MathSeer: A Math-Aware Search Interface with Intuitive Formula Editing, Reuse, and Lookup[⋆]

Gavin Nishizawa, Jennifer Liu, Yancarlos Diaz, Abishai Dmello, Wei Zhong, and Richard Zanibbi

Rochester Institute of Technology, Rochester NY 14623, USA
{ghn6069,jwt7689,yxd3549,ad7527,wxz8033,rxzvcs}@rit.edu
https://www.cs.rit.edu/~dprl/mathseer

Abstract. There has been growing interest in math-aware search engines that support retrieval using both formulas and keywords. An important unresolved issue is the design of search interfaces: for wide adoption, they must be engaging and easy-to-use, particularly for non-experts. The MathSeer interface addresses this with straightforward formula creation, editing, and lookup. Formulas are stored in 'chips' created using handwriting, LaTeX, and images. MathSeer sessions are also stored at automatically generated URLs that save all chips and their editing history. To avoid re-entering formulas, chips can be reused, edited, or used in creating other formulas. As users enter formulas, our novel autocompletion facility returns entity cards searchable by formula or entity name, making formulas easy to (re)locate, and descriptions of symbols and notation available before queries are issued.

Keywords: Mathematical information retrieval · User interface design · Multimodal input

1 Introduction

Math-aware search engines supporting keyword and formula search have been around since at least 2003, when the Digital Library of Mathematical Functions[1] supported LaTeX in queries [13]. The new information that sophisticated math retrieval would provide, such as more easily locating definitions of symbols and other notations, finding usage, proofs and mathematical properties across disciplines, and compiling information on applications (e.g., variations of the log loss for machine learning) has stimulated work in math-aware search, alongside parallel developments in math question answering within the Natural Language Processing community [2]. To realize their full potential, math-aware search interfaces must be engaging and easy-to-use for different levels of expertise, and particularly for non-experts (e.g., students in middle school).

[⋆] Supported by National Science Foundation (USA) and Alfred P. Sloan Foundation.

[1] https://dlmf.nist.gov

2 Interface Design Elements

Formula Entry. Let's first consider the problem of creating formulas. While formulas such as '$y = x^2 + 1$' can be easily written in LaTeX, others such as:

$$\boldsymbol{\nabla} \times \boldsymbol{F} = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}\right)\mathbf{i} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}\right)\mathbf{j} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right)\mathbf{k} \tag{1}$$

are large, complex, and contain symbols that many non-experts cannot name let alone express in a query. Despite this, most math-aware search engines are restricted to two forms of input: 1) LaTeX (or MathML) entry in text boxes, and 2) visual template editors similar to the Microsoft Equation Editor [11,13]. Many users find template editors confining, and so the text box approach is the most common, often in combination with a palette used to insert symbols and structures in the entry box. Text input is used by most online math-aware systems, including DLMF [8], WebMIAS [10], Math WebSearch [3], Wolfram Alpha, SymboLab, SearchOnMath, and the (now-defunct) Springer LaTeX Search.

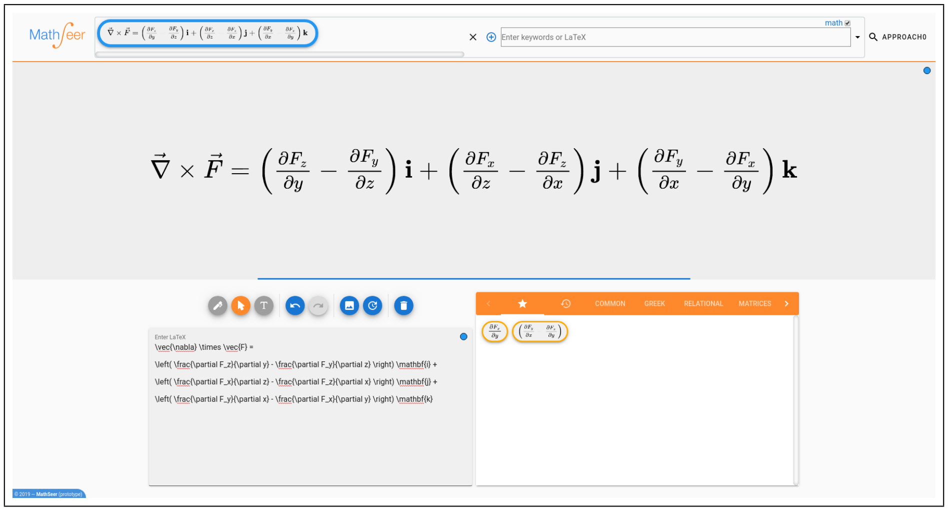
Two challenges for text-based input are 1) most users are unfamiliar with LaTeX (even fewer know MathML), and 2) rendered formulas are shown separately from input, leading to users having difficulty locating entry errors [14]. Appealing solutions to these issues are handwritten formula input, formula image upload, and supporting the analogy of physically moving symbols around on a page [15]. These are key design elements in the MathSeer interface. In one study, a majority of the undergraduate participants reported preferring drawing over typing formulas given a choice between the two [12]. They also expressed formulas with handwriting that they could not using a keyboard (e.g., $\binom{4}{2}$).

To address these issues, our MathSeer search interface (see Figure 1) allows formula input using a combination of typing LaTeX, uploading formula images, and drawing formulas by hand.[2] In MathSeer handwritten symbols are recognized each time a user stops drawing for a short time. Pressing a button recognizes formula structure, and copies the LaTeX result into the panel at the bottom-left of the interface. The LaTeX can then be edited, with a rendering of the formula updated in real-time (e.g., to quickly change 'p' to 'P'). At bottom-right, palettes containing symbols and structures may be used to insert corresponding LaTeX at the cursor position in the LaTeX panel.

Images may be dragged-and-dropped on the canvas or uploaded using a button that presents a file navigation pop-up window. This produces a formula 'chip' on the canvas, which can be used directly in a query, edited, or used in constructing other formulas. A line-of-sight graph-based parsing technique is used to recognize formula images and handwritten formulas [4].

Users can freely alternate between drawing and manipulating symbols on the canvas, uploading images, and editing the LaTeX panel contents. Robust undo/redo operations are provided to easily reverse operations. Formulas in the query bar can be chosen for editing by clicking on them, allowing for quick switching between formulas. Mansouri et al. found that users search for math

[2] Video Demonstration: https://cs.rit.edu/~dprl/mathseer/demos



$$\vec{\nabla} \times \vec{F} = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \right) \mathbf{i} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \right) \mathbf{j} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) \mathbf{k}$$

```
\vec{\nabla} \times \vec{F} =
\left( \frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} \right) \mathbf{i} +
\left( \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x} \right) \mathbf{j} +
\left( \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right) \mathbf{k}
```

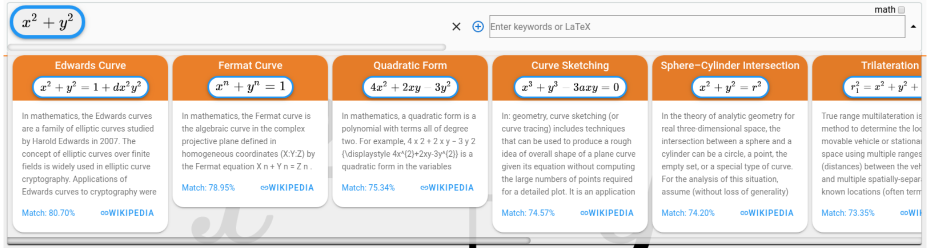
Fig. 1. MathSeer interface. Query formulas and keywords are 'chips' at top left; keywords are entered using the box at top right. Formulas are created by manipulating symbols on the canvas, uploading formula images, and editing LaTeX in the panel at bottom left. At bottom-right is a panel for 'favorite' formulas (two are shown here), the formula history, and palettes for symbols and structures to insert in the LaTeX panel.

with keywords or in the context of a question [6]. In order to help the user add additional information for their query, MathSeer also supports keywords in their search queries (see Figure 1).

Formula Containers and Reuse ('Chips'). Handwritten formula entry is convenient for small expressions, but for large expressions such as Equation 1 handwriting is slow [12], and accurate recognition is challenging [5]. Users may also want to avoid re-entering formulas, and to share formulas with others [12].

MathSeer introduces a new model for formula reuse, flexible containers that we call formula 'chips.' Figure 1 shows a chip in the query bar, and there are two 'favorite' chips in a list at bottom-right. Chips can be created and used in a number of ways. In addition to the formula creation operations described above, chips can be created by selecting symbols on the canvas and 'popping' them up into a chip. All formula chips have their creation history automatically recorded, and are stored in a 'history' menu in the symbol palette panel. On the canvas, chips may be easily moved, resized, and 'pushed' onto the canvas (i.e., the symbols on the chip are added to the canvas, and the chip disappears).

Chips have two possible states: 'recognized' chips containing a LaTeX string, and 'template' chips representing only symbols on a canvas. Chips that are 'favorites' are shown using an orange border, and are either a recognized or template chip. As an example use for **template (grey) chips**, '$\int dx$' with a large space in the middle can be used as a template to quickly create other formulas with an integral, by dragging and dropping the chip from the favorites or history tab in the palette panel to the canvas. **Recognized (blue) chips** in



Edwards Curve

$x^2 + y^2 = 1 + dx^2y^2$

In mathematics, the Edwards curves are a family of elliptic curves studied by Harold Edwards in 2007. The concept of elliptic curves over finite fields is widely used in elliptic curve cryptography. Applications of Edwards curves to cryptography were

Match: 80.70% WIKIPEDIA

Fermat Curve

$x^n + y^n = 1$

In mathematics, the Fermat curve is the algebraic curve in the complex projective plane defined in homogeneous coordinates (X:Y:Z) by the Fermat equation X n + Y n = Z n .

Match: 78.95% WIKIPEDIA

Quadratic Form

$4x^2 + 2xy - 3y^2$

In mathematics, a quadratic form is a polynomial with terms all of degree two. For example, 4 x 2 + 2 x y − 3 y 2 {\displaystyle 4x^{2}+2xy-3y^{2}} is a quadratic form in the variables

Match: 75.34% WIKIPEDIA

Curve Sketching

$x^3 + y^3 - 3axy = 0$

In: geometry, curve sketching (or curve tracing) includes techniques that can be used to produce a rough idea of overall shape of a plane curve given its equation without computing the large numbers of points required for a detailed plot. It is an application

Match: 74.57% WIKIPEDIA

Sphere−Cylinder Intersection

$x^2 + y^2 = r^2$

In the theory of analytic geometry for real three-dimensional space, the intersection between a sphere and a cylinder can be a circle, a point, the empty set, or a special type of curve. For the analysis of this situation, assume (without loss of generality)

Match: 74.20% WIKIPEDIA

Trilateration

$r_1^2 = x^2 + y^2 +$

True range multilateration is method to determine the loc movable vehicle or stationa space using multiple ranges (distances) between the veh and multiple spatially-separ known locations (often term

Match: 73.35% WI

Fig. 2. Expanded auto-complete results displaying entity cards with similar formulas.

the history and favorites tabs in the palette panel can also be used like palette buttons - clicking on them inserts their interpretation in the LaTeX panel, making it easy to re-use and insert large formulas. Chips may also be exported as images with metadata containing all chip data, allowing chip images to be later reused in MathSeer (e.g., using drag-and-drop) or shared with others (e.g., over email). Using chips for formula containers was inspired by the Approach0 interface.[3]

MathSeer records the entire editing session, including all formula chips using an automatically generated URL that users can revisit later. The idea to use a URL to record editing state came from discussions with creators of the 2dsearch system [9].

Math Entity Cards. To support formula autocompletion using online data (e.g., from Wikidata), we use a new type of entity card that provides concept names and descriptions for formulas. We use these to provide names and descriptions for individual symbols and formulas in real-time as they are entered [1]. Formula search over the card collection is done using Tangent-CFT embedding vectors [7]. In addition, we will soon allow formulas to be quickly found by searching concept names on cards (e.g., typing 'Pyt' brings up the card and formula for the Pythagorean Theorem). Further, we plan to allow users to create their own entity cards for formula chips. An illustration of math entity cards is shown in Figure 2. This view is expanded to show the full cards; in the unexpanded view only the formulas and titles are visible.

3 Conclusion and Future Work

The MathSeer interface addresses limitations of the standard text box + symbol palette formula entry technique common in math-aware search interfaces. MathSeer's interface supports multimodal formula editing through handwritten, LaTeX, and image input. We have introduced formula chips, a new container to support storage, reuse, editing, and sharing of formulas. The chip creation history and favorites list support quick query reformulation and reuse. In future work, we are considering manual editing operations to define spatial relationships between symbols and/or sub-expressions to avoid recognizing complex formulas.

[3] https://www.approach0.xyz/search

References

1. Dmello, A.: Representing Mathematical Concepts Associated With Formulas Using Math Entity Cards. Master's thesis, Rochester Institute of Technology (2019), https://scholarworks.rit.edu/theses/10238
2. Hopkins, M., Bras, R.L., Petrescu-Prahova, C., Stanovsky, G., Hajishirzi, H., Koncel-Kedziorski, R.: Semeval-2019 task 10: Math question answering. In: Proc. SemEval@NAACL-HLT 2019. pp. 893–899. Minneapolis, MN, USA (2019)
3. Kohlhase, M., Matican, B., Prodescu, C.: Mathwebsearch 0.5: Scaling an open formula search engine. In: Proc. CICM. pp. 342–357. Bremen, Germany (2012)
4. Mahdavi, M., Condon, M., Davila, K., Zanibbi, R.: LPGA: Line-of-sight Parsing with Graph-based Attention for math formula recognition. In: Proc. ICDAR. pp. 647–654. Sydney, Australia (2019)
5. Mahdavi, M., Zanibbi, R., Mouchère, H., Viard-Gaudin, C., Garain, U.: ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection. In: Proc. ICDAR. pp. 1533–1538. Sydney, Australia (2019)
6. Mansouri, B., Zanibbi, R., Oard, D.W.: Characterizing searches for mathematical concepts. In: 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL). pp. 57–66 (2019)
7. Mansouri, B., Rohatgi, S., Oard, D.W., Wu, J., Giles, C.L., Zanibbi, R.: Tangent-cft: An embedding model for mathematical formulas. In: Proc. ICTIR 2019. pp. 11–18 (2019)
8. Miller, B.R., Youssef, A.: Technical aspects of the digital library of mathematical functions. Ann. Math. Artif. Intell. **38**(1-3), 121–136 (2003)
9. Russell-Rose, T., Chamberlain, J., Kruschwitz, U.: Rethinking 'advanced search': A new approach to complex query formulation. In: Proc. ECIR. pp. 236–240. Cologne, Germany (2019)
10. Sojka, P., Ruzicka, M., Novotný, V.: Mias: Math-aware retrieval in digital mathematical libraries. In: Proc. CIKM. pp. 1923–1926. Torino, Italy (2018)
11. Wangari, K.: Discovering real-world usage scenarios for a multimodal math search interface. Master's thesis, Rochester Institute of Technology (Dec 2013)
12. Wangari, K.D.V., Zanibbi, R., Agarwal, A.: Discovering real-world use cases for a multimodal math search interface. In: Proc. SIGIR. pp. 947–950. Gold Coast, Australia (2014)
13. Zanibbi, R., Blostein, D.: Recognition and retrieval of mathematical expressions. IJDAR **15**(4), 331–357 (2012)
14. Zanibbi, R., Novins, K., Arvo, J., Zanibbi, K.: Aiding manipulation of handwritten mathematical expressions through style-preserving morphs. In: Proc. Graphics Interface. pp. 127–134. Ottawa, Canada (2001)
15. Zanibbi, R., Orakwue, A.: Math search for the masses: Multimodal search interfaces and appearance-based retrieval. In: Proc. CICM. pp. 18–36. Washington, DC, USA (2015)