz), 512GB RAM and 3.5TB disk space.
Machine B	A Linux Mint 19.1 Server with an Intel Core i5-8250U Processor (4 Cores 8 Threads, up to 3.40GHz), 24GB RAM and 512GB disk space.

All indexing was performed on Machine A, yielding the following performance characteristics:

Corpus	See Section	Data Size (GB)	Index Size (GB)	Indexing Speed (sec)
Document Corpus	3.2	23	4.1	4394
Formula Corpus	3.4.1	34	4.7	4834
Document Corpus (for holistic search)	3.4.3	23	0.6	167

Note that data and index sizes show the values reported by the du command on Linux, which measures disk space usage based on blocks; thus the many small documents in the formula corpus require much more disk space than might be expected. (In fact, the total size of the data in the formula corpus is only 9.2 GB.)

Runs for ARQMath-2 were executed on Machine B with the following average, minimum, and maximum query times per topic as follows:

Run \ Query Time	Avg. (sec)	Min. (sec)	Max. (sec)
Task 1			
primary	1.90	0.34 (A.264)	6.39 (A.221)
holisticSearch	7.77	2.37 (A.264)	24.5 (A.221)
duplicate	1.92	0.30 (A.264)	6.04 (A.272)
Task 2			
(pre-computing Answer-Ranking)	1.94	0.48 (A.264)	6.03 (A.221)
formulaBase	1.16	0.22 (B.244)	3.79 (B.270)
docBase	56.5	16.5 (B.209)	122 (B.221)

The proximityReRank run uses Machine A to rerank the output from the primary run, thus requiring first the time shown for the primary run on Machine B and then an additional 8 hours to re-rank all topics on Machine A.

6. Conclusions and Further Work

We conclude that a traditional math-aware search system continues to be an efficient and effective approach to tackle the CQA task, which is proven by producing the best participant run in Task 1 again this year. In particular, a significant boost in effectiveness for Task 1 can be observed on both years' math topics after parameter selection based on tests on the ARQMath-1

benchmark. The best result is achieved through several aspects of improvement of the formula matching capability of Tangent-L, demonstrating the competitiveness of this math-aware search engine in handling text and mathematical notations together.

We also develop a simple but strong baseline for the in-context formula retrieval task. Being the best automatic run and competitive with the best participant run, our formula-centric run demonstrates again the strong formula matching ability of Tangent-L.

Nevertheless, several aspects of our runs turn out to be somewhat disappointing again. In the CQA task, we explore the incorporation of proximity in two approaches and the result does not improve effectiveness over using a bag-of-terms approach:

Proximity Re-Ranking: Re-ranking based on proximity is unsatisfactory, despite some proximity difference being observed based on the relevancy of judged documents. Perhaps proximity is a more important measure when the $\mathrm{BM25}^{+}$ score is low, and therefore it needs to be incorporated into the initial retrieval [14, 16] rather than used for re-ranking. Alternatively, despite the percentage differences observed, the actual differences might be too small to serve as a reliable signal of relevance.

Matching Formulas Holistically: The proposed method to match formulas holistically shows some promise but does not perform as well as matching based on math tokens. Perhaps Equation 5 can be improved to make better use of the formula similarity scores returned from the formula corpus. Improvements here might also provide insights into further improving our formula-centric approach in Task 2.

Additionally, our proposed document-centric baseline for the in-context formula retrieval task, which selects formulas from top matching math answers, does not perform as well as expected given our strong result in the answer retrieval task. Investigation into the distribution of matching formulas among the top relevant answers might be helpful in further exploring this simple tactic for the task.

All in all, while our updated system with Tangent-L continues to excel in both tasks, there is still a huge room for improvement in how we might use the document relevancy signals observed from the ARQMath-1 benchmark to propose new approaches that might further improve effectiveness. In retrospect, approaches that we attempted through re-ranking did not benefit sufficiently from the raw signals obtained from the ARQMath-1 benchmark. With the additional new evaluation data available from the ARQMath-2 benchmark, we expect to gain better insights, and we are excited to continue exploring question answering for the mathematical domain.

Acknowledgments

This research has been funded by the Waterloo-Huawei Joint Innovation Lab and NSERC, the Natural Science and Engineering Research Council of Canada. The NTCIR Math-IR dataset used for earlier benchmarks and as a source of relevant keywords was made available through an agreement with the National Institute of Informatics.

References

[1] R. Zanibbi, B. Mansouri, D. W. Oard, A. Agarwal, Overview of ARQMath-2 (2021): Second CLEF lab on answer retrieval for questions on math, in: CLEF 2021, volume 12880 of *LNCS*, 2021.

[2] R. Zanibbi, D. W. Oard, A. Agarwal, B. Mansouri, Overview of ARQMath 2020 (updated working notes version): CLEF lab on answer retrieval for questions on math, in: CLEF 2020, volume 2696 of *CEUR Workshop Proceedings*, 2020.

[3] Y. K. Ng, D. J. Fraser, B. Kassaie, G. Labahn, M. S. Marzouk, F. W. Tompa, K. Wang, Dowsing for math answers with Tangent-L, in: CLEF 2020, volume 2696 of *CEUR Workshop Proceedings*, 2020.

[4] B. Mansouri, D. W. Oard, R. Zanibbi, DPRL Systems in the CLEF 2020 ARQMath Lab, in: CLEF 2020, volume 2696 of *CEUR Workshop Proceedings*, 2020.

[5] V. Novotný, P. Sojka, M. Štefánik, D. Lupták, Three is Better than One Ensembling Math Information Retrieval Systems, in: CLEF 2020, volume 2696 of *CEUR Workshop Proceedings*, 2020.

[6] S. Rohatgi, J. Wu, C. L. Giles, PSU at CLEF-2020 ARQMath Track: Unsupervised Re-ranking using Pretraining, in: CLEF 2020, volume 2696 of *CEUR Workshop Proceedings*, 2020.

[7] P. Scharpf, M. Schubotz, A. Greiner-Petter, M. Ostendorff, O. Teschke, B. Gipp, ARQMath Lab: An Incubator for Semantic Formula Search in zbMATH Open?, in: CLEF 2020, volume 2696 of *CEUR Workshop Proceedings*, 2020.

[8] D. J. Fraser, A. Kane, F. W. Tompa, Choosing math features for BM25 ranking with Tangent-L, in: DocEng 2018, 2018, pp. 17:1–17:10.

[9] Y. K. Ng, D. J. Fraser, B. Kassaie, F. W. Tompa, Dowsing for math answers, in: CLEF 2021, volume 12880 of *LNCS*, 2021.

[10] A. Białecki, R. Muir, G. Ingersoll, Apache Lucene 4, in: SIGIR 2012 Workshop on Open Source Information Retrieval, 2012, pp. 17–24.

[11] R. Zanibbi, D. Blostein, Recognition and retrieval of mathematical expressions, Int. J. Document Anal. Recognit. 15 (2012) 331–357.

[12] Y. Lv, C. Zhai, Lower-bounding term frequency normalization, in: CIKM'11, 2011, pp. 7–16.

[13] B. Mansouri, A. Agarwal, D. W. Oard, R. Zanibbi, Advancing math-aware search: The ARQMath-2 lab at CLEF 2021, in: ECIR 2021, volume 12657 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 631–638.

[14] T. Tao, C. Zhai, An exploration of proximity measures in information retrieval, in: SIGIR 2007, 2007, pp. 295–302.

[15] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389.

[16] Y. Rasolofo, J. Savoy, Term proximity scoring for keyword-based retrieval systems, in: Advances in Information Retrieval, Proceedings of the 27th European Conference on IR Research (ECIR 2003), volume 2633, Springer, 2003, pp. 207–218.