Searching the ACL Anthology with Math Formulas and Text

Bryan Amador*
Matt Langsenkamp*
{ma5339,ml2513}@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Abhisek Dey
Ayush Kumar Shah
{ad4529,as1211}@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Richard Zanibbi
rxzvcs@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

ABSTRACT

Mathematical notation is a key analytical resource for science and technology. Unfortunately, current math-aware search engines require LaTeX or template palettes to construct formulas, which can be challenging for non-experts. Also, their indexed collections are primarily web pages where formulas are represented explicitly in machine-readable formats (e.g., LaTeX, Presentation MathML). The new MathDeck system searches PDF documents in a portion of the ACL Anthology using both formulas and text, and shows matched words and formulas along with other extracted formulas in-context. In PDF, formulas are not demarcated: a new indexing module extracts formulas using PDF vector graphics information and computer vision techniques. For non-expert users and visual editing, a central design feature of MathDeck's interface is formula 'chips' usable in formula creation, search, reuse, and annotation with titles and descriptions in cards. For experts, LaTeX is supported in the text query box and the visual formula editor. MathDeck is open-source, and our demo is available online.

CCS CONCEPTS

• **Information systems** → **Search interfaces**; **Search engine indexing**; *Content analysis and feature selection*; **Mathematics retrieval**; **Structured text search**.

KEYWORDS

Mathematical Information Retrieval (MIR), math-aware search, LaTeX, multimodal retrieval, PDF

ACM Reference Format:
Bryan Amador, Matt Langsenkamp, Abhisek Dey, Ayush Kumar Shah, and Richard Zanibbi. 2023. Searching the ACL Anthology with Math Formulas and Text. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3539618.3591803

*The first two authors contributed equally to this research.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591803

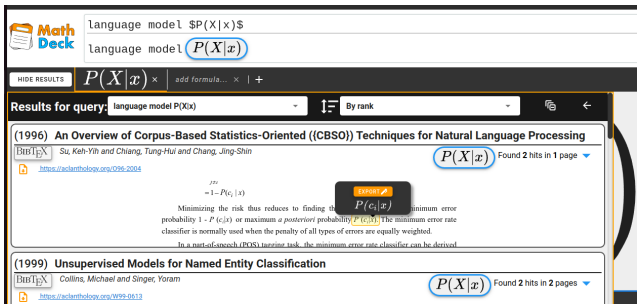

Figure 1: A user searches for a Natural Language Processing (NLP) model using conditional probability. They do not remember the model name, just the general form of the formula. This information need is represented by the query *"language model* $P(X|x)$*".*

1 INTRODUCTION

Math-aware search has been of growing interest for the academic community due to its application in many fields of science and technology [8, 9, 13, 15, 23]. Standard text-based search engines such as Google, Bing, or DuckDuckGo provide limited support for math formula search: formulas are treated as plain text. This limits a user's ability to locate a formula in a collection, or to find similar or related formulas and associated information. A number of math-aware search engines that support formulas in queries such as Approach Zero [24] and Zentralblatt Math[1] require using LaTeX to enter formulas. This presents a challenge for non-experts, who may be both unfamiliar with LaTeX and identify a formula primarily by appearance rather than mathematical content [22].

Available math-aware search engines generally index content in web sites where formulas are represented explicitly using LaTeX or MathML markup, making extraction reliable (e.g., Wikipedia or Math Stack Exchange). However, many PDF documents also contain formulas – particularly in the educational and technical literatures (e.g., math textbooks and research papers). Unfortunately, formulas are not demarcated in PDF [20]. Search engines generally convert PDFs to plain text for indexing, making formula extraction and retrieval unreliable.

In this demonstration we present an updated version of the MathDeck math-aware search interface [7, 17] shown in Figure 1, along with new modules for formula search in PDF documents. MathDeck is designed for easy-to-use retrieval with text and formulas, and to accommodate both math experts and non-experts. A focus of

[1] Zentralblatt uses MathWebSearch [10] as its search engine

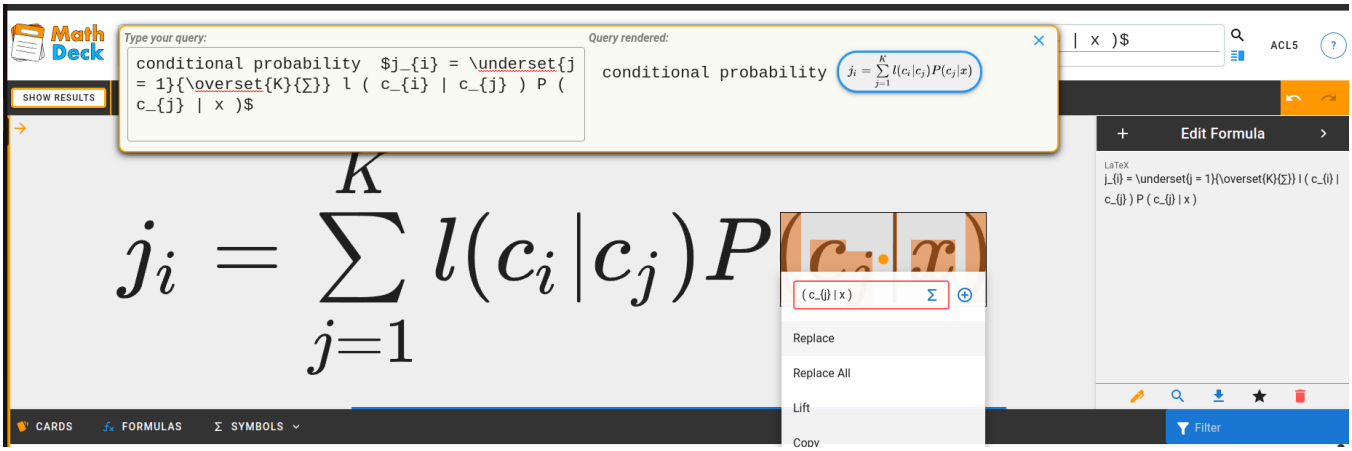

Figure 2: After browsing results for the query in Figure 1, the user sees a more concrete formula that they export to the canvas for editing/refinement. Formulas may also be entered directly or edited using handwriting and dropping formula chips.

MathDeck's design is closely integrating formula editing with creating search queries and re-using formulas found in search results. MathDeck is open-source, and our demo is available online.[2]

To better integrate formula reuse and editing within query formulation, we present a new query bar where formulas are entered using both raw LaTeX and formula *chips* (see Figure 2). Chips are made using a combination of visual editing operations, handwriting, other formula chips, and LaTeX. Chips can be dragged, copied, saved as images, and annotated in 'concept cards' with a title and description. Also, our new search results pages highlight matching formulas and terms in PDF pages (see Figure 1), and allow users to browse and export formulas for use in queries and users' documents (e.g., using the LaTeX embedded in chips).

Math formula extraction from PDF documents is performed using a combination of PDF vector graphics drawing instructions and computer vision techniques [6, 20]. For our first test collection, we have crawled the publicly available ACL Anthology of Natural Language Processing (NLP) research papers, which provides a complete BibTeX database for archived papers along with PDFs.[3]

We believe that MathDeck and similar systems will help expert and non-expert users find and share mathematical information in PDFs and other sources more easily, through (1) UI designs that tightly integrate formula editing, search, reuse, and annotation, and (2) formula extraction modules for PDFs and other formats where formulas are still generally inaccessible (e.g., videos [5]). We aim to make expressing navigational queries with formulas and looking up unfamiliar notation easier, and to better support more complex sensemaking activities, such as for coursework and literature reviews. Further, we believe these benefits will be seen in practice within a couple of years' time.

2 RELATED WORK

Formula input. Some interfaces such as Zentralblatt Math provide a text input field that supports LaTeX, rendering the query below the input. Wolfram Alpha and Symbolab provide a query input where templates for math expressions (exponents, logarithms, fractions, etc) can be added from pallettes to construct the query. Other systems such as Approach Zero [24] permit LaTeX entry for formulas surrounded by dollar signs ($), which are rendered as they are entered. SearchOnMath [18] is similar, but demarcates LaTeX formulas using `\(` and `\)`, and provides a formula template palette and renders the query above the search box. The Digital Library of Mathematical Functions (DLMF) [16] has a query language similar to LaTeX, with support for binary operators (e.g., `(x OR t)^3`).

MathDeck's new query bar supports LaTeX, with a rendered view below the input and LaTeX indicated by dollar signs. To support formula re-use and visual formula editing, we also include a functionality similar to the predefined templates in Wolfram and Symbolab, which is the support for copying or dropping formula *chips* into the query bar. Chips are created using operations on a canvas (see Figure 2). Unlike static palette-style menus, chips may be dynamically created, combined, and edited. Chips are automatically stored and made available in the interface as they are created, with options to favorite chips and to annotate them in formula concept cards [7].

Formula hits. Systems such as ApproachZero and SearchOnMath show matched formulas by highlighting their bounding box in rendered HTML. We extend this functionality by both highlighting matches, and also making formulas exportable to the editing canvas along with an associated chip for later reuse.

Formula extraction in PDF. Most math-aware search engines index web pages that represent math in explicit markup languages such as LaTeX or MathML, making extraction straight-forward [23]. In PDF graphic regions are not identified, and so detection techniques must be applied. An early system for extracting graphics from PDF documents was proposed by Chao [3], traversing PDF files to identify relationships between content elements, and then classifying elements as graphics or text. Extraction of figures has been an area of focus here, for example in Physics-related documents [19] using PDF commands that produce figures. Academic publishers including Springer and Elsevier allow users to access figures present in documents, and the Semantic Scholar system presents extracted figures in online paper summaries [4].

[2] https://people.rit.edu/ma5339/mathdeck_landing/
[3] https://aclanthology.org

Sorge et al. did early work in extracting math formulas from PDF [2, 21]; in our work, symbols in PDFs are extracted using the SymbolScraper tool [20] which removes the need for OCR to identify symbols when formulas are represented by PDF vector graphics drawing instructions. However, the lack of formula demarcation in PDFs led us to use a visual detector for formulas in rendered page images, a modified Single Shot Detector [6] (a neural network), and then recognize formula structure from PDF symbols where available, and otherwise directly from formula images [3].

MathDeck. The previous version of MathDeck [7] introduced a novel multi-modal visual formula editor. A canvas allows formulas to be created using chips, handwriting, LaTeX, and structure editing operations for symbols and selected subexpressions. Editing operations include deletion, copying, replacement, and insertion of chips or LaTeX at locations around individual symbols (e.g., as a superscript; see Fig 2). Chips with common symbols and structures are provided in a palette, along with a collection of formulas from Wikipedia stored in formula concept cards [17]. 'Wiki cards' also provide a simple form of autocompletion, showing similar formulas as formulas are entered. We have refactored the query bar, provided a new search results view, and added the ability to search PDF collections for this demonstration. Previously, MathDeck only generated query strings passed to external search engines (e.g., Approach Zero, DLMF, Google, etc.).

3 UPDATED SEARCH INTERFACE

Query bar: A primary goal of MathDeck is providing an easy-to-use interface for searching mathematical information. Hence, the way users expresses their information needs (i.e., input a query) is crucial. A user-friendly query bar is needed for users to communicate queries comfortably, regardless of their expertise.

Our refactored query bar provides two views: a raw text input and a rendered view. The text input allows users to construct queries by entering text and math expressions in LaTeX surrounded by dollar signs ($). Below this, a second view shows the query with formulas in chips that may be edited by dragging a chip to the canvas, or using the pencil icon in the context menu accessed by clicking on a chip (see Figure 2). Changes made to query chips on the canvas are synchronized with the original chip. Chips can also be dropped anywhere in the text input box to insert a formula. Chip operations from the previous version are still available (delete, download, add to favorites, editing, etc). Since queries may not fit on the screen in a single-line view, an expandable query bar is available to the user (see Figure 2): raw-text input is shown at left, and its rendering at right, similar to commercial LaTeX editors such as Overleaf.[4]

Result visualization and navigation: Previous searching capabilities of MathDeck were limited to external search engines. We developed a new slide-out window for examining and interacting with PDF search results. To help users recognize and browse through relevant documents, results should be easy to scan and navigate. Each matched document is initially shown as a small page window with formulas and/or text shown for the highest scoring page of the PDF; in this way, users can see hits in-context while scrolling through search results. Hits can be expanded to show complete PDF pages, also revealing navigation buttons used to reach other pages matched in the document. To organize search results, we designed an expandable search panel capable of showing results for multiples queries, providing a simple search history mechanism (see Figure 1). Search results may be sorted by decreasing relevance score for a document page, and ascending/descending by publication year.

Matches are visualized using highlighting of bounding boxes for both text (in blue) and formulas (in yellow). Highlighted formulas may be converted to a chip, and sent to the editing canvas. This allows users to construct new queries from formulas in search results. A button to show *all* extracted formulas on a page is also available. When clicked, unmatched formulas are highlighted in gray, and can also be extracted to a chip. Chips may also be downloaded as an image that can be reused as a chip, and added to a favorites list.

4 MULTIMODAL SEARCH ENGINE

Our search engine indexes mathematical expressions and plain text from PDF documents. For this demonstration, we have indexed 90,810 pages from 11,021 conference papers and abstracts in the ACL Anthology for the years 1952 through 2005. The collection contains 463,050 detected formulas. Part of our open source extraction and indexing system is a module that downloads the current version of the ACL Anthology BibTeX database, along with associated PDF files identified in the BibTeX file.

We use an updated formula extraction pipeline described in [20], along with text and formula indexing modules to index the collection. Figure 3 illustrates the extraction and indexing pipeline used for math formulas and text. The search engine is implemented in OpenSearch, using separate indexes for text and formulas.

Formula extraction: We extract locations for both formula and plain text from PDF documents. Three main tools are used in the extraction pipeline: (1) SymbolScraper extracts locations for both text and graphics from PDF files by intercepting the PDF rendering pipeline [20]; (2) ScanSSD-XYc locates bounding boxes of the formulas using visual features, applying a convolutional neural network in windows from the document image [6]; and (3) QD-GGA takes locations of formulas and PDF symbols where available to parse the hierarchical structure of the formula using a convolutional neural network [12]. The system produces Presentation MathML, which is converted to LaTeX. Using QD-GGA allows us to recognize formulas both from provided PDF symbols when they are available and reliable, and purely image-based recognition otherwise (e.g., for older documents with noisy or missing OCR information). The end product of this process is a Page-Region-Object Table (PROTable), a TSV file identifying regions in a document along with related information: in this case, the formula and character/symbol locations (bounding boxes) in their respective documents.

Because ScanSSD-XYc and QD-GGA use PNG page images, they must be rendered for every page in every indexed document. QD-GGA also uses an inefficient representation for the connected components (CCs) sent to the GPU for parsing, preventing multiple large-sized formulas from fitting into a single batch. Currently, the speed of the parser is the main bottleneck for extraction. Indexing takes between 24 and 36 hours across 3 machines, two of which have 4 Nvidia 2080 GPUs, one of which has 2 Nvidia 1080 GPUs. To

[4] https://www.overleaf.com/

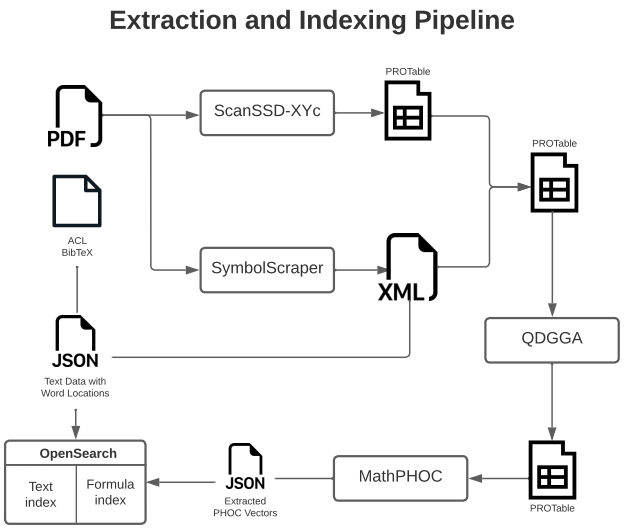

Figure 3: Indexing formulas and text. Page-Region-Object Tables (i.e., PROTable TSV files) record formula locations (from ScanSSD-XYc), symbol locations (SymbolScraper), and Presentation MathML for extracted formulas (QD-GGA). SymbolScraper also provides word locations, and paper metadata (e.g., titles) is taken from ACL-provided BibTeX data.

date, our detector and parser have been trained using isolated formulas without matrix or tabular structures, and on relatively small collections and formula sizes, resulting in some missed and split formulas, and skipping very large formulas during indexing. We plan to address these limitations in future versions of the system.

Indexing: Both formulas and text are indexed using OpenSearch.[5] Once the final PROTable files have been created by QD-GGA, formulas are indexed using an improved version of MathPHOC [11]. 'PHOC' stands for Pyramidal Histogram of Characters, which was originally devised for word spotting in handwritten text [1]. MathPHOC uses binary vectors to represent the location of symbols in formulas using a pyramid of equal-sized partitions (e.g., whole formula, left/right, left/center/right, etc.). We used an **R10** configuration, which captures symbol locations using between 1 and 10 equally-spaced concentric rectangles, producing 55 overlapping regions ($\sum_{i=1}^{10} i = 10(11)/2 = 55$). A 55-bit vector is then produced for each symbol in a formula, which are indexed as 64-bit integers.

We build our text index by taking the symbol data obtained through SymbolScraper, and indexing each page as its own document with fields for body text, title, author(s), year, publisher, ACL anthology document URL, file name, page number, and the document identifier. Locations for text words are stored in a nested map, where document identifiers map to pages, which then map to the words on a page, which finally maps to a list of bounding box locations for the word on a specific document page. These mapping allow us to render 2D bounding boxes over matched words on PDF pages. These mappings are stored as JSON files, and loaded as required at query time.

Retrieval: We perform retrieval using two inverted indices: one for text and one for formulas. A query is split up into text tokens and a list of formulas represented by LaTeX and chips in MathDeck. For example, consider this query:

`If I have a tf term of` $\boxed{w_{d,t} = 1 + ln(f_{d,t})}$

`how do I apply my idf term of` $\boxed{w_{q,t} = ln\left(\frac{N}{f_t} + 1\right)}$?

becomes *"If I have a tf term of how do I apply my idf term of?"* and two formula terms, represented in LaTeX as `$w_{d,t}=$` and `$w_{q,t}=ln\left(\frac{N}{f_{t}}+1\right)$`. For the text we use a built-in OpenSearch BM25 function to score pages. The two formula terms are rendered as SVG from their LaTeX using MathJax,[6] and PHOCs are generated from symbol locations in the SVG files. PHOC postings for query symbols are retrieved from the formula inverted index, and then retrieved formulas are scored by cosine similarity over their associated PHOC vectors [11].

Individual formula search results are then grouped by document and page. The top formula score for each query term on each page is summed and multiplied by a constant α, set to 3, and added to the the text score for that page. Finally, the document score is set to the highest scoring page in that document.

For visualizing search results, when sending search results back to MathDeck, we attach the associated location data from the SymbolScraper maps for text, and from PRO tables for formulas.

5 CONCLUSION

Our demonstration allows users to try an updated MathDeck search engine that can search PDFs in the ACL Anthology using formulas and text. For this demonstration we have created a tool to download ACL anthology data, an improved math formula extraction pipeline, and a new indexing pipeline for text and formulas in PDF documents. We have also updated the MathDeck search interface to be more flexible and user-friendly: the search box now supports both text terms and formulas entered using a combination of raw LaTeX and structure editing operations organized around formula *chips*. Search results show the precise location of words and formulas matching the query on PDF pages, and formulas seen in search results may be saved directly to chips.

In the future, we plan to index the complete ACL Anthology and IR Anthology[7] collections, improve our formula extraction pipeline, strengthen the multimodal retrieval model (e.g., through the use of embeddings [14]), improve the MathDeck interface, and look into adapting MathDeck for other domains, such as chemistry.

ACKNOWLEDGMENTS

We thank Doug Oard for suggesting that we use ACL as our test collection, and Shaurya Rohatgi for suggesting using OpenSearch. This work was supported by the Alfred P. Sloan Foundation under Grant No. G-2017-9827 and the National Science Foundation (USA) under Grant No. IIS-1717997.

[5] https://opensearch.org/

[6] https://www.mathjax.org

[7] https://ir.webis.de/anthology

REFERENCES

[1] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. 2014. Word Spotting and Recognition with Embedded Attributes. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 12 (2014), 2552–2566. https://doi.org/10.1109/TPAMI.2014.2339814

[2] Josef B. Baker, Alan P. Sexton, and Volker Sorge. 2010. Faithful mathematical formula recognition from PDF documents. In *Document Analysis Systems (ACM International Conference Proceeding Series)*. ACM, 485–492.

[3] Hui Chao. 2003. Graphics extraction in a PDF document. In *Document Recognition and Retrieval X*, Tapas Kanungo, Elisa H. Barney Smith, Jianying Hu, and Paul B. Kantor (Eds.), Vol. 5010. International Society for Optics and Photonics, SPIE, 317 – 325. https://doi.org/10.1117/12.479683

[4] Christopher Clark and Santosh Divvala. 2016. PDFFigures 2.0: Mining Figures from Research Papers. (2016).

[5] Kenny Davila and Richard Zanibbi. 2018. Visual Search Engine for Handwritten and Typeset Math in Lecture Videos and LATEX Notes. In *ICFHR*. IEEE Computer Society, 50–55.

[6] Abhisek Dey and Richard Zanibbi. 2021. ScanSSD-XYc: faster detection for math formulas. In *Document Analysis and Recognition–ICDAR 2021 Workshops: Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16*. Springer, 91–96.

[7] Yancarlos Diaz, Gavin Nishizawa, Behrooz Mansouri, Kenny Davila, and Richard Zanibbi. 2021. The MathDeck Formula Editor: Interactive Formula Entry Combining LaTeX , Structure Editing, and Search. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 192, 5 pages. https://doi.org/10.1145/3411763.3451564

[8] Deborah Ferreira, Marco Valentino, Andre Freitas, Sean Welleck, and Moritz Schubotz (Eds.). 2022. *Proceedings of the 1st Workshop on Mathematical Natural Language Processing (MathNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid). https://aclanthology.org/2022.mathnlp-1.0

[9] Ferruccio Guidi and Claudio Sacerdoti Coen. 2016. A Survey on Retrieval of Mathematical Knowledge. *Math. Comput. Sci.* 10, 4 (2016), 409–427.

[10] Michael Kohlhase, Bogdan A Matican, and Corneliu-Claudiu Prodescu. 2012. Mathwebsearch 0.5: Scaling an open formula search engine. In *Intelligent Computer Mathematics: 11th International Conference, AISC 2012, 19th Symposium, Calculemus 2012, 5th International Workshop, DML 2012, 11th International Conference, MKM 2012, Systems and Projects, Held as Part of CICM 2012, Bremen, Germany, July 8-13, 2012. Proceedings 5*. Springer, 342–357.

[11] Matt Langsenkamp, Behrooz Mansouri, and Richard Zanibbi. 2022. Expanding Spatial Regions and Incorporating IDF for PHOC-Based Math Formula Retrieval at ARQMath-3. *Proc. CLEF 2022 (CEUR Working Notes)* (2022).

[12] Mahshad Mahdavi and Richard Zanibbi. 2020. Visual parsing with query-driven global graph attention (QD-GGA): preliminary results for handwritten math formula recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 570–571.

[13] Behrooz Mansouri, Vít Novotný, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. 2022. Overview of ARQMath-3 (2022): Third CLEF Lab on Answer Retrieval for Questions on Math (Working Notes Version). In *CLEF (Working Notes) (CEUR Workshop Proceedings, Vol. 3180)*. CEUR-WS.org, 1–27.

[14] Behrooz Mansouri, Douglas W. Oard, and Richard Zanibbi. 2022. Contextualized Formula Search Using Math Abstract Meaning Representation. In *CIKM*. ACM, 4329–4333.

[15] Jordan Meadows and André Freitas. 2022. A Survey in Mathematical Language Processing. *CoRR* abs/2205.15231 (2022).

[16] Bruce R Miller and Abdou Youssef. 2003. Technical aspects of the digital library of mathematical functions. *Annals of Mathematics and Artificial Intelligence* 38, 1 (2003), 121–136.

[17] Gavin Nishizawa, Jennifer Liu, Yancarlos Diaz, Abishai Dmello, Wei Zhong, and Richard Zanibbi. 2020. MathSeer: A Math-Aware Search Interface with Intuitive Formula Editing, Reuse, and Lookup. In *Advances in Information Retrieval*, Joemon M. Jose, Emine Yilmaz, João Magalhães, Pablo Castells, Nicola Ferro, Mário J. Silva, and Flávio Martins (Eds.). Springer International Publishing, Cham, 470–475.

[18] Ricardo M Oliveira, Flavio B Gonzaga, Valmir C Barbosa, and Geraldo B Xexéo. 2017. A distributed system for SearchOnMath based on the Microsoft BizSpark program. *arXiv preprint arXiv:1711.04189* (2017).

[19] Piotr Adam Praczyk and Javier Nogueras-Iso. 2013. Automatic extraction of figures from scientific publications in high-energy physics. *Information Technology and Libraries* 32, 4 (2013), 25–52.

[20] Ayush Kumar Shah, Abhisek Dey, and Richard Zanibbi. 2021. A Math Formula Extraction and Evaluation Framework for PDF Documents. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*. Springer, 19–34.

[21] Volker Sorge, Akashdeep Bansal, Neha M. Jadhav, Himanshu Garg, Ayushi Verma, and Meenakshi Balakrishnan. 2020. Towards generating web-accessible STEM documents from PDF. In *W4A*. ACM, 19:1–19:5.

[22] Keita Del Valle Wangari, Richard Zanibbi, and Anurag Agarwal. 2014. Discovering real-world use cases for a multimodal math search interface. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 947–950.

[23] Richard Zanibbi and Dorothea Blostein. 2012. Recognition and retrieval of mathematical expressions. *International Journal on Document Analysis and Recognition (IJDAR)* 15, 4 (2012), 331–357.

[24] Wei Zhong, Yuqing Xie, and Jimmy Lin. 2022. Applying Structural and Dense Semantic Matching for the ARQMath Lab 2022, CLEF. *Proceedings of the Working Notes of CLEF 2022* (2022), 5–8.