Chapter 7

Examination of the state of the art of mathematical formula search for zbMATH Open

Johannes Stegmüller, André Greiner-Petter, Petr Sojka, Olaf Teschke, and Moritz Schubotz

The service for abstracting and editing mathematical content zbMATH Open, which also offers a formula search, is constantly being developed. Since the beginning of 2021, zbMATH Open has been open for public access. To leverage the opportunities of some recent developments in zbMATH Open and for formula search, we examine the state of the art in math search engines and their applications. Also, based on our investigation, we present several proposals for improvements to the formula search of zbMATH Open.

1 Introduction

zbMATH Open[1] (shorter zbMATH, formerly Zentralblatt MATH) is an abstracting and reviewing service for mathematical content. At the time of writing, it contains 4.3 million bibliographic entries with publication years between 1826 and 2022. There have been 1,123,159 reviews since 1868 by the community of reviewers, which currently counts 7,677 active associates.

The publicly available web-interface of zbMATH offers specialised search opportunities for finding entries in the huge collection of mathematical publications. Entries can be found by specifying keywords that refer to information about the document, its author or its classification in the MSC2020 [5] as well as other attributes linked to the document. A significant attribute in the context of this work is the search of documents by specifying mathematical formulae. Currently, over 160 million formulae are indexed. The first prototype for a formula search in zbMATH was established in a research collaboration between FIZ Karlsruhe and the Jacobs University Bremen.[2]

Since recent developments in 2021, all zbMATH content is openly available for free to the public domain. Open interfaces enable the integration of other services, e.g., better search functions for full texts from free digital libraries such as arXiv and EuDML. Opening up the content offers another dimension of new applications by linking it to mathematical research data that has been largely isolated and inadequately tapped in the past [18].

---

[1] https://zbMATH.org
[2] https://zbMATH.org/formulae

Due to the current progress of zbMATH and to constantly improving the formula search for the mathematical community, we present the current work. In this work, we investigate various math search engines and formula-search-applications to outline the state-of-the-art in mathematical formula search. Furthermore, we check the results of our investigation for their applicability to the extension of formula search in zbMATH. On the foundation of our investigation, we present suggestions for several extensions.

This work is organized as follows. First, in Section 2 we provide an overview of math search engines and applications for formula search and point out their major attributes. At the start of the section, we provide a summary of the currently used math search engine, *MathWebSearch*.

In Section 3 we propose extensions for formula search in zbMATH based on the investigation of the search engines and applications in the previous section.

Section 4 renders a brief synopsis, concludes our paper and gives an outlook on the future.

2 Overview of math search applications and engines

In this section, we follow the example of [8, 19] and test real-world math search engines and applications related to formula search with functional demos. For math search engines without a functional demo, we consult literature that evaluates them.

2.1 MathWebSearch

2.1.1 Description. The MathWebSearch system (MWS)[3] is the math search engine currently utilized in zbMATH. MWS was actively developed by the KWARC group at Jacobs University, mostly by C. Prodescu, until the current latest release in 2014.[4]

2.1.2 Functionalities. MWS is a content-based full-text search engine that concentrates on low-latency query responses in interactive applications. It combines exact formula matching with full-text search capabilities for simultaneous search for keywords and formulae. For the full-text queries, it uses Elasticsearch [10].

For creating the search engine index, MWS reads *Content MathML* formatted formulae using a technique derived from automated theorem proving called substitution tree indexing [7].

Most scientific publications (e.g. the arXiv corpus) in STEM fields use LaTeX formula notation [9]. Since LaTeX denotation is quite common, it is used in the web-interface of zbMATH for the input of formulae. To create suitable Content MathML

[3] https://search.mathweb.org

[4] https://github.com/MathWebSearch/mws

queries and enable indexing for input documents, the LaTeXML converter developed by Bruce Miller[5] at NIST is utilized for LaTeX conversion to MathML [15].

MWS offers system components for multiple stages in the process of enabling formula search [10]: A MWS component enables parsing HTML and XHTML documents to annotated XML, which contains document metadata as well as the formula encoded as Content MathML. The annotated XML is read from a folder by the Formula Indexer and a formula search index is created. The Indexer also provides an RESTful API for formula query. For text-based document queries, Elasticsearch is used. For concurrent keyword and formula queries, Elasticsearch and the Indexer are prepended with a proxy. The proxy prioritizes the proportion of search hits in keyword and formula query responses in the final response.

2.1.3 Review. The core engine of MWS is not actively developed. The last MWS release was in December 2014 and since then mainly support for containerisation has been added to the codebase. The source code for MWS is publicly available on GitHub and licensed under GPLv3. In 2014 KWARC was participating with MWS 1.0 in the NTCIR-11 Math-2 Task, which is specially dedicated to information access to mathematical content. It scored above average for retrieval precision among eight task participants [2].

2.2 MathDeck

2.2.1 Description. MathDeck[6] is a math search related application which aims to offer simplified methods for entering formulae [6]. Entered formulae are rendered and can be exported as image files or their LaTeX notation can be forwarded to common math search engines.

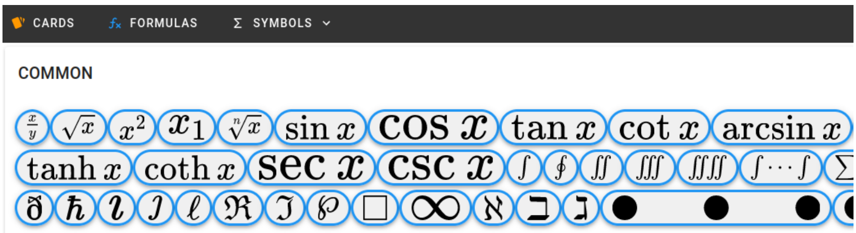

Figure 1. Symbol palette for well-known maths symbols in MathDeck.

[5]https://dlmf.nist.gov/LaTeXML
[6]https://mathdeck.org

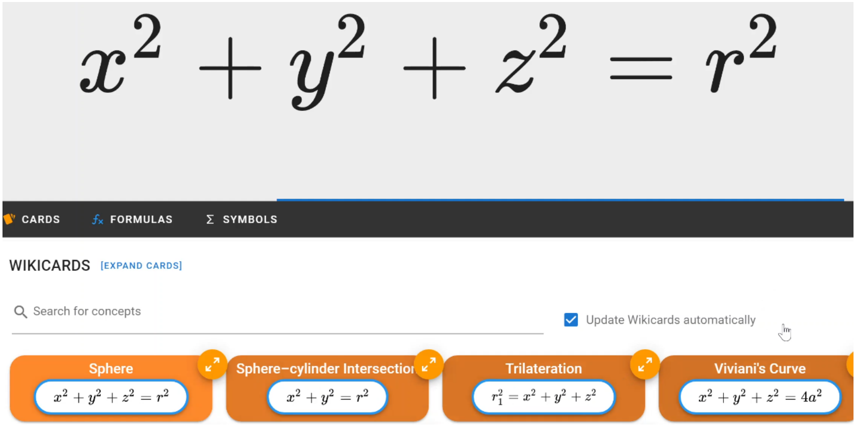

Figure 2. Annotating a formula with Wikidata-concepts in MathDeck.

2.2.2 Functionalities. MathDeck offers possibilities to draw handwritten formulae with a graphics editor or upload formulae in a picture, which then can be converted to LaTeX. It provides a symbol palette with a selection of well-known mathematical symbols to compose LaTeX-formatted formulae (see Figure 1). Also, with its *Wikicards* functionality, MathDeck can automatically link concepts from Wikipedia (via Wikidata) to well-known formulae (see Figure 2) to obtain a label and contextual information [6].

The MathDeck frontend is developed using *Vue.js*,[7] and makes use of a customized MathJax library for rendering math [6]. For obtaining the Wikicards suggestions, a modified version of Tangent-CFT [13] is used.

2.2.3 Review. MathDeck provides an example of a modern and advanced user interface for entering math formulae. Also linking, or even search for formulae as semantic concept might be a valuable addition for zbMATH users. MathDeck has been published recently [6] and contains proposals for further enhancements of the Wikicards. To our knowledge, the source code is not open-source. As of this writing, the Wikicards functionality did not suggest any cards for several well-known formulas in a test with multiple popular browsers.

2.3 Approach Zero

2.3.1 Description. Approach Zero[8](AZ) is a math search engine that can search for math expressions and keywords simultaneously. The referenced website also includes

[7] https://vuejs.org
[8] https://approach0.xyz

a demo-application. The search-engine, crawlers and more components are openly available on GitHub.[9]. The core of the search-engine is authorized under the MIT license. There have been no major changes to the core repository since the last release in 2016.

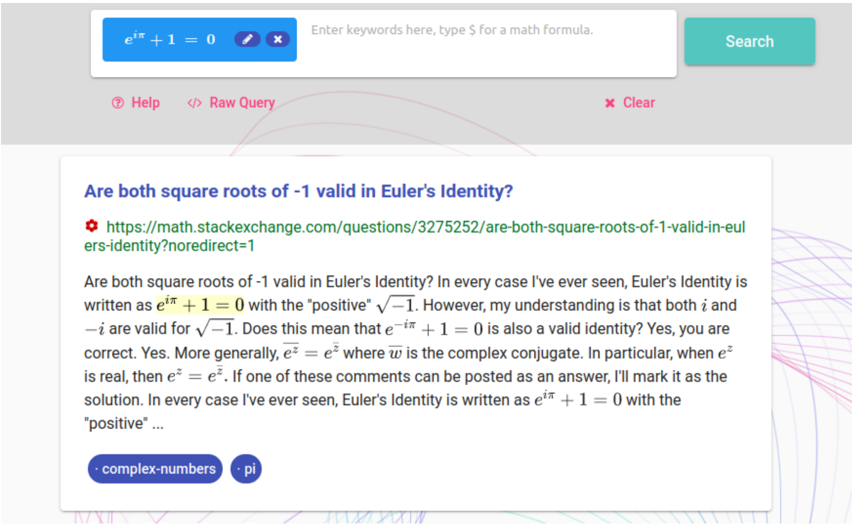

Figure 3. Found formula highlighting in the query results of Approach Zero.

2.3.2 Functionalities. AZ currently indexes the websites theartofproblemsolving.org and math.stackexchange.com. As a formula input format for indexing it exclusively reads LaTeX [24]. AZ uses a special search engine called OPMES (Operation-tree Pruning based Math Expression Search). This parses a math expression into an operator tree. It then extracts leaf root paths from the tree to represent structural information [23]. OPMES was evaluated retroactively with the NTCIR-12 MathIR Wikipedia Formula Browsing Task which is a benchmark for isolated formula retrieval [24]. For most configurations, it reaches higher scores than MCAT and an improved version of Tangent 3 [4], called Tangent-S, in terms of retrieval performance. It also achieved the best results among the systems compared in ARQMath-2 Task 2 for formula retrieval [14].

The user interface of the application has a symbol palette for entering well-known math symbols. The UI also parses input LaTeX while typing and directly renders the formula within the user input. In the list of search results for a query, the found formula in the text is directly highlighted (see Figure 3).

[9]https://github.com/approach0

Further notable features are cache on-disk index and the option to specify the memory usage limit.[10]

2.3.3 Review. The formula search engine can be considered for further investigation for use in zbMATH due to its advanced functions, free licensing and comprehensive documentation. The search results in the demo application provide a very illustrative example for the highlighting of found formulae.

2.4 SearchOnMath

2.4.1 Description. SearchOnMath[11] is a formula search engine equipped with a publicly available web application. It enables to search for combinations of keywords and formulae.

2.4.2 Functionalities. Similar to Approach Zero, the frontend of SearchOnMath has one single-line input field with combined keyword and formula queries. The input format for formulae is LaTeX and MathJax is used for rendering query results. Found formulae in the full text are also highlighted in the query results. The web application indexes a list of popular websites which contain math (e.g. MathOverflow[12]) as well as arXiv. Preview search results are rendered using MathJax. The indexed data can also be queried through an OAS3-specified RESTful API.

2.4.3 Review. SearchOnMath was a research project until 2015, then it became a start-up [17]. Since then, the project is developing in a more commercial direction. To our knowledge, there is no public code repository. This complicates reusability in zbMATH.

2.5 MIaS and EuDML

2.5.1 Description. The Math Indexer and Searcher (MIaS) is a math-aware full-text search engine based on Apache Lucene [20]. The engine differs from conventional search engines in a way that it allows fuzzy formulae search on joint text and math inverted index. The European Digital Mathematics Library[13] (EuDML) is an online library with more than 26,000 indexed items, where MIaS is used as a formula search engine.

---

[10] https://approach0.xyz/docs/content/en/features.html
[11] https://www.searchonmath.com/about
[12] https://mathoverflow.net
[13] https://eudml.org/search

2.5.2 Functionalities. The search form of EuDML enables one to define a set of keywords and a formula denoted in LaTeX or MathML notation. The search terms can then be combined with boolean operators. The frontend has an element displaying a rendered live preview of the given formula input. In the keyword-input fields, suggestions for common keywords are provided. Also, it shows comprehensive statistics about the overall search-results for a query.

MIaS is openly available on GitHub[14] and it is licensed with Apache 2.0. Also, under the same license, a publicly available web interface (WebMIaS)[15] exists. In all NTCIR-11 Math-2 tasks, it exceeded in its math retrieval performance [2]. The group from Masaryk University also participated in the main tasks NTCIR-12 MathIR. At the time of evaluation with NCTIR-12, MIaS was more in the mid-range of results [22].

2.5.3 Review. Similar to Approach Zero, the MIaS formula search engine can be considered for further investigation for use in zbMATH due to its free licensing and open availability. Also, we consider making the EuDML available through the zbMATH capabilities.

2.6 MCAT

2.6.1 Description. The MCAT group from the National Institute of Informatics in Tokyo participated in the NTCIR MathIR tasks 11 and 12 with an indexing scheme for mathematical expressions within an Apache Solr (Lucene) database. With this scheme, they enabled mathematical expressions searching using queries which contain both formulae and keywords [2, 22].

2.6.2 Functionalities. The method of MCAT reads Presentation as well as Content MathML and utilizes Apache Solr as a full-text search engine. Their search method obtains context window and description of formulae during the indexing process. It includes three levels of granularity for obtaining textual information (math, paragraph, and document levels). Also, it utilizes a dependency graph of mathematical expressions and a post-retrieval re-ranking method [11].

2.6.3 Review. MCAT has achieved excellent results in all tasks at NCTIR-12 [22]. The project is to our knowledge not publicly available, and this could make re-usage more difficult for zbMATH. Technological aspects from the publications describing the system could be extracted to build a custom system.

[14] https://github.com/MIR-MU/MIaS
[15] https://github.com/MIR-MU/WebMIaS

2.7 Tangent based search engines

2.7.1 Description. Tangent was introduced in 2015 as a method for indexing and retrieving mathematical expressions [21]. Since then, many methods have been introduced using Tangent as a baseline. The ARQMath labs, one and two, both contain a task for formula retrieval. These tasks have similarities in design to the NTCIR-12 Wikipedia Formula Browsing task, but differ in a way that relevance is defined contextually and evaluation is based on visually distinct formulae, rather than all formula instances [12].

2.7.2 Review. In the ARQMath formula retrieval tasks, modifications of Tangent were used by the participating teams from *Mathdowsers*[16] (Tangent-L) [16] and *DPRL* (Tangent-S) [12]. Both achieved comparable results.

2.8 Tangent-CFT

2.8.1 Description. Tangent Combined FastText (Tangent-CFT) is an embedding model for mathematical formulas which can be used for mathematical formula retrieval. It makes use of the SLT and OPT formula representations produced by the Tangent-S formula search engine. [13] Also, it utilizes the FastText n-gram embedding model [3]. The source code for Tangent-CFT and further model variations are publicly available[17].

2.8.2 Review. The *TanApp* introduced in [13] leverages linear combined retrieval scores from Tangent-CFT and Approach Zero. With this combination, the formula retrieval precision in the NTCIR-12 formula browsing task can be significantly boosted in comparison to the original Approach Zero. Also, in the same task, Tangent-CFT in its standalone application outperforms MCAT [13]. Tangent-CFT could be utilised as an additional method with MWS in zbMATH, to enhance the retrieval precision.

3 Extension proposal

By examining the previously mentioned math engines and applications, several potential extensions for zbMATH were extracted. This section presents the proposals for the new functionalities.

3.1 Search function for systems of equations

In some cases (e.g. in systems of linear equations) a mathematical expression can consist of a collection of multiple equations involving the same set of variables. Traditional math search currently allows querying for a single search term at a time, and

[16] https://github.com/kiking0501/MathDowsers-ARQMath
[17] https://github.com/BehroozMansouri/TangentCFT

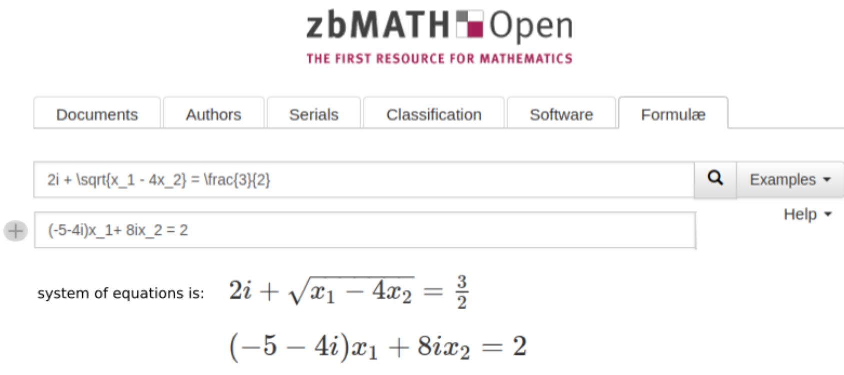

Figure 4. Design of a user input for systems of equations in zbMATH formula search.

in some cases for a Boolean combination of multiple terms. In our formula search, we propose to add a search input element that allows using a consistent set of variables across multiple search terms. Figure 4 shows what such user input for a system of equations could look like in zbMATH in the future.

3.2 Syntax verification

Users may not use the correct LaTeX syntax when entering the formula in the formula search input field. Therefore, we propose an automated syntax check that highlights the errors in the input field or its surroundings.

3.3 Math expression simplification

Mathematical expressions can take different forms while articulating the same functionality. Math search users may not have the simplest form of the formula they want to search for at their fingertips. This is a suggestion for introducing a term simplification within the formula search, which can be optionally activated. This term simplification will also provide the zbMATH user with the simplified input term as an additional output value.

3.4 Improving accessibility

This section proposes several minor features that we propose for improving the accessibility of zbMATH. One of them is the introduction of a symbol palette for the zbMATH math search, containing well-known and frequently used mathematical symbols. When the user selects the visual symbol, its LaTeX representation is added to the search bar. We also consider introducing a user input component for math search that allows drawing math symbols in an image editor embedded in the zbMATH web interface. Examples of both proposed changes can be seen in MathDeck.[18] Another proposed feature is to provide an auto-completion list for the current user input of LaTeX expressions in the zbMATH search bar.

3.5 Semantic concept annotation

Hereby, we propose the annotation of the given formula in the search box of zbMATH to semantic concepts in a knowledge graph (KG). With the human-readable labels to a formula extracted from the KG, keyword suggestions can be realised. Also, additional information about research data obtained from the KG can be linked to the search results. The MathDeck *Wikicards* (see Figure 2) can be viewed as an existing example of concept annotation. We consider annotating semantic concepts and obtaining additional information from the KG of the MaRDI Portal.[19] This KG is currently being set up by FIZ Karlsruhe together with the Zuse Institut Berlin. It will connect vast amounts of mathematical research data and formulae.

3.6 Rendered search results

We propose a highlighting for the found formulae in the query results similar to Approach Zero in zbMATH (see Figure 3). The found formula and the surrounding text from the summaries of the indexed publications will be displayed directly in the list of results. Also, we consider rendering the found formula in this preview.

3.7 Improving retrieval accuracy

Currently, the ranking of documents in the search results of a formula search in zbMATH is realised using similarity scores calculated by MWS. MWS uses traditional tree-based search engine indices. Following the example of [13], we propose improving the formula retrieval accuracy by computing a linear combination of the traditional MWS-based score and a newly introduced score based on formula embeddings such as Tangent-CFT.

3.8 Finding mathematical symbols by facetted search

When typing a formula in the input of math search, the names of mathematical symbols can be unknown or may differ between cultural and lingual contexts. While the

[18] https://www.mathdeck.org
[19] https://portal.mardi4nfdi.de

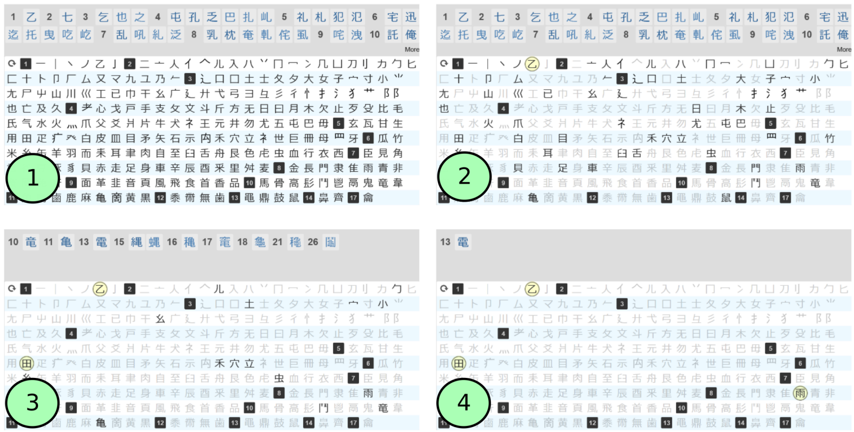

Figure 5. The four stages of Jisho's facetted search [1] to find the 13 stroke Kanji 電 ('*den*'). Selecting radicals in each step (2-4) narrows down the possible search results.

US or Germany uses $\geq$ to express a greater or equal relation, in Japan, the notation $\geqq$ is more common. Considering the sheer amount of different math notations, it might not be obvious to a student from Japan that $\geq$ and $\geqq$ refer to the same relation.

Consider an example case, where a Japanese student, reading a German math publication, encounters the visual appearance of a formula. Lacking the cultural context, the student does not know the meaning or the denomination of the formula since the explicit meaning of symbols is often not mentioned, even in educational literature.

To find an explanation for the formula and its meaning, the researcher could consider using a math search engine to find literature explaining the formula. In a conventional math search engine, the input notation will be difficult, since a written notation (e.g. LaTeX symbols) is not known to the researcher.

One solution to aid the search for mathematical symbols based on their visual appearance is using the aforementioned symbol palette. This can be rather confusing since there is a vast variety of mathematical symbols which will overload the symbol palette.

The problem of searching an unknown math symbol without knowing its name and meaning is significantly related to finding unknown words or morphemes in a natural language that uses logograms, such as *hanzi* in Chinese or *Kanji* in Japanese. In the Japanese writing system, a character (Kanji) can often be composed of one or more combinations of the 214 radicals.[20] The radicals represent smaller basic symbols of the Kanji and also can represent Kanji themselves. Japanese Kanji are often classified by the order of strokes and the radicals they consist of, which allows a

[20] 214 is the number of traditional radicals, but the exact number may vary

reader to find a Kanji fairly easy in a dictionary. The *Jisho* [1] digital library provides a facetted search system to find Japanese characters by their parts (see Figure 5).

To simplify the search for mathematical formula where neither the name nor the meaning is known to the user, we propose a new approach for defining the search input. The user input will be defined by adopting the Logogram classification of Chinese and Japanese characters. This can be realized by deriving a limited set of basic visual components of mathematical symbols. Like the mentioned radicals, these can be used as base symbols to implement a faceted search system similar to *Jisho*, which can find mathematical expressions instead of Kanji characters.

Engine	Input Formats	Availability	Technology	Evaluated	Applications
MathWebSearch [10]	CMML, LaTeX (with converter)	Public repo, GPLv3	Substitution tree indexing	NTCIR-11 and 12 ARQMATH-1 and 2	zbMATH Open, Mediawiki Extension
Approach Zero [24]	LaTeX	Public repo, MIT-license	Operation-tree pruning based math expression search	Retroactively NTCIR 12, ARQMATH-1 and 2	Approach Zero
SearchOnMath [17]	LaTeX	Public API, no code repo	Lexical analysis, degree of similarity	SearchOnMath has been used to evaluate hardware performance	SearchOnMath
Tangent CFT [13]	LaTeX	Public repo license N/A	Embedding model for mathematical formulae	Retroactively NTCIR-12 ARQMATH-1 and 2	—
MIaS [20]	CMML, PMML, LaTeX	Public repo Apache 2.0	Lucene index for text, variables and constants unification for formula ordering	NTCIR-11 and 12	EUDML

Notes: CMML stands for Content Math ML, PMML stands for Presentation Math ML

Table 1. Overview of the core math search engines and applications investigated and their main attributes

4 Conclusion and outlook

For our investigation, we examined various math search engines and formula search related applications. A summary of the main engines and their applications and attributes can be found in Table 1. We found several suggestions for improvements to zbMATH. We have collected suggestions for extensions to the user interface that allow for increased accessibility, as well as for linking search terms and search results with additional information from knowledge graphs. Furthermore, from the investigation of several search engines, we proposed that retrieval performance can be improved by linear combination of traditional tree-based ranking scores with a scoring based on formula-embeddings.

The third ARQMath lab[21] announced for the middle of 2022 can be a source for future evaluations of formula search engines. Synergy effects in the implementation of extensions to zbMATH regarding the knowledge graph and data linking may result from the MaRDI portal and the associated KG, which have been under construction since the beginning of 2022.

References

[1] K. Ahlström, M. Ahlström, and A. Plummer, Jisho.org: Japanese dictionary. https://Jisho.org visited on 17 March 2024

[2] A. Aizawa, M. Kohlhase, I. Ounis, and M. Schubotz, NTCIR-11mmath-2 task overview. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, December 9–12, 2014*, edited by N. Kando, H. Joho, and K. Kishida, pp. 88–98, National Institute of Informatics (NII), 2014

[3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5** (2017), 135–146

[4] K. Davila, R. Zanibbi, A. Kane, and F. W. Tompa, Tangent-3 at the NTCIR-12 MathIR task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, June 7–10, 2016*, edited by N. Kando, T. Sakai and M. Sanderson, pp. 338–345, National Institute of Informatics (NII), 2016

[5] MSC2020 Mathematics Subject Classification System. 2021 https://zbmath.org/static/msc2020.pdf visited on 17 March 2024

[6] Y. Diaz, G. Nishizawa, B. Mansouri, K. Davila, and R. Zanibbi, The MathDeck formula editor: Interactive formula entry combining LaTeX, structure editing, and search. In *CHI EA '21: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing System*, edited by Y. Kitamura, A. Quigley, K. Isbister, and T. Igarashi, pp. 1–5, Association for Computing Machinery, New York, NY, 2021

[7] P. Graf, Substitution tree indexing. In International Conference on Rewriting Techniques and Applications RTA 1995: Rewriting Techniques and Applications, edited by J. Hsiang, pp. 117–131, Lecture Notes in Computer Science 914, Springer, Berlin, 1995

[21] https://www.cs.rit.edu/~dprl/ARQMath

[8] F. Guidi and C. Sacerdoti Coen, A survey on retrieval of mathematical knowledge. In *International Conference on Intelligent Computer Mathematics CICM 2015: Intelligent Computer Mathematics*, edited by M. Kerber, J. Carette, C. Kaliszyk, F. Rabe, and V. Sorge, pp. 296–315, Lecture Notes in Computer Science 9150, Springer, Cham, 2015

[9] R. Hambasan and M. Kohlhase, Faceted search for mathematics. In *International Conference on Mathematical Aspects of Computer and Information Sciences MACIS 2015: Mathematical Aspects of Computer and Information Sciences*, edited by R. Hambasan and M. Kohlhase, pp. 406–420, Springer, Cham, 2016

[10] R. Hambasan, M. Kohlhase, and C.-C. Prodescu, nMathWebSearch at NTCIR-11. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, December 9–12, 2014*, edited by N. Kando, H. Joho, and K. Kishida, pp. 114–119, National Institute of Informatics (NII), 2014

[11] G. Y. Kristianto, G. Topić, and A. Aizawa, MCAT Math Retrieval System for NTCIR-12 MathIR Task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, Tokyo, June 7–10, 2016*, edited by N. Kando, T. Sakai and M. Sanderson, pp. 323–330, National Institute of Informatics (NII), 2016

[12] B. Mansouri, D. Oard, and R. Zanibbi, DPRL systems in the CLEF 2020 ARQMath Lab. 2020. In *Working Notes of CLEF 2020 – Conference and Labs of the Evaluation Forum, Thessaloniki, September 22–25, 2020, edited by L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol, article no. 223, CEUR Workshop Proceedings 2696, CEUR-WS.org, 2020*

[13] B. Mansouri, S. Rohatgi, D. W. Oard, J. Wu, C. L. Giles, and R. Zanibbi, Tangent-CFT: An embedding model for mathematical formulas. In *ICTIR '19: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, Santa Clara, CA, October 2–5, 2019, edited by Y. Fang, Y. Zhang, J. Allan, K. Balog, B. Carterette, and J Guo,* pp. 11–18, Association for Computing Machinery, New York, NY, 2019

[14] B. Mansouri, R. Zanibbi, D. W. Oard, and A. Agarwal, Overview of ARQMath-2 (2021): Second CLEF lab on answer retrieval for questions on math. In *Experimental IR meets multilinguality, multimodality, and interaction. 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings*, edited by K. S. Candan, B. Ionescu, L. Goeuriot, B. Larsen, H. Müller, A. Joly, M. Maistro, F. Piroi, G. Faggioli, and N. Ferro, pp. 215–238, Springer, Cham, 2021

[15] F. Müller and O. Teschke, Full text formula search in zbMATH. *Eur. Math. Soc. Newsl.* **102** (2016), 51–51

[16] Y. K. Ng, D. J. Fraser, B. Kassaie, G. Labahn, M. S. Marzouk, and F. Wm. Tompa, and K. Wang, Dowsing for math answers with tangent-L. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, September 22–25, 2020*, edited by L. Cappellato, C. Eickhoff, N. Ferro, and A. Névéol, article no. 16, CEUR Workshop Proceedings 2696, CEUR-WS.org, 2020

[17] R. M. Oliveira, F. B. Gonzaga, V. C. Barbosa, and G. B. Xexéo, A distributed system for SearchOnMath based on the Microsoft BizsPark program. In *2018: Proceedings of the 33rd Brazilian Symposium on Databases, edited by B. F. Lóscio, pp. 289–294, Sociedade Brasileira de Computação, 2018*

[18] M. Petrera, D. Trautwein, I. Beckenbach, D. Ehsani, F. Müller, O. Teschke, B. Gipp, and M. Schubotz, zbMATH Open: API solutions and research challenges. In *DISCO 2021: Digital Infrastructures for Scholarly Content Objects 2021*, edited by W.-T. Balke, A. de Waard, Y. Fu, B. Hua, J. Schneider, N. Song, X. Wang, pp. 4–13, CEUR Workshop Proceedings 2976, CEUR-WS.org, 2021

[19] M. Schubotz, *Augmenting mathematical formulae for more effective querying & efficient presentation*. Ph.D. thesis, Technical University of Berlin, 2017, Epubli, Berlin, 2017

[20] P. Sojka and M. Líška, The art of mathematics retrieval. In *DocEng '11: Proceedings of the 11th ACM symposium on document engineering*, edited by M. Hardy, pp. 57–60, Association of Computing Machinery, Mountain View, CA, 2011

[21] D. Stalnaker and R. Zanibbi, Math expression retrieval using an inverted index over symbol pairs. In *Document recognition and retrieval XXII*, edited by E. K. Ringger and B. Lamiroy, pp. 34–45, Proceedings Vol. 9402, International Society for Optics and Photonics, SPIE, 2015

[22] R. Zanibbi, A. Aizawa, M. Kohlhase, I. Ounis, G. Topić, and K. Davila, NTCIR-12 MathIR task overview. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, Tokyo, June 7–10, 2016*, edited by N. Kando, T. Sakai and M. Sanderson, pp. 299–308, National Institute of Informatics (NII), 2016

[23] W. Zhong and H. Fang, OPMES: A similarity search engine for mathematical content. In *European Conference on Information Retrieval ECIR 2016: Advances in Information Retrieval*, edited by N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Nunzio, C. Hauff, and G. Silvello, pp. 849–852, Lecture Notes in Computer Science 9626, Springer, Cham, 2016

[24] W. Zhong and R. Zanibbi, Structural similarity search for formulas using leaf-root paths in operator subtrees. In *European Conference on Information Retrieval ECIR 2019: Advances in Information Retrieval*, edited by L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, pp. 116–129, Lecture Notes in Computer Science 11437, Springer, Cham, 2019