Discovering Mathematical Objects of Interest—A Study of Mathematical Notations

André Greiner-Petter[1], Moritz Schubotz[1,2], Fabian Müller[2], Corinna Breitinger[1,5], Howard S. Cohl[3], Akiko Aizawa[4], and Bela Gipp[1,5]

[1]University of Wuppertal, Germany
(andre.greiner-petter@zbmath.org, {last}@uni-wuppertal.de)
[2]FIZ-Karlsruhe, Germany ({first.last}@fiz-karlsruhe.de)
[3]National Institute of Standards and Technology, Mission Viejo, California, U.S.A. (howard.cohl@nist.gov)
[4]National Institute of Informatics, Japan ({last}@nii.ac.jp)
[5]University of Konstanz, Germany ({first.last}@uni-konstanz.de)

February 19, 2020

Abstract

Mathematical notation, i.e., the writing system used to communicate concepts in mathematics, encodes valuable information for a variety of information search and retrieval systems. Yet, mathematical notations remain mostly unutilized by today's systems. In this paper, we present the first in-depth study on the distributions of mathematical notation in two large scientific corpora: the open access arXiv (2.5B mathematical objects) and the mathematical reviewing service for pure and applied mathematics zbMATH (61M mathematical objects). Our study lays a foundation for future research projects on mathematical information retrieval for large scientific corpora. Further, we demonstrate the relevance of our results to a variety of use-cases. For example, to assist semantic extraction systems, to improve scientific search engines, and to facilitate specialized math recommendation systems.

The contributions of our presented research are as follows: (1) we present the first distributional analysis of mathematical formulae on arXiv and zbMATH; (2) we retrieve relevant mathematical objects for given textual search queries (e.g., linking $P_n^{(\alpha,\beta)}(x)$ with 'Jacobi polynomial'); (3) we extend zbMATH's search engine by providing relevant mathematical formulae; and (4) we exemplify the applicability of the results by presenting auto-completion for math inputs as the first contribution to math recommendation systems. To expedite future research projects, we have made available our source code and data.

1

1   Introduction

Taking into account mathematical notation in the literature leads to a better understanding of scientific literature on the Web and allows one to make use of semantic information in specialized Information Retrieval (IR) systems. Nowadays applications in Math Information Retrieval (MathIR) [21], such as search engines [4, 8, 10, 11, 13, 22, 25], semantic extraction systems [23, 28, 30], recent efforts in math embeddings [26, 35, 37, 43], and semantic tagging of math formulae [16, 31] either consider an entire equation as one entity or only focus on single symbols. Since math expressions often contain meaningful and important subexpressions, these applications could benefit from an approach that lies between the extremes of examining only individual symbols or considering an entire equation as one entity. Consider for example, the explicit definition for Jacobi polynomials [46, (18.5.7)]

$$P_n^{(\alpha,\beta)}(x) = \frac{\Gamma(\alpha+n+1)}{n!\Gamma(\alpha+\beta+n+1)} \sum_{m=0}^{n} \binom{n}{m} \frac{\Gamma(\alpha+\beta+n+m+1)}{\Gamma(\alpha+m+1)} \left(\frac{x-1}{2}\right)^m. \tag{1}$$

The *interesting* components in this equation are $P_n^{(\alpha,\beta)}(x)$ on the left-hand side, and the appearance of the gamma function $\Gamma(s)$ on the right-hand side, implying a direct relationship between Jacobi polynomials and the gamma function. Considering the entire expression as a single object misses this important relationship. On the other hand, focusing on single symbols can result in the misleading interpretation of Γ as a variable and $\Gamma(\alpha+n+1)$ as a multiplication between Γ and $(\alpha+n+1)$. A system capable of identifying the important components, such as $P_n^{(\alpha,\beta)}(x)$ or $\Gamma(\alpha+n+1)$, is therefore desirable. Hereafter, we define these components as *Mathematical Objects of Interest* (MOIs) [37].

The *importance* of math objects is a somewhat imprecise description and thus difficult to measure. Currently, not much effort has been made in identifying meaningful subexpressions. Kristianto et al. [28] introduced dependency graphs between formulae. With this approach, they were able to build dependency graphs of mathematical expressions, but only if the expressions appeared as single expressions in the context. For example, if $\Gamma(\alpha+n+1)$ appears as a stand-alone expression in the context, the algorithm will declare a dependency with Equation (1). However, it is more likely that different forms, such as $\Gamma(s)$, appear in the context. Since this expression does not match any subexpression in Equation (1), the approach cannot establish a connection with $\Gamma(s)$. Kohlhase et al. studied in [27, 33, 34] another approach to identify essential components in formulae. They performed eye-tracking studies to identify important areas in rendered mathematical formulae. While this is an interesting approach that allows one to learn more about the insights of human behaviors of reading and understanding math, it is inaccessible for extensive studies.

This paper presents the first extensive frequency distribution study of mathematical equations in two large scientific corpora, the e-Print archive arXiv.org (hereafter referred to as arXiv[1]) and the international reviewing service for pure

---

[1] https://arxiv.org/ [Accessed: Sep. 1, 2019]

and applied mathematics zbMATH[2]. We will show that math expressions, similar to words in natural language corpora, also obey Zipf's law [15], and therefore follows a *Zipfian* distribution. Related research projects observed a relation to Zipf's law for single math symbols [16, 23]. In the context of quantitative linguistics, Zipf's law states that given a text corpus, the frequency of any word is inversely proportional to its rank in the frequency table. Motivated by the similarity to linguistic properties, we will present a novel approach for ranking formulae by their relevance via a customized version of the ranking function BM25 [7]. We will present results that can be easily embedded in other systems in order to distinguish between common and uncommon notations within formulae. Our results lay a foundation for future research projects in MathIR.

Fundamental knowledge on frequency distributions of math formulae is beneficial for numerous applications in MathIR, ranging from educational purposes [3] to math recommendation systems, search engines [22, 25], and even automatic plagiarism detection systems [29, 39, 41]. For example, students can search for the conventions to write certain quantities in formulae; document preparation systems can integrate an auto-completion or auto-correction service for math inputs; search or recommendation engines can adjust their ranking scores with respect to standard notations; and plagiarism detection systems can estimate whether two identical formulae indicate potential plagiarism or are just using the conventional notations in a particular subject area. To exemplify the applicability of our findings, we present a textual search approach to retrieve mathematical formulae. Further, we will extend zbMATH's faceted search by providing facets of mathematical formulae according to a given textual search query. Lastly, we present a simple auto-completion system for math inputs as a contribution towards advancing mathematical recommendation systems. Further, we show that the results provide useful insights for plagiarism detection algorithms. We provide access to the source code, the results, and extended versions of all of the figures appearing in this paper at `https://github.com/ag-gipp/FormulaCloudData`.

Related Work: Today, mathematical search engines index formulae in a database. Much effort has been undertaken to make this process as efficient as possible in terms of precision and runtime performance [4, 8, 14, 24, 25]. The generated databases naturally contain the information required to examine the distributions of the indexed mathematical formulae. Yet, no in-depth studies of these distributions have been undertaken. Instead, math search engines focus on other aspects, such as devising novel similarity measures and improving runtime efficiency. This is because the goal of math search engines is to retrieve relevant (i.e., similar) formulae which correspond to a given search query that partially [13, 14, 22] or exclusively [8, 11, 25] contains formulae. However, for a fundamental study of distributions of mathematical expressions, no similarity measures nor efficient lookup or indexing is required. Thus, we use the general-purpose query language XQuery and employ the BaseX[3] implementation. Ba-

---

[2] `https://zbmath.org` [Accessed: Sep. 1, 2019]
[3] `http://basex.org/` [Accessed: Sep. 2019]; We used BaseX 9.2 for our experiments.

3

seX is a free open-source XML database engine, which is fully compatible with the latest XQuery standard [6, 19]. Since our implementations rely on XQuery, we are able to switch to any other database which allows for processing via XQuery.

2 Data Preparation

LaTeX is the de facto standard for the preparation of academic manuscripts in the fields of mathematics and physics [5]. Since LaTeX allows for advanced customizations and even computations, it is challenging to process. For this reason, LaTeX expressions are unsuitable for an extensive distribution analysis of mathematical notations. For mathematical expressions on the web, the XML formatted MathML[4] is the current standard, as specified by the World Wide Web Consortium (W3C). The tree structure and the fixed standard, i.e., MathML tags, cannot be changed, thus making this data format reliable. Several available tools are able to convert from LaTeX to MathML [36] and various databases are able to index XML data. Thus, for this study, we have chosen to focus on MathML. In the following, we investigate the databases arXMLiv (08/2018) [32] and zbMATH[5] [40].

The arXMLiv dataset ($\approx$1.2 million documents) contains HTML5 versions of the documents from the e-Print archive arXiv.org. The HTML5 documents were generated from the TeX sources via LaTeXML [45]. LaTeXML converted all mathematical expressions into MathML with parallel markup, i.e., presentation and content MathML. In this study we only consider the subsets *no-problem* and *warning*, which generated no errors during the conversion process. Nonetheless, the MathML data generated still contains some errors or falsely annotated math. For example, we discovered several instances of affiliation and footnotes, SVG[6] and other unknown tags, encoded in MathML. Regarding the footnotes, we presumed that authors falsely used mathematical environments for generating footnote or affiliation marks. We used the TeX string, provided as an attribute in the MathML data, to filter out expressions that match the string '{}^{*}', where '*' indicates any possible expression. In addition, we filtered out SVG and other unknown tags. We assume that these expressions were generated by mistake due to limitations of LaTeXML. The final arXiv dataset consisted of 841,008 documents which contained at least one mathematical formula. The dataset contained a total of 294,151,288 mathematical expressions.

In addition to arXiv, we investigated zbMATH, an international reviewing service for pure and applied mathematics which contains abstracts and reviews of articles, hereafter uniformly called abstracts, mainly from the domains of pure and applied mathematics. The abstracts in zbMATH are formatted in TeX [40]. To be able to compare arXiv and zbMATH, we manually generated MathML via LaTeXML for each mathematical formula in zbMATH and performed the

---

[4] https://www.w3.org/TR/MathML3/ [Accessed: Sep. 1, 2019]
[5] https://zbmath.org/ [Accessed: Sep. 1, 2019]
[6] Scalable Vector Graphics

same filters as used for the arXiv documents. The zbMATH dataset contained 2,813,451 abstracts, of which 1,349,297 contained at least one formula. In total, the dataset contained 11,747,860 formulae. Even though the total number of formulae is smaller compared to arXiv, we hypothesize that math formulae in abstracts are particularly meaningful.

2.1 Data Wrangling

Since we focused on the frequency distributions of visual expressions, we only considered presentational MATHML (pMML). Rather than normalizing the pMML data, e.g., via MathMLCan [9], which would also change the tree structure and visual core elements in pMML, we only eliminated the attributes. These attributes are used for minor visual changes, e.g., stretched parentheses or inline limits of sums and integrals. Thus, for this first study, we preserved the core structure of the pMML data, which might provide insightful statistics for the MATHML community to further cultivate the standard. After extracting all MATHML expressions, filtering out falsely annotated math and SVG tags, and eliminating unnecessary attributes and annotations, the datasets required 83GB of disk space for arXiv and 6GB for zbMATH, respectively.

In the following, we indexed the data via BaseX. The indexed datasets required a disk space of 143.9GB in total (140GB for arXiv and 3.9GB for zbMATH). Due to the limitations[7] of databases in BaseX, it was necessary to split our datasets into smaller subsets. We split the datasets according to the 20 major article categories of arXiv[8] and classifications of zbMATH. To increase performance, we use BaseX in a server-client environment. We experienced performance issues in BaseX when multiple clients repeatedly requested data from the same server in short intervals. We determined that the best workaround for this issue was to launch BaseX servers for each database, i.e., each category/classification.

Mathematical expressions often consist of multiple meaningful subexpressions, which we defined as MOIs. However, without further investigation of the

Listing 1: MathML representation of $P_n^{(\alpha,\beta)}(x)$.

```
1   <math><mrow>
2    <msubsup>
3     <mi>P</mi>
4     <mi>n</mi>
5     <mrow>
6      <mo>(</mo>
7      <mi>α</mi>
8      <mo>,</mo>
9      <mi>β</mi>
10     <mo>)</mo>
11     <mo></mo>
12    </mrow>
13   </msubsup>
14   <mo></mo>
15   <mrow>
16    <mo>(</mo>
17    <mi>x</mi>
18    <mo>)</mo>
19   </mrow>
20  </mrow></math>
```

---

[7] A detailed overview of the limitations of BaseX databases can be found at `http://docs.basex.org/wiki/Statistics` [Accessed: Sep. 1, 2019].

[8] The arXiv categories *astro-ph* (astro physics), *cond-mat* (condensed matter), and *math* (mathematics) were still too large for a single database. Thus, we split those categories into two equally sized parts.

context, it is impossible to determine meaningful subexpressions. As a consequence, every equation is a potential MOI on its own and potentially consists of multiple other MOIs. For an extensive frequency distributional analysis, we aim to discover all possible mathematical objects. Hence, we split every formula into its components. Since MathML is an XML data format (essentially a tree-structured format), we define subexpressions of equations as subtrees of its MathML format.

Listing 1 illustrates a Jacobi polynomial $P_n^{(\alpha,\beta)}(x)$ in pMML. The `<mo>` element on line 14 contains the *invisible times* UTF-8 character. By definition, the `<math>` element is the root element of MathML expressions. Since we cut off all other elements besides pMML nodes, each `<math>` element has one and only one child element[9]. Thus, we define the child element of the `<math>` element as the root of the expression. Starting from this root element, we explore all subexpressions. For this study, we presume that every meaningful mathematical object (i.e., MOI) must contain at least one identifier.

Hence, we only study subtrees which contain at least one `<mi>` node. Identifiers, in the sense of MathML, are *'symbolic names or arbitrary text'* [10], e.g., single Latin or Greek letters. Identifiers do not contain special characters (other than Greek letters) or numbers. As a consequence, arithmetic expressions, such as $(1+2)^2$, or sequences of special characters and numbers, such as $\{1,2,...\} \cap \{-1\}$, will not appear in our distributional analysis. However, if a sequence or arithmetic expression consists of an identifier somewhere in the pMML tree (such as in $\{1,2,...\} \cap A$), the entire expression will be recognized. The Jacobi polynomial $P_n^{(\alpha,\beta)}(x)$, therefore consists of the following subexpressions: $P_n^{(\alpha,\beta)}$, (α,β), (x), and the single identifiers P, n, α, β, and x. The entire expression is also a mathematical object. Hence, we take entire expressions with an identifier into account for our analysis. In the following, the set of subexpressions will be understood to include the expression itself.

For our experiments, we also generated a string representation of the MathML data. The string is generated recursively by applying one of two rules for each node: (i) if the current node is a leaf, the node-tag and the content will be merged by a colon, e.g., `<mi>x</mi>` will be converted to `mi:x`; (ii) otherwise the node-tag wraps parentheses around its content and separates the children by a comma, e.g., `<mrow><mo>(</mo><mi>x</mi><mo>)</mo></mrow>` will be converted to `mrow(mo:(,mi:x,mo:))`. Furthermore, the special UTF-8 characters for invisible times (U+2062) and function application (U+2061) are replaced by `ivt` and `fa`, respectively. For example, the gamma function with argument $x+1$, $\Gamma(x+1)$ would be represented by

$$\text{mrow(mi:}\Gamma\text{,mo:ivt,mrow(mo:(,mrow(mi:x,mo:+,mn:1),mo:))).} \tag{2}$$

Between Γ and $(x+1)$, there would most likely be the special character for *invisible times* rather than for *function application*, because LaTeXML is not able to parse Γ as a function. Note that this string conversion is a bijective

---

[9]Sequences are always nested in an `<mrow>` element.
[10]https://www.w3.org/TR/MathML3/chapter3.html [Accessed: Sep. 1, 2019]

mapping. The string representation reduces the verbose XML format to a more concise presentation. Thus, an equivalence check between two expressions is more efficient.

2.2 Complexity of Math

Mathematical expressions can become complex and lengthy. The tree structure of MATHML allows us to introduce a measure that reflects the complexity of mathematical expressions. More complex expressions usually consist of more extensively nested subtrees in the MATHML data. Thus, we define the complexity of a mathematical expression by the maximum depth of the MATHML tree. In XML the content of a node and its attributes are commonly interpreted as children of the node. Thus, we define the depth of a single node as 1 rather than 0, i.e., single identifiers, such as `<mi>P</mi>`, have a complexity of 1. The Jacobi polynomial from Listing 1 has a complexity of 4.

We perform the extraction of subexpressions from MATHML in BaseX. The algorithm for the extraction process is written in XQuery. The algorithm traverses recursively downwards from the root to the leaves. In each iteration, it checks whether there is an identifier, i.e., `<mi>` element, among the descendants of the current node. If there is no such element, the subtree will be ignored. It seems counterintuitive to start from the root and check if an identifier is among the descendants rather than starting at each identifier and traversing upwards to the root. If an XQuery requests a node in BaseX, BaseX loads the entire subtree of the requested node into the cache (up to a specified size). If the algorithm traverses upwards through the MATHML tree, the XQuery will trigger database requests in every iteration. Hence, the downwards implementation performs better, since there is only one database request for every expression rather than for every subexpression.

Since we only minimize the pMML data rather than normalizing it, two identically rendered expressions may have different complexities. For instance, `<mrow><mi>x</mi></mrow>` consists of two distinct subexpressions, but both of them are displayed the same. Another problem often appears for arrays or similar visually complicated structures. The extracted expressions are not necessarily logical subexpressions. We will consider applying more advanced embedding techniques such as special tokenizers [14], symbol layout trees [24, 25], and a MATHML normalization via MathMLCan [9] in future research to overcome these issues.

3 Frequency Distributions of Mathematical Formulae

By splitting each formula into subexpressions, we generated longer documents and a bias towards low complexities. Note that, hereafter, we only refer to the mathematical content of documents. Thus, the length of a document refers to the number of math formulae—here the number of subexpressions—in the

Category	arXiv	zbMATH
Documents	841,008	1,349,297
Formulae	294,151,288	11,747,860
Subexpressions	2,508,620,512	61,355,307
Unique Subexpressions	350,206,974	8,450,496
Average Document Length	2,982.87	45.47
Average Complexity	5.01	3.89
Maximum Complexity	218	26

Table 1: Dataset overview. Average Document Length is defined as the average number of subexpressions per document.
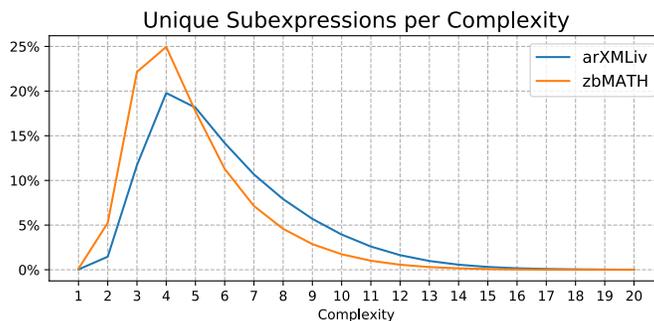
Figure 1: Unique subexpressions for each complexity in arXiv and zbMATH.

document. After splitting expressions into subexpressions, arXiv consists of 2.5B and zbMATH of 61M expressions, which raised the average document length to 2,982.87 for arXiv and 45.47 for zbMATH, respectively.

For calculating frequency distributions, we merged two subexpressions if their string representations were identical. Remember, the string representation is unique for each MathML tree. After merging, arXiv consisted of 350,206,974 unique mathematical subexpressions with a maximum complexity of 218 and an average complexity of 5.01. For high complexities over 70, the formulae show some erroneous structures that might be generated from LaTeXML by mistake. For example, the expression with the highest complexity is a long sequence of a polynomial starting with '$P_4(t_1, t_3, t_7, t_{11}) =$' followed by 690 summands. The complexity is caused by a high number of unnecessarily deeply nested `<mrow>` nodes. The highest complexity with a minimum document frequency of two is 39, which is a continued fraction. Since continued fractions are nested fractions, they naturally have a large complexity. One of the most complex expressions

(complexity 20) with a minimum document frequency of three was the formula

$$\left(\sum_{j_1=1}^{n}\left(\sum_{j_2=1}^{n}\left(\cdots\left(\sum_{j_m=1}^{n}|T(e_{j_1},\ldots,e_{j_m})|^{q_m}\right)^{\frac{q_{m-1}}{q_m}}\cdots\right)^{\frac{q_2}{q_3}}\right)^{\frac{q_1}{q_2}}\right)^{\frac{1}{q_1}}\leq C_{m,p,\mathbf{q}}^{\mathbb{K}}\,\|T\|. \quad (3)$$

In contrast, zbMATH only consisted of 8,450,496 unique expressions with a maximum complexity of 26 and an average complexity of 3.89. One of the most complex expressions in zbMATH with a minimum document frequency of three was

$$M_p(r,f)=\left(\frac{1}{2\pi}\int_0^{2\pi}\left|f\left(re^{i\theta}\right)\right|^p d\theta\right)^{1/p}. \quad (4)$$

As we expected, reviews and abstracts in zbMATH were generally shorter and consisted of less complex mathematical formulae. The dataset also appeared to contain fewer erroneous expressions, since expressions of complexity 25 are still readable and meaningful.

Figure 1 shows the ratio of unique subexpressions for each complexity in both datasets. The figure illustrates that both datasets share a peak at complexity four. Compared to zbMATH, the arXiv expressions are slightly more evenly distributed over the different levels of complexities. Interestingly, complexities one and two are not dominant in either of the two datasets. Single identifiers only make up 0.03% in arXiv and 0.12% in zbMATH, which is comparable to expressions of complexity 19 and 14, respectively. This finding illustrates the problem of capturing semantic meanings for single identifiers rather than for more complex expressions [30]. It also substantiates that entire expressions, if too complex, are not suitable either for capturing the semantic meanings [28]. Instead, a middle ground is desirable, since the most unique expressions in both datasets have a complexity between 3 and 5. Table 1 summarizes the statistics of the examined datasets.

3.1 Zipf's Law

In linguistics, it is well known that word distributions follow Zipf's Law [15], i.e., the r-th most frequent word has a frequency that scales to

$$f(r)\propto\frac{1}{r^{\alpha}} \quad (5)$$

with $\alpha\approx 1$. A better approximation can be applied by a shifted distribution

$$f(r)\propto\frac{1}{(r+\beta)^{\alpha}}, \quad (6)$$

where $\alpha\approx 1$ and $\beta\approx 2.7$. In a study on Zipf's law, Piantadosi [15] illustrated that not only words in natural language corpora follow this law surprisingly accurately, but also many other human-created sets. For instance, in programming languages, in biological systems, and even in music. Since mathematical
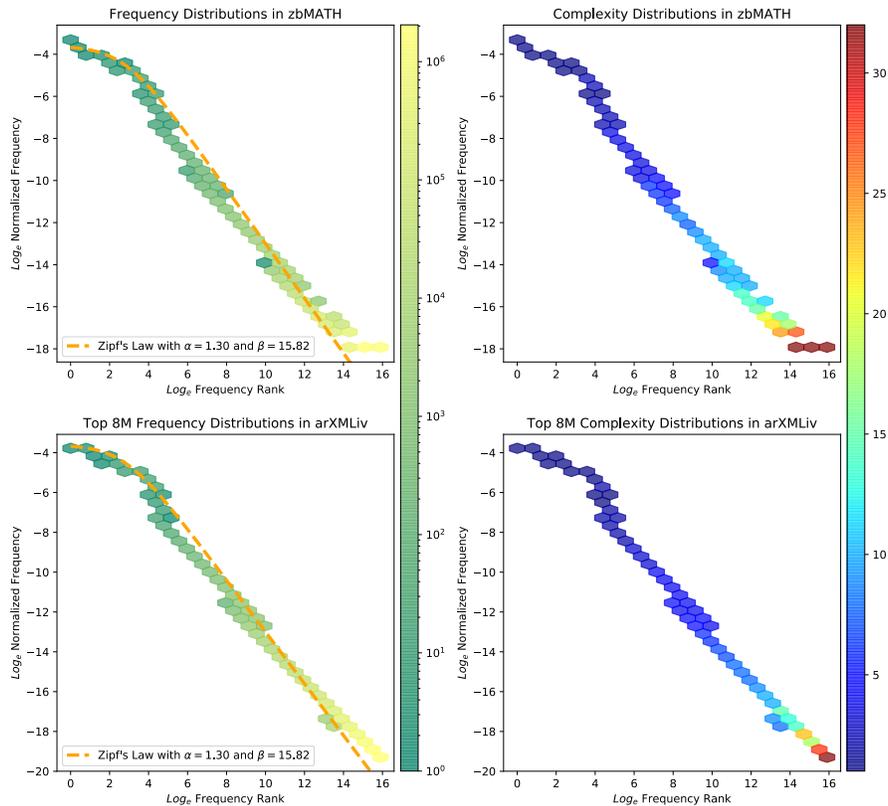
(a) Frequency Distributions (b) Complexity Distributions

Figure 2: Each figure illustrates the relationship between the frequency ranks (x-axis) and the normalized frequency (y-axis) in zbMATH (top) and arXiv (bottom). For arXiv, only the first 8 million entries are plotted to be comparable with zbMATH (≈ 8.5 million entries). Subfigure (a) shades the hexagonal bins from green to yellow using a logarithmic scale according to the number of math expressions that fall into a bin. The dashed orange line represents Zipf's distribution (6). The values for α and β are provided in the plots. Subfigure (b) shades the bins from blue to red according to the maximum complexity in each bin.

communication has derived as the result of centuries of research, it would not be surprising if mathematical notations would also follow Zipf's law. The primary conclusion of the law illustrates that there are some very common tokens against a large number of symbols which are not used frequently. Based on this assumption, we can postulate that a score based on frequencies might be able to measure the peculiarity of a token. The infamous TF-IDF ranking functions and their derivatives [2, 7] have performed well in linguistics for many years and are still widely used in retrieval systems [20]. However, since we split every expression into its subexpressions, we generated an anomalous bias towards shorter, i.e., less complex, formulae. Hence, distributions of subexpressions may not obey Zipf's law.

Figure 2 visualizes a comparison between Zipf's law and the frequency distributions of mathematical subexpressions in arXiv and zbMATH. The dashed orange line visualizes the power law (6). The plots demonstrate that the distributions in both datasets obey this power law. Interestingly, there is not much difference in the distributions between both datasets. Both distributions seem to follow the same power law, with $\alpha = 1.3$ and $\beta = 15.82$. Moreover, we can observe that the developed complexity measure seems to be appropriate, since the complexity distributions for formulae are similar to the distributions for the length of words [15]. In other words, more complex formulae, as well as long words in natural languages, are generally more specialized and thus appear less frequent throughout the corpus. Note that colors of the bins for complexities fluctuate for rare expressions because the color represents the maximum rather than the average complexity in each bin.

3.2   Analyzing and Comparing Frequencies

Figure 3 shows in detail the most frequently used mathematical expressions in arXiv for the complexities 1 to 5. The orange dashed line visible in all graphs represents the normal Zipf's law distribution from Equation (5). We explore the total frequency values without any normalization. Thus, Equation (5) was multiplied by the highest frequency for each complexity level to fit the distribution. The plots in Figure 3 demonstrate that even though the parameter α varies between 0.35 and 0.62, the distributions in each complexity class also obey Zipf's law.

The plots for each complexity class contain some interesting fluctuations. We can spot a set of five single identifiers that are most frequently used throughout arXiv: n, i, x, t, and k. Even though the distributions follow Zipf's law accurately, we can explore that these five identifiers are proportionally more frequently used than other identifiers and clearly separate themselves above the rest (notice the large gap from k to a). All of the five identifiers are known to be used in a large variety of scenarios. Surprisingly, one might expect that common pairs of identifiers would share comparable frequencies in the plots. However, typical pairs, such as x and y, or α and β, possess a large discrepancy.

The plot of complexity two also reveals that two expressions are proportionally more often used than others: (x) and (t). These two expressions appear
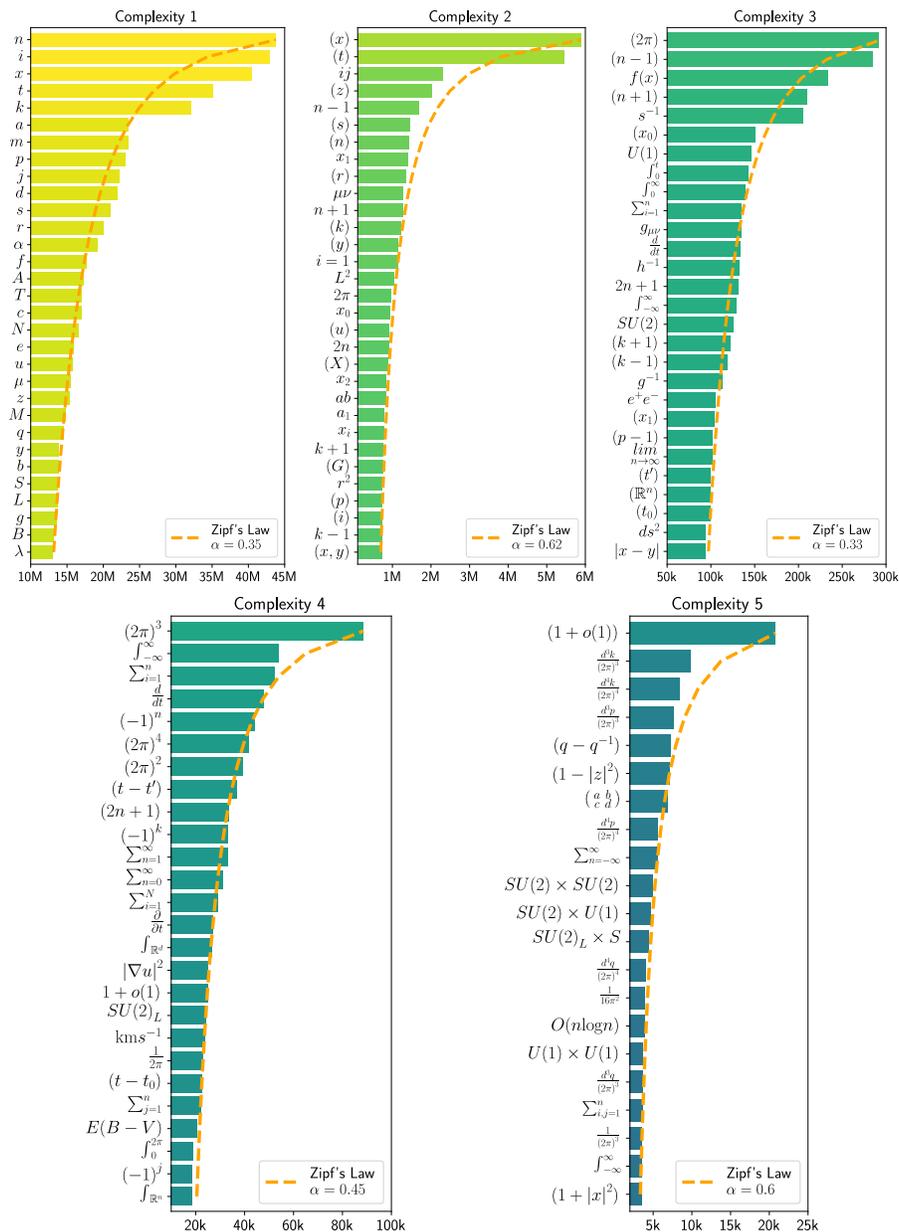
11

Figure 3: Overview of the most frequent mathematical expressions in arXiv for complexities 1-5. The color gradient from yellow to blue represents the frequency in the dataset. Zipf's law (5) is represented by a dashed orange line.

more than three times as often in the corpus than any other expression of the same complexity. On the other hand, the quantitative difference between (x) and (t) is negligible. We may assume that arXiv's primary domain, physics, causes the quantitative disparity between (x), (t), and the other tokens. The primary domain of the dataset becomes more clearly visible for higher complexities, such as $SU(2)$ (C3[11]) or kms^{-1} (C4).

Another surprising property of arXiv is that symmetry groups, such as $SU(2)$, appear to play an essential role in the majority of articles on arXiv, see $SU(2)$ (C3), $SU(2)_L$ (C4), and $SU(2) \times SU(2)$ (C5), among others. The plots of higher complexities[12], which we do not show here, made this even more noticeable. Given a complexity of six, for example, the most frequently used expression was $SU(2)_L \times SU(2)_R$, and for a complexity of seven it was $SU(3) \times SU(2) \times U(1)$. Given a complexity of eight, ten out of the top-12 expressions were from symmetry group calculations.

It is also worthwhile to compare expressions among different levels of complexities. For instance, (x) and (t) appeared almost six million times in the corpus, but $f(x)$ (at position three in C3) was the only expression which contained one of these most common expressions. Note that subexpressions of variations, such as (x_0), (t_0), or $(t - t')$, do not match the expression of complexity two. This may imply that (x), and especially (t), appear in many different scenarios. Further, we can examine that even though (x) is a part of $f(x)$ in only approximately 3% of all cases, it is still the most likely combination. These results are especially useful for recommendation systems that make use of math as input. Moreover, plagiarism detection systems may also benefit from such a knowledge base. For instance, it might be evident that $f(x)$ is a very common expression, but for automatic systems that work on a large scale, it is not clear whether duplicate occurrences of $f(x)$ or $\Xi(x)$ should be scored differently, e.g., in the case of plagiarism detection.

Figure 3 shows only the most frequently occurring expressions in arXiv. Since we already explored a bias towards physics formulae in arXiv, it is worth comparing the expressions present within both datasets. Figure 4 compares the 25-top expressions for the complexities one to four. In zbMATH, we discovered that computer science and graph theory appeared as popular topics, see for example $G = (V, E)$ (in C3 at position 20) and the Bachmann-Landau notations in $O(\log n)$, $O(n^2)$, and $O(n^3)$ (C4 positions 2, 3, and 19).

From Figure 4, we can also deduce useful information for MathIR tasks which focus on semantic information. Current semantic extraction tools [30] or LaTeX parsers [36] still have difficulties distinguishing *multiplications* from *function calls*. For example as mentioned before, LaTeXML [45] adds an *invisible times* character between $f(x)$ rather than a *function application*. Investigating the most frequently used terms in zbMATH in Table 4 reveals that u is most likely considered to be a function in the dataset: $u(t)$ (rank 8), $u(x)$ (rank 13), u_{xx} (rank 16), $u(0)$ (rank 17), $|\nabla u|$ (rank 22). Manual investigations of extended

---

[11]We refer to a given complexity n with Cn, i.e., C3 refers to complexity 3.

[12]More plots showing higher complexities are available at `https://github.com/ag-gipp/FormulaCloudData`
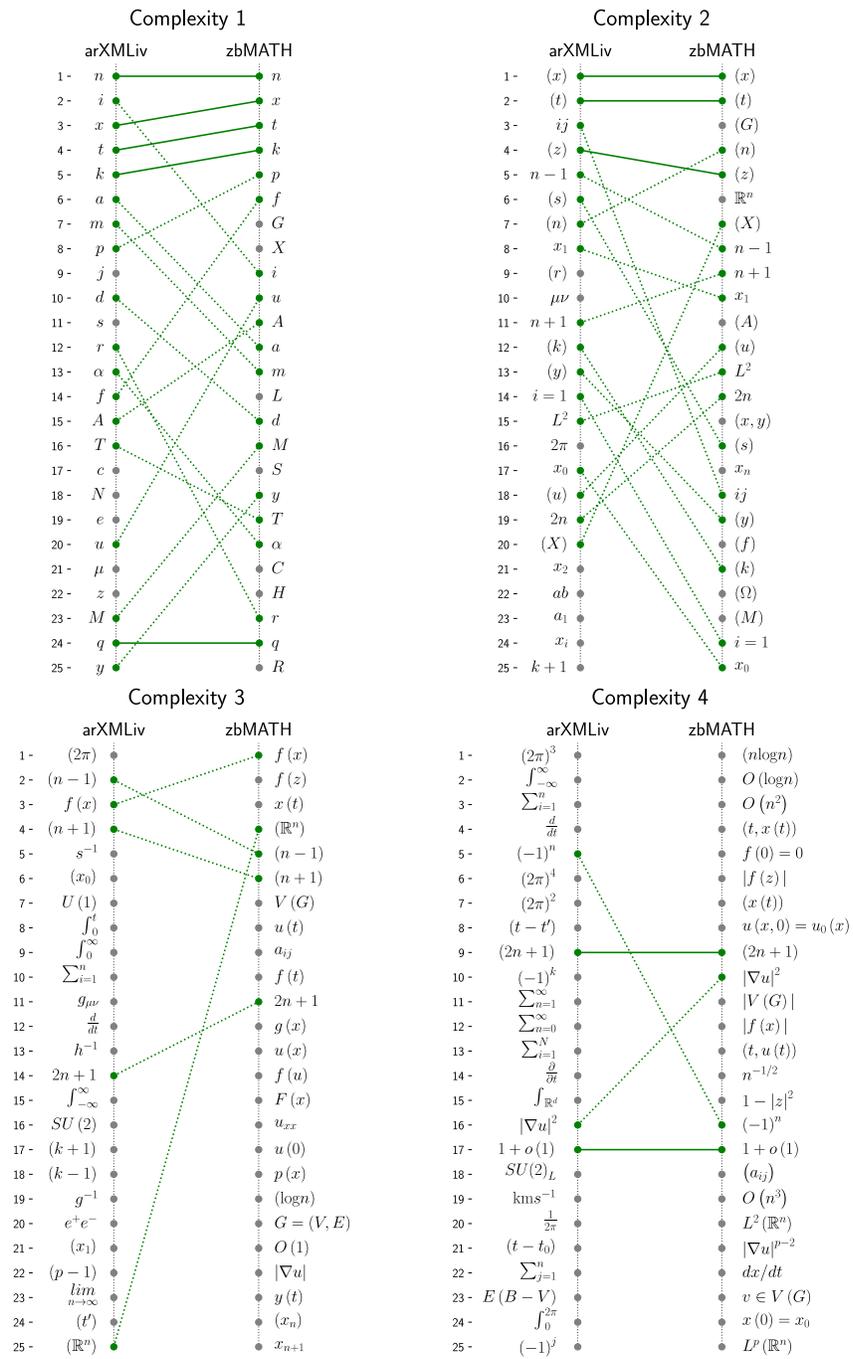
Figure 4: The top-25 most frequent expressions in arXiv (left) and zbMATH (right) for complexities 1-4. A line between both sets indicates a matching set. Bold lines indicate that the matches share a similar rank (distance of 0 or 1).

C3		C4		C5		C6	
114.84	$(n!)$	129.44	$i,j=1,\dots,n$	119.21	$\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$	110.83	$(1+|z|^2)^\alpha$
108.85	ϕ^{-1}	108.52	x_{ij}	112.55	$|f(z)|^p$	105.69	$f(re^{i\theta})$
100.19	z^{n-1}	108.50	$\dot{x}=A(t)x$	110.52	$\left(1+|x|^2\right)$	94.14	$f(z)=z+\sum_{n=2}^{\infty} a_n z^n$
100.06	(c_n)	106.66	$|x-x_0|$	109.19	$|f(x)|^p$	92.33	$\left(|\nabla u|^{p-2}\nabla u\right)$
100.05	$B(G)$	105.52	S^{2n+1}	106.22	$|\nabla u|^2 dx$	87.27	$(\log n/\log\log n)$
99.87	$\log_2 n$	104.91	$L^2(\mathbb{R}^2)$	102.86	$n(n-1)/2$	78.54	$O(n\log^2 n)$
99.65	$\xi(x)$	103.70	$\dot{x}=Ax+Bu$	101.40	$O(n^{-1})$	–	

C7	
98.72	$\mathrm{div}\left(|\nabla u|^{p-2}\nabla u\right)$

Table 2: Top $s(t,D)$ scores, where D is the set of all zbMATH documents with a minimum document frequency of 200, maximum document frequency of 500k, and a minimum complexity of 3.

lists reveal even more hits: $u_0(x)$ (rank 30), $-\Delta u$ (rank 32), and $u(x,t)$ (rank 33). Since all eight terms are among the most frequent 35 entries in zbMATH, it implies that u can most likely be considered to imply a function in zbMATH. Of course, this does not imply that u must always be a function in zbMATH (see $f(u)$ on rank 14 in C3), but this allows us to exploit probabilities for improving MathIR performance. For instance, if not stated otherwise, u could be interpreted as a function by default, which could help increase the precision of the aforementioned tools.

Figure 4 also demonstrates that our two datasets diverge for increasing complexities. Hence, we can assume that frequencies of less complex formulae are more topic-independent. Conversely, the more complex a math formula is, the more context-specific it is. In the following, we will further investigate this assumption by applying TF-IDF rankings on the distributions.

4 Relevance Ranking for Formulae

Zipf's law encourages the idea of scoring the relevance of words according to their number of occurrences in the corpus and in the documents. The family of BM25 ranking functions based on TF-IDF scores are still widely used in several retrieval systems [7, 20]. Since we demonstrated that mathematical formulae (and their subexpressions) obey Zipf's law in large scientific corpora, it appears intuitive to also use TF-IDF rankings, such as a variant of BM25, to calculate their relevance. In its original form [7], *Okapi BM25* was calculated as follows

$$\mathrm{bm25}(t,d) := \frac{(k+1)\,\mathrm{IDF}(t)\,\mathrm{TF}(t,d)}{\mathrm{TF}(t,d)+k\left(1-b+\frac{b|d|}{\mathrm{AVG}_{\mathrm{DL}}}\right)}, \qquad (7)$$

where $\mathrm{TF}(t,d)$ is the term frequency of t in the document d, $|d|$ the length of the document d (in our case, the number of subexpressions), $\mathrm{AVG}_{\mathrm{DL}}$ the average

length of the documents in the corpus (see Table 1), and IDF (t) is the inverse document frequency of t, defined as

$$\mathrm{IDF}(t) := \log \frac{N - n(t) + \frac{1}{2}}{n(t) + \frac{1}{2}}, \qquad (8)$$

where N is the number of documents in the corpus and $n(t)$ the number of documents which contain the term t. By adding $\frac{1}{2}$, we avoid $\log 0$ and division by 0. The parameters k and b are free, with b controlling the influence of the normalized document length and k controlling the influence of the term frequency on the final score. For our experiments, we chose the standard value $k = 1.2$ and a high impact factor of the normalized document length via $b = 0.95$.

As a result of our subexpression extraction algorithm, we generated a bias towards low complexities. Moreover, longer documents generally consist of more complex expressions. As demonstrated in Section 2.1, a document that only consists of the single expression $P_n^{(\alpha,\beta)}(x)$, i.e., the document had a length of one, would generate eight subexpressions, i.e., it results in a document length of eight. Thus, we modify the BM25 score in Equation (7) to emphasize higher complexities and longer documents. First, the average document length is divided by the average complexity AVG_C in the corpus that is used (see Table 1), and we calculate the reciprocal of the document length normalization to emphasize longer documents.

Moreover, in the scope of a single document, we want to emphasize expressions that do not appear frequently in this document, but are the most frequent among their level of complexity. Thus, less complex expressions are ranked more highly if the document overall is not very complex. To achieve this weighting, we normalize the term frequency of an expression t according to its complexity $c(t)$ and introduce an inverse term frequency according to all expressions in the document

$$\mathrm{ITF}(t, d) := \log \frac{|d| - \mathrm{TF}(t, d) + \frac{1}{2}}{\mathrm{TF}(t, d) + \frac{1}{2}}. \qquad (9)$$

Finally, we define the score $\mathrm{s}(t, d)$ of a term t in a document d as

$$s(t, d) := \frac{(k + 1)\,\mathrm{IDF}(t)\,\mathrm{ITF}(t, d)\,\mathrm{TF}(t, d)}{\max_{t' \in d|_{c(t)}} \mathrm{TF}(t', d) + k \left(1 - b + \frac{b\,\mathrm{AVG}_{DL}}{|d|\,\mathrm{AVG}_C}\right)}. \qquad (10)$$

The TF-IDF ranking functions and the introduced $\mathrm{s}(t, d)$ are used to retrieve relevant documents for a given search query. However, we want to retrieve relevant subexpressions over a set of documents. Thus, we define the score of a formula (mBM25) over a set of documents as the maximum score over all documents

$$\mathrm{mBM25}(t, D) := \max_{d \in D} \mathrm{s}(t, d), \qquad (11)$$

where D is a set of documents. We used *Apache Flink* [38] to count the expressions and process the calculations. Thus, our implemented system scales well for large corpora.
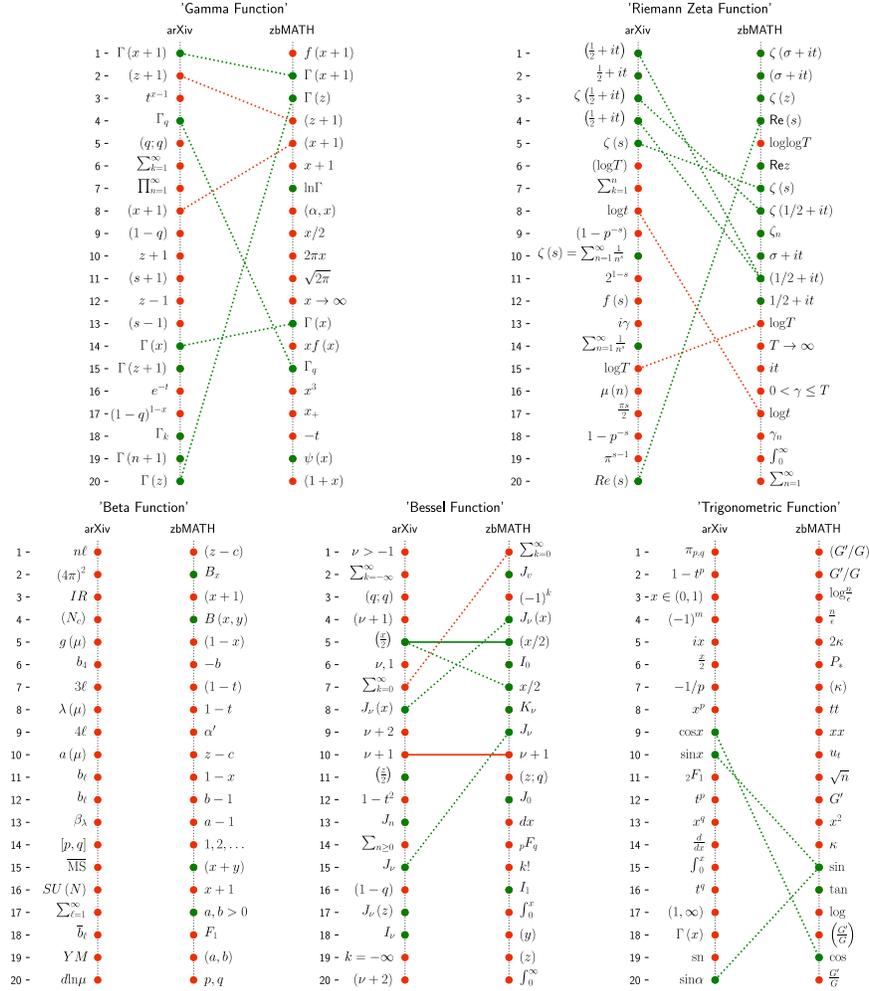
16

'Gamma Function'

arXiv	zbMATH
1 – $\Gamma(x+1)$	$f(x+1)$
2 – $(z+1)$	$\Gamma(x+1)$
3 – t^{x-1}	$\Gamma(z)$
4 – Γ_q	$(z+1)$
5 – $(q;q)$	$(x+1)$
6 – $\sum_{k=1}^{\infty}$	$x+1$
7 – $\prod_{n=1}^{\infty}$	$\ln\Gamma$
8 – $(x+1)$	(α,x)
9 – $(1-q)$	$x/2$
10 – $z+1$	$2\pi x$
11 – $(s+1)$	$\sqrt{2\pi}$
12 – $z-1$	$x\to\infty$
13 – $(s-1)$	$\Gamma(x)$
14 – $\Gamma(x)$	$xf(x)$
15 – $\Gamma(z+1)$	Γ_q
16 – e^{-t}	x^3
17 – $(1-q)^{1-x}$	x_+
18 – Γ_k	$-t$
19 – $\Gamma(n+1)$	$\psi(x)$
20 – $\Gamma(z)$	$(1+x)$

'Riemann Zeta Function'

arXiv	zbMATH
1 – $\left(\frac{1}{2}+it\right)$	$\zeta(\sigma+it)$
2 – $\frac{1}{2}+it$	$(\sigma+it)$
3 – $\zeta\left(\frac{1}{2}+it\right)$	$\zeta(z)$
4 – $\left(\frac{1}{2}+it\right)$	$\mathrm{Re}(s)$
5 – $\zeta(s)$	$\log\log T$
6 – $(\log T)$	$\mathrm{Re}z$
7 – $\sum_{k=1}^{n}$	$\zeta(s)$
8 – $\log t$	$\zeta(1/2+it)$
9 – $(1-p^{-s})$	ζ_n
10 – $\zeta(s)=\sum_{n=1}^{\infty}\frac{1}{n^s}$	$\sigma+it$
11 – 2^{1-s}	$(1/2+it)$
12 – $f(s)$	$1/2+it$
13 – $i\gamma$	$\log T$
14 – $\sum_{n=1}^{\infty}\frac{1}{n^s}$	$T\to\infty$
15 – $\log T$	it
16 – $\mu(n)$	$0<\gamma\le T$
17 – $\frac{\pi s}{2}$	$\log t$
18 – $1-p^{-s}$	γ_n
19 – π^{s-1}	$\int_0^{\infty}$
20 – $\mathrm{Re}(s)$	$\sum_{n=1}^{\infty}$

'Beta Function'

arXiv	zbMATH
1 – $n\ell$	$(z-c)$
2 – $(4\pi)^2$	B_x
3 – IR	$(x+1)$
4 – (N_c)	$B(x,y)$
5 – $g(\mu)$	$(1-x)$
6 – b_4	$-b$
7 – 3ℓ	$(1-t)$
8 – $\lambda(\mu)$	$1-t$
9 – 4ℓ	α'
10 – $a(\mu)$	$z-c$
11 – b_ℓ	$1-x$
12 – b_ℓ	$b-1$
13 – β_λ	$a-1$
14 – $[p,q]$	$1,2,\ldots$
15 – $\overline{\mathrm{MS}}$	$(x+y)$
16 – $SU(N)$	$x+1$
17 – $\sum_{\ell=1}^{\infty}$	$a,b>0$
18 – $\overline{b_\ell}$	F_1
19 – YM	(a,b)
20 – $d\ln\mu$	p,q

'Bessel Function'

arXiv	zbMATH
1 – $\nu>-1$	$\sum_{k=0}^{\infty}$
2 – $\sum_{k=-\infty}^{\infty}$	J_v
3 – $(q;q)$	$(-1)^k$
4 – $(\nu+1)$	$J_\nu(x)$
5 – $\left(\frac{x}{2}\right)$	$(x/2)$
6 – $\nu,1$	I_0
7 – $\sum_{k=0}^{\infty}$	$x/2$
8 – $J_\nu(x)$	K_ν
9 – $\nu+2$	J_ν
10 – $\nu+1$	$\nu+1$
11 – $\left(\frac{z}{2}\right)$	$(z;q)$
12 – $1-t^2$	J_0
13 – J_n	dx
14 – $\sum_{n\ge 0}$	$_pF_q$
15 – J_ν	$k!$
16 – $(1-q)$	I_1
17 – $J_\nu(z)$	$\int_0^x$
18 – I_ν	(y)
19 – $k=-\infty$	(z)
20 – $(\nu+2)$	$\int_0^{\infty}$

'Trigonometric Function'

arXiv	zbMATH
1 – $\pi_{p,q}$	(G'/G)
2 – $1-t^p$	G'/G
3 – $x\in(0,1)$	$\log\frac{n}{\epsilon}$
4 – $(-1)^m$	$\frac{n}{\epsilon}$
5 – ix	2κ
6 – $\frac{x}{2}$	P_s
7 – $-1/p$	(κ)
8 – x^p	tt
9 – $\cos x$	xx
10 – $\sin x$	u_t
11 – $_2F_1$	$\sqrt{n}$
12 – t^p	G'
13 – x^q	x^2
14 – $\frac{d}{dx}$	κ
15 – $\int_0^x$	$\sin$
16 – t^q	$\tan$
17 – $(1,\infty)$	$\log$
18 – $\Gamma(x)$	$\left(\frac{G'}{G}\right)$
19 – sn	$\cos$
20 – $\sin\alpha$	$\frac{G'}{G}$

Figure 5: Top-20 ranked expressions retrieved from a topic-specific subset of documents D_q. The search query q is given above the plots. Retrieved formulae are annotated by a domain expert with green dots for relevant and red dots for non-relevant hits. A line is drawn if a hit appears in both result sets. The line is colored in green when the hit was marked as relevant.

Table 2 shows the top-7 scored expressions, where D is the entire zbMATH dataset. The retrieved expressions can be considered as meaningful and real-world examples of MOIs, since most expressions are known for specific mathematical concepts, such as $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$, which refers to the Galois group of $\overline{\mathbb{Q}}$ over $\mathbb{Q}$, or $L^2(\mathbb{R}^2)$, which refers to the L^2-space (also known as *Lebesgue space*) over $\mathbb{R}^2$. However, a more topic-specific retrieval algorithm is desirable. To achieve this goal, we (i) retrieved a topic-specific subset of documents $D_q \subset D$ for a given textual search query q, and (ii) calculated the scores of all expressions in the retrieved documents. To generate D_q, we indexed the text sources of the documents from arXiv and zbMATH via elasticsearch (ES)[13] and performed the pre-processing steps: filtering stop words, stemming, and ASCII-folding[14]. Table 3 summarizes the settings we used to retrieve MOIs from a topic-specific subset of documents D_q. We also set a minimum hit frequency according to the number of retrieved documents an expression appears in. This requirement filters out uncommon notations.

	arXiv	zbMATH
Retrieved Doc.	40	200
Min. Hit Freq.	7	7
Min. DF	50	10
Max. DF	10k	10k

Table 3: Settings for the retrieval experiments.

Figure 5 shows the results for five search queries. We asked a domain expert from the National Institute of Standards and Technology (NIST) to annotate the results as related (shown as green dots in Figure 5) or non-related (red dots). We found that the results range from good performances (e.g., for the Riemann zeta function) to bad performances (e.g., beta function). For instance, the results for the Riemann zeta function are surprisingly accurate, since we could discover that parts of Riemann's hypothesis[15] were ranked highly throughout the results (e.g., $\zeta(\frac{1}{2} + it)$). On the other hand, for the beta function, we retrieved only a few related hits, of which only one had a strong connection to the beta function $B(x, y)$. We observed that the results were quite sensitive to the chosen settings (see Table 3). For instance, according to the beta function, the minimum hit frequency has a strong effect on the results, since many expressions are shared among multiple documents. For arXiv, the expressions $B(\alpha, \beta)$ and $B(x, y)$ only appear in one document of the retrieved 40. However, decreasing the minimum hit frequency would increase noise in the results.

Even though we asked a domain expert to annotate the results as relevant or not, there is still plenty of room for discussion. For instance, $(x + y)$ (rank 15

---

[13] https://github.com/elastic/elasticsearch [Accessed Sep. 2019]. We used version 7.0.0

[14] This means that non-ASCII characters are replaced by their ASCII counterparts or will be ignored if no such counterpart exists.

[15] Riemann proposed that the real part of every non-trivial zero of the Riemann zeta function is $1/2$. If this hypothesis is correct, all the non-trivial zeros lie on the critical line consisting of the complex numbers $1/2 + it$.

Riemann Zeta Function

C1		C2		C3		C4	
15,051	n	4,663	(s)	1,456	$\zeta(s)$	349	$(\frac{1}{2}+it)$
11,709	s	2,460	(x)	340	$\sigma+it$	232	$(1/2+it)$
9,768	x	2,163	(n)	310	$\sum_{n=1}^{\infty}$	195	$(\sigma+it)$
8,913	k	1,485	(t)	275	$(\log T)$	136	$\frac{1}{2}+it$
8,634	T	1,415	it	264	$1/2+it$	97	$s=\sigma+it$

C5		C6		TF-IDF	mBM25
203	$\zeta(\frac{1}{2}+it)$	105	$\lvert\zeta(1/2+it)\rvert$	$\zeta(s)$	$\zeta(1/2+it)$
166	$\zeta(1/2+it)$	88	$\lvert\zeta(\frac{1}{2}+it)\rvert$	$\zeta(1/2+it)$	$(1/2+it)$
124	$\zeta(\sigma+it)$	81	$\lvert\zeta(\sigma+it)\rvert$	$(1/2+it)$	$(\frac{1}{2}+it)$
54	$\zeta(1+it)$	32	$\lvert\zeta(1+it)\rvert$	$\frac{1}{2}+it$	$\zeta(\frac{1}{2}+it)$
44	$\zeta(2n+1)$	22	$\lvert\zeta(+it)\rvert$	$(\frac{1}{2}+it)$	$(\sigma+it)$

Eigenvalue

C1		C2		C3		C4	
45,488	n	12,515	(x)	686	$-\Delta u$	218	$\lvert\nabla u\rvert^{p-2}$
43,090	x	6,598	(t)	555	$(n-1)$	218	$-\Delta_p u$
37,434	λ	4,377	λ_1	521	$\lvert\nabla u\rvert$	133	$W_0^{1,p}(\Omega)$
35,302	u	2,787	(Ω)	512	a_{ij}	127	$\lvert\nabla u\rvert^2$
22,460	t	2,725	$\mathbb{R}^n$	495	$u(x)$	97	(a_{ij})

C5		C6		TF-IDF	mBM25
139	$\lvert\nabla u\rvert^{p-2}\nabla u$	137	$\left(\lvert\nabla u\rvert^{p-2}\nabla u\right)$	$Ax=\lambda Bx$	$-\operatorname{div}\left(\lvert\nabla u\rvert^{p-2}\nabla u\right)$
68	$-d^2/dx^2$	35	$-(py')'$	$-\Delta p$	$\operatorname{div}\left(\lvert\nabla u\rvert^{p-2}\nabla u\right)$
51	$A=(a_{ij})$	26	$(\lvert u'\rvert^{p-2}u')$	$P(\lambda)$	$p=\frac{N+2}{N-2}$
46	$-\frac{d^2}{dx^2}$	18	$(\phi_p(u'))'$	λ_{k+1}	$(\phi_p(u'))'$
45	$u\in W_0^{1,p}(\Omega)$	18	$\int_\Omega \lvert\nabla u\rvert^2\,dx$	$\lambda_1>0$	$\lambda\in(0,\lambda^*)$

Table 4: The top-5 frequent mathematical expressions in the result set of zb-MATH for the search queries 'Riemann Zeta Function' (top) and 'Eigenvalue' (bottom) grouped by their complexities (left) and the hits reordered according to their relevance scores (right). The TF-IDF score was calculated with normalized term frequencies.

Auto-completion for '$E = m$'			Suggestions for '$E = \{m, c\}$'		
Sug. Expression	TF	DF	Sug. Expression	TF	DF
$E = mc^2$	558	376	$E = mc^2$	558	376
$E = m\cosh\theta$	23	23	$E = \gamma mc^2$	39	38
$E = mv_0$	7	7	$E = \gamma m_e c^2$	41	36
$E = m/\sqrt{1-\dot{q}^2}$	12	6	$E = m\cosh\theta$	23	23
$E = m/\sqrt{1-\beta^2}$	10	6	$E = -mc^2$	35	17
$E = mc^2\gamma$	6	6	$E = \sqrt{m^2c^4 + p^2c^2}$	10	8

Table 5: Suggestions to complete '$E = m$' and '$E = \{m, c\}$' (the right-hand side contains m and c) with term and document frequency based on the distributions of formulae in arXiv.

in zbMATH, 'Beta Function') is the argument of the gamma function $\Gamma(x + y)$ that appears in the definition of the beta function [46, (5.12.1)] $B(x, y) := \Gamma(x)\Gamma(y)/\Gamma(x + y)$. However, this relation is weak at best, and thus might be considered as not related. Other examples are $\mathrm{Re}z$ and $\mathrm{Re}(s)$, which play a crucial role in the scenario of the Riemann hypothesis (all non-trivial zeroes have $\mathrm{Re}(s) = \frac{1}{2}$). Again, this connection is not obvious, and these expressions are often used in multiple scenarios. Thus, the domain expert did not mark the expressions as being related.

Considering the differences in the documents, it is promising to have observed a relatively high number of shared hits in the results. Further, we were able to retrieve some surprisingly good insights from the results, such as extracting the full definition of the Riemann zeta function [46, (25.2.1)] $\zeta(s) := \sum_{n=1}^{\infty} \frac{1}{n^s}$. Even though a high number of shared hits seem to substantiate the reliability of the system, there were several aspects that affected the outcome negatively, from the exact definition of the search queries to retrieve documents via ES, to the number of retrieved documents, the minimum hit frequency, and the parameters in mBM25.

5 Applications

The presented results are beneficial for a variety of use-cases. In the following, we will demonstrate and discuss several of the applications that we propose.

Extension of zbMATH's Search Engine: Formula search engines are often counterintuitive when compared to textual search, since the user must know how the system operates to enter a search query properly (e.g., does the system supports LaTeX inputs?). Additionally, mathematical concepts can be difficult to capture using only mathematical expressions. Consider, for example, someone who wants to search for mathematical expressions that are related to eigenvalues. A textual search query would only retrieve entire documents that require further investigation to find related expressions. A mathematical search engine, on the other hand, is impractical since it is not clear what would be a

fitting search query (e.g., $Av = \lambda v$?). Moreover, formula and textual search systems for scientific corpora are separated from each other. Thus, a textual search engine capable of retrieving mathematical formulae can be beneficial. Also, many search engines allow for narrowing down relevant hits by suggesting filters based on the retrieved results. This technique is known as faceted search. The zbMATH search engine also provides faceted search, e.g., by authors, or year. Adding facets for mathematical expressions allows users to narrow down the results more precisely to arrive at specific documents.

Our proposed system for extracting relevant expressions from scientific corpora via mBM25 scores can be used to search for formulae even with textual search queries, and to add more filters for faceted search implementations. Table 4 shows two examples of such an extension for zbMATH's search engine. Searching for 'Riemann Zeta Function' and 'Eigenvalue' retrieved 4,739 and 25,248 documents from zbMATH, respectively. Table 4 shows the most frequently used mathematical expressions in the set of retrieved documents. It also shows the reordered formulae according to a default TF-IDF score (with normalized term frequencies) and our proposed mBM25 score. The results can be used to add filters for faceted search, e.g., show only the documents which contain $u \in W_0^{1,p}(\Omega)$. Additionally, the search system now provides more intuitive textual inputs even for retrieving mathematical formulae. The retrieved formulae are also interesting by themselves, since they provide insightful information on the retrieved publications. As already explored with our custom document search system in Figure 5, the Riemann hypothesis is also prominent in these retrieved documents.

The differences between TF-IDF and mBM25 ranking illustrates the problem of an extensive evaluation of our system. From a broader perspective, the hit $Ax = \lambda Bx$ is highly correlated with the input query 'Eigenvalue'. On the other hand, the raw frequencies revealed a prominant role of $\text{div}(|\nabla u|^{p-2} \nabla u)$. Therefore, the top results of the mBM25 ranking can also be considered as relevant.

Math Notation Analysis: A faceted search system allows us to analyze mathematical notations in more detail. For instance, we can retrieve documents from a specific time period. This allows one to study the evolution of mathematical notation over time [1], or for identifying trends in specific fields. Also, we can analyze standard notations for specific authors since it is often assumed that authors prefer a specific notation style which may vary from the standard notation in a field.

Math Recommendation Systems: The frequency distributions of formulae can be used to realize effective math recommendation tasks, such as type hinting or error-corrections. These approaches require long training on large datasets, but may still generate meaningless results, such as $G_i = \{(x, y) \in \mathbb{R}^n : x_i = x_i\}$ [42]. We propose a simpler system which takes advantage of our frequency distributions. We retrieve entries from our result database, which contain all unique expressions and their frequencies. We implemented a simple prototype that retrieves the entries via pattern matching. Table 5 shows two examples. The left side of the table shows suggested autocompleted expressions for the
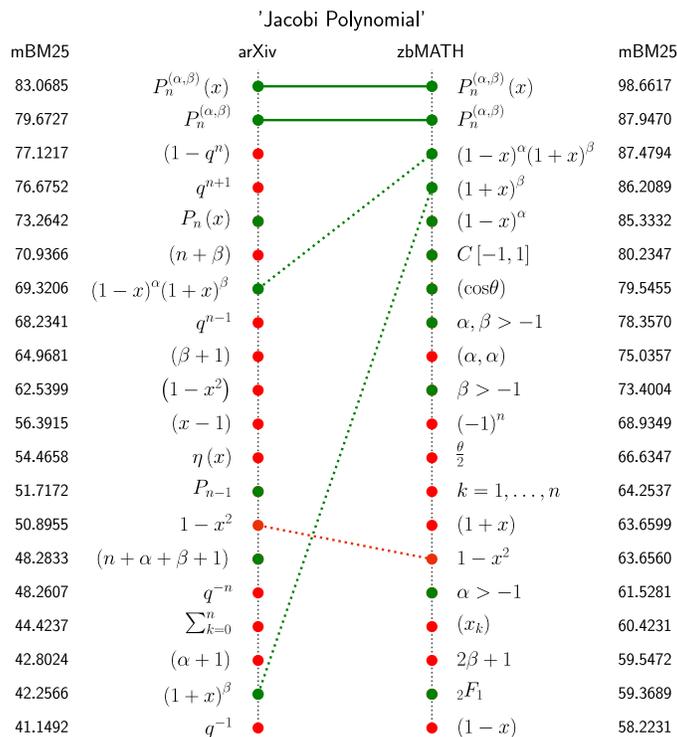
'Jacobi Polynomial'

mBM25	arXiv	zbMATH	mBM25
83.0685	$P_n^{(\alpha,\beta)}(x)$	$P_n^{(\alpha,\beta)}(x)$	98.6617
79.6727	$P_n^{(\alpha,\beta)}$	$P_n^{(\alpha,\beta)}$	87.9470
77.1217	$(1-q^n)$	$(1-x)^\alpha (1+x)^\beta$	87.4794
76.6752	q^{n+1}	$(1+x)^\beta$	86.2089
73.2642	$P_n(x)$	$(1-x)^\alpha$	85.3332
70.9366	$(n+\beta)$	$C[-1,1]$	80.2347
69.3206	$(1-x)^\alpha (1+x)^\beta$	$(\cos\theta)$	79.5455
68.2341	q^{n-1}	$\alpha,\beta > -1$	78.3570
64.9681	$(\beta+1)$	(α,α)	75.0357
62.5399	$(1-x^2)$	$\beta > -1$	73.4004
56.3915	$(x-1)$	$(-1)^n$	68.9349
54.4658	$\eta(x)$	$\frac{\theta}{2}$	66.6347
51.7172	P_{n-1}	$k=1,\ldots,n$	64.2537
50.8955	$1-x^2$	$(1+x)$	63.6599
48.2833	$(n+\alpha+\beta+1)$	$1-x^2$	63.6560
48.2607	q^{-n}	$\alpha > -1$	61.5281
44.4237	$\sum_{k=0}^n$	(x_k)	60.4231
42.8024	$(\alpha+1)$	$2\beta+1$	59.5472
42.2566	$(1+x)^\beta$	$_2F_1$	59.3689
41.1492	q^{-1}	$(1-x)$	58.2231

Figure 6: The top ranked expression for '*Jacobi polynomial*' in arXiv and zb-MATH. For arXiv, 30 documents were retrieved with a minimum hit frequency of 7.

query '$E = m$'. The right side shows suggestions for '$E =$', where the right-hand side of the equation should contain m and c in any order. A combination using more advanced retrieval techniques, such as similarity measures based on symbol layout trees [24, 25], would enlarge the number of suggestions. This kind of autocomplete and error-correction type-hinting system would be beneficial for various use-cases, e.g., in educational software or for search engines as a pre-processing step of the input.

Plagiarism Detection Systems: As previously mentioned, plagiarism detection systems [29, 39, 41] would benefit from a system capable of distinguishing conventional from uncommon notations. The approaches described by Meuschke et al. [39] outperform existing approaches by considering frequency distributions of single identifiers (expressions of complexity one). Considering that single identifiers make up only 0.03% of all unique expressions in arXiv, we presume that better performance can be achieved by considering more complex expressions. The conferred string representation also provides a simple format to embed complex expressions in existing learning algorithms.

Expressions with high complexities that are shared among multiple documents may provide further hints to investigate potential plagiarisms. For instance, the most complex expression that was shared among three documents in arXiv was Equation (3). A complex expression being identical in multiple documents could indicate a higher likelihood of plagiarism. Further investigation revealed that similar expressions, e.g., with infinite sums, are frequently used among a larger set of documents. Thus, the expression seems to be a part of a standard notation that is commonly shared, rather than a good candidate for plagiarism detection. Resulting from manual investigations, we could identify the equation as part of a concept called *generalized Hardy-Littlewood inequality* and Equation (3) appears in the three documents [12, 18, 17]. All three documents shared one author in common. Thus, this case also demonstrates a correlation between complex mathematical notations and authorship.

Semantic Taggers and Extraction Systems: We previously mentioned that semantic extraction systems [23, 28, 30] and semantic math taggers [16, 31] have difficulties in extracting the essential components (MOIs) from complex expressions. Considering the definition of the Jacobi polynomial in Equation (1), it would be beneficial to extract the groups of tokens that belong together, such as $P_n^{(\alpha,\beta)}(x)$ or $\Gamma(\alpha + m + 1)$. With our proposed search engine for retrieving MOIs, we are able to facilitate semantic extraction systems and semantic math taggers. Imagine such a system being capable of identifying the term 'Jacobi polynomial' from the textual context. Figure 6 shows the top relevant hits for the search query 'Jacobi polynomial' retrieved from zbMATH and arXiv. The results contain several relevant and related expressions, such as the constraints $\alpha, \beta > -1$ and the weight function for the Jacobi polynomial $(1 - x)^\alpha (1 + x)^\beta$, which are essential properties of this orthogonal polynomial. Based on these retrieved MOIs, the extraction systems can adjust its retrieved math elements to improve precision, and semantic taggers or a tokenizer could re-organize parse trees to more closely resemble expression trees.

6 Conclusion & Future Work

In this study we showed that analyzing the frequency distributions of mathematical expressions in large scientific datasets can provide useful insights for a variety of applications. We demonstrated the versatility of our results by implementing prototypes of a type-hinting system for math recommendations, an extension of zbMATH's search engine, and a mathematical retrieval system to search for topic-specific MOIs. Additionally, we discussed the potential impact and suitability in other applications, such as math search engines, plagiarism detection systems, and semantic extraction approaches. We are confident that this project lays a foundation for future research in the field of MathIR.

We plan on developing a web application which would provide easy access to our frequency distributions, the MOI search engine, and the type-hinting recommendation system. We hope that this will further expedite related future research projects. Moreover, we will use this web application for an online

evaluation of our MOI retrieval system. Since the level of agreement among annotators will be predictably low, an evaluation by a large community is desired.

In this first study, we preserved the core structure of the MATHML data which provided insightful information for the MATHML community. However, this makes it difficult to properly merge formulae. In future studies, we will normalize the MATHML data via MathMLCan [9]. In addition to this normalization, we will include wildcards for investigating distributions of formula patterns rather than exact expressions. This will allow us to study connections between math objects, e.g., between $\Gamma(z)$ and $\Gamma(x+1)$. This would further improve our recommendation system and would allow for the identification of regions for parameters and variables in complex expressions.

References

[1] Florian Cajori. *A History of Mathematical Notations.* Vol. 1 & 2. London, UK: The Open Court Company, 1929.

[2] Akiko N. Aizawa. An information-theoretic perspective of tf-idf measures. In: *Inf. Process. Manage.* (2003), pp. 45–65. DOI: `10.1016/S0306-4573(02)00021-3`.

[3] Glenn Gordon Smith and David Ferguson. Diagrams and math notation in e-learning: growing pains of a new generation. In: *International Journal of Mathematical Education in Science and Technology* (5 2004), pp. 681–695. DOI: `10.1080/0020739042000232583`.

[4] Ashish Lohia, Kirti Sinha, Soujanya Vadapalli, and Kamalakar Karlapalem. An Architecture for Searching and Indexing Latex Equations in Scientific Literature. In: *Proc. COMAD.* Goa, India: Computer Society of India, 2005, pp. 122–130.

[5] Alex Gaudeul. Do Open Source Developers Respond to Competition?: The LATEX Case Study. In: *Review of Network Economics* (2 June 2007), pp. 239–263. DOI: `10.2202/1446-9022.1119`.

[6] Christian Grün, Sebastian Gath, Alexander Holupirek, and Marc Scholl. XQuery Full Text Implementation in BaseX. In: *Database and XML Technologies.* Springer Berlin, 2009, pp. 114–128.

[7] Stephen E. Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. In: *Foundations and Trends in Information Retrieval* (2009), pp. 333–389. DOI: `10.1561/1500000019`.

[8] Shahab Kamali and Frank Wm. Tompa. A new mathematics retrieval system. In: *Proc. ACM CIKM.* Toronto, Ontario, Canada: ACM, 2010, pp. 1413–1416. DOI: `10.1145/1871437.1871635`.

[9]   David Formánek, Martin Líška, Michal Růžička, and Petr Sojka. Normal-
      ization of Digital Mathematics Library Content. In: *Proc. of OpenMath/
      MathUI/ CICM-WiP*. CEUR Workshop Proceedings. Bremen, Germany,
      2012, pp. 91–103.

[10] Michael Kohlhase, Bogdan A. Matican, and Corneliu-Claudiu Prodescu.
      MathWebSearch 0.5: Scaling an Open Formula Search Engine. In: *Intelli-
      gent 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. Proceed-
      ings*. Bremen, Germany: Springer Berlin Heidelberg, 2012, pp. 342–357.
      DOI: 10.1007/978-3-642-31374-5_23.

[11] Shahab Kamali and Frank Wm. Tompa. Retrieving documents with math-
      ematical content. In: *Proceedings of the 36th International ACM SIGIR
      Conference on Research and Development in Information Retrieval, SI-
      GIR '13, Dublin, Ireland - July 28 - August 01, 2013*. Dublin, Ireland:
      ACM, 2013, pp. 353–362. DOI: 10.1145/2484028.2484083.

[12] Gustavo Araujo and Daniel Pellegrino. On the constants of the Bohnenblust-
      Hille inequality and Hardy–Littlewood inequalities. In: *CoRR* (2014). arXiv:
      1407.7120.

[13] Giovanni Yoko Kristianto, Goran Topic, Florence Ho, and Akiko Aizawa.
      The MCAT Math Retrieval System for NTCIR-11 Math Track. In: *Proc.
      11th NTCIR Conference on Evaluation of Information Access Technolo-
      gies, National Center of Sciences,* Tokyo, Japan: National Institute of
      Informatics (NII), 2014.

[14] Aldo Lipani, Linda Andersson, Florina Piroi, Mihai Lupu, and Allan Han-
      bury. TUW-IMP at the NTCIR-11 Math-2. In: *Proceedings of the 11th
      NTCIR Conference on Evaluation of Information Access Technologies,
      NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12,
      2014*. Tokyo, Japan: National Institute of Informatics (NII), 2014.

[15] Steven T. Piantadosi. Zipf's word frequency law in natural language: A
      critical review and future directions. In: *Psychonomic Bulletin & Review*
      (Mar. 2014), pp. 1112–1130. DOI: 10.3758/s13423-014-0585-6.

[16] Pao-Yu Chien and Pu-Jen Cheng. Semantic Tagging of Mathematical Ex-
      pressions. In: *Proc. WWW'2015*. Florence, Italy: ACM, 2015, pp. 195–204.
      DOI: 10.1145/2736277.2741108.

[17] Daniel Pellegrino. A short communication on the constants of the multi-
      linear Hardy–Littlewood inequality. In: *CoRR* (2015). arXiv: 1510.00367.

[18] Jamilson R. Campos, Wasthenny Cavalcante, Vinícius V. Fávaro, Daniel
      Nuñez-Alarcón, Daniel Pellegrino, and Diana M. Serrano-Rodríguez. Poly-
      nomial and multilinear Hardy–Littlewood inequalities: analytical and nu-
      merical approaches. In: *CoRR* (2015). arXiv: 1503.00618.

[19] Leonard Wörteler, Michael Grossniklaus, Christian Grün, and Marc Scholl. Function inlining in XQuery 3.0 optimization. In: *Proc. 15th DBLP*. Pittsburgh, PA, USA: ACM, 2015, pp. 45–48. DOI: 10.1145/2815072.2815079.

[20] Jöran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: a literature survey. In: *Int. J. on Digital Libraries* (2016), pp. 305–338. DOI: 10.1007/s00799-015-0156-0.

[21] Ferruccio Guidi and Claudio Sacerdoti Coen. A Survey on Retrieval of Mathematical Knowledge. In: *Mathematics in Computer Science* (2016), pp. 409–427. DOI: 10.1007/s11786-016-0274-0.

[22] Shunsuke Ohashi, Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. Efficient Algorithm for Math Formula Semantic Search. In: *IEICE Transactions* (2016), pp. 979–988. DOI: 10.1587/transinf.2015DAP0023.

[23] Moritz Schubotz, Alexey Grigorev, Marcus Leich, Howard S. Cohl, Norman Meuschke, Bela Gipp, Abdou S. Youssef, and Volker Markl. Semantification of Identifiers in Mathematics for Better Math Information Retrieval. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. Full Paper. Pisa, Italy: ACM, 2016, pp. 135–144. DOI: 10.1145/2911451.2911503.

[24] Richard Zanibbi, Kenny Davila, Andrew Kane, and Frank Wm. Tompa. Multi-Stage Math Formula Search: Using Appearance-Based Similarity Metrics at Scale. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. Pisa, Italy: ACM, 2016, pp. 145–154. DOI: 10.1145/2911451.2911512.

[25] Kenny Davila and Richard Zanibbi. Layout and Semantics: Combining Representations for Mathematical Formula Search. In: *Proc. ACM SIGIR*. Shinjuku, Tokyo: ACM, 2017, pp. 1165–1168. DOI: 10.1145/3077136.3080748.

[26] Liangcai Gao, Zhuoren Jiang, Yue Yin, Ke Yuan, Zuoyu Yan, and Zhi Tang. Preliminary Exploration of Formula Embedding for Mathematical Information Retrieval: can mathematical formulae be embedded like a natural language? In: *CoRR* (2017). arXiv: 1707.05154.

[27] Andrea Kohlhase, Michael Kohlhase, and Michael Fürsich. Visual Structure in Mathematical Expressions. In: *Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings*. Lecture Notes in Computer Science. Edinburgh, UK: Springer, 2017, pp. 208–223. DOI: 10.1007/978-3-319-62075-6_15.

[28] Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. Utilizing dependency relationships between math expressions in math IR. In: *Information Retrieval Journal* (2017), pp. 132–167. DOI: 10.1007/s10791-017-9296-8.

[29] Norman Meuschke, Moritz Schubotz, Felix Hamborg, Tomás Skopal, and Bela Gipp. Analyzing Mathematical Content to Detect Academic Plagiarism. In: *Proc. ACM CIKM*. Singapore: ACM, 2017, pp. 2211–2214. DOI: `10.1145/3132847.3133144`.

[30] Moritz Schubotz, Leonard Krämer, Norman Meuschke, Felix Hamborg, and Bela Gipp. Evaluating and Improving the Extraction of Mathematical Identifier Definitions. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings*. Lecture Notes in Computer Science. Springer, 2017, pp. 82–94. DOI: `10.1007/978-3-319-65813-1_7`.

[31] Abdou Youssef. Part-of-Math Tagging and Applications. In: *Intelligent Computer Mathematics*. Cham: Springer International Publishing, 2017, pp. 356–374.

[32] Deyan Ginev. *arXMLiv:08.2018 dataset, an HTML5 conversion of arXiv.org*. SIGMathLing – Special Interest Group on Math Linguistics. 2018. URL: `https://sigmathling.kwarc.info/resources/arxmliv/`.

[33] Andrea Kohlhase. Factors for Reading Mathematical Expressions. In: *Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", LWDA 2018, Mannheim, Germany, August 22-24, 2018*. CEUR Workshop Proceedings. Mannheim, Germany: CEUR-WS.org, 2018, pp. 195–202.

[34] Andrea Kohlhase, Michael Kohlhase, and Taweechai Ouypornkochagorn. Discourse Phenomena in Mathematical Documents. In: *Intelligent Computer Mathematics - 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings*. Lecture Notes in Computer Science. Hagenberg, Austria: Springer, 2018, pp. 147–163. DOI: `10.1007/978-3-319-96812-4_14`.

[35] Kriste Krstovski and David M. Blei. Equation Embeddings. In: *CoRR* (2018). arXiv: `1803.09123`.

[36] Moritz Schubotz, André Greiner-Petter, Philipp Scharpf, Norman Meuschke, Howard S. Cohl, and Bela Gipp. Improving the Representation and Conversion of Mathematical Formulae by Considering their Textual Context. In: *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL 2018, Fort Worth, TX, USA, June 03-07, 2018*. Fort Worth, USA: ACM, 2018, pp. 233–242. DOI: `10.1145/3197026.3197058`.

[37] André Greiner-Petter, Terry Ruas, Moritz Schubotz, Akiko Aizawa, William I. Grosky, and Bela Gipp. Why Machines Cannot Learn Mathematics, Yet. In: *Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019) co-located with the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), Paris, France, July 25, 2019*. Paris, France: CEUR-WS.org, 2019, pp. 130–137.

[38] Fabian Hueske and Timo Walther. Apache Flink. In: *Encyclopedia of Big Data Technologies.* Springer, 2019. DOI: `10.1007/978-3-319-63962-8_303-1`.

[39] Norman Meuschke, Vincent Stange, Moritz Schubotz, Michael Kramer, and Bela Gipp. Improving Academic Plagiarism Detection for STEM Documents by Analyzing Mathematical Content and Citations. In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL).* Urbana-Champaign, USA, June 2019, pp. 120–129. DOI: `10.1109/JCDL.2019.00026`.

[40] Moritz Schubotz and Olaf Teschke. Four decades of TeX at zbMATH. In: *Newsletter of the European Mathematical Society (EMS)* (2019), pp. 50–52. DOI: `10.4171/NEWS/112/15`.

[41] Moritz Schubotz, Olaf Teschke, Vincent Stange, Norman Meuschke, and Bela Gipp. Forms of Plagiarism in Digital Mathematical Libraries. In: *Intelligent Computer Mathematics - 12th International Conference, CICM 2019, Prague, Czech Republic, July 8-12, 2019, Proceedings.* Lecture Notes in Computer Science. Prague, Czech Republic: Springer, 2019, pp. 258–274. DOI: `10.1007/978-3-030-23250-4_18`.

[42] Michihiro Yasunaga and John Lafferty. TopicEq: A Joint Topic and Mathematical Equation Model for Scientific Texts. In: *CoRR* (2019). arXiv: `1902.06034`.

[43] Abdou Youssef and Bruce R. Miller. Explorations into the Use of Word Embedding in Math Search and Math Semantics. In: *Intelligent Computer Mathematics - 12th International Conference, CICM 2019, Prague, Czech Republic, July 8-12, 2019, Proceedings.* Lecture Notes in Computer Science. Prague, Czech Republic: Springer, 2019, pp. 291–305. DOI: `10.1007/978-3-030-23250-4_20`.

[44] André Greiner-Petter, Moritz Schubotz, Fabian Müller, Corinna Breitinger, Howard S. Cohl, Akiko Aizawa, and Bela Gipp. Discovering Mathematical Objects of Interest - A Study of Mathematical Notations. In: *Proceedings of The Web Conference 2020 (WWW'20), April 20–24, 2020, Taipei, Taiwan.* DOI: `10.1145/3366423.3380218`.

[45] Bruce R. Miller. *LaTeXML A LaTeX to XML/HTML/MathML Converter.* `http://dlmf.nist.gov/LaTeXML/`. Accessed: 2019-09-01.

[46] *NIST Digital Library of Mathematical Functions.* `http://dlmf.nist.gov/`, Release 1.0.25 of 2019-12-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

Listing 2: Use the following `BibTeX` code to cite this article

```
 1  @inproceedings{GreinerPetter2020,
 2    author = {Greiner-Petter, Andr{\'{e}} and
 3      Schubotz, Moritz and
 4      M\"{u}ller, Fabian and
 5      Breitinger, Corinna and
 6      Cohl, Howard S. and
 7      Aizawa, Akiko and
 8      Gipp, Bela},
 9    booktitle = {Proceedings of The Web Conference 2020 (WWW'20), April
          20--24, 2020, Taipei, Taiwan},
10    doi = {10.1145/3366423.3380218},
11    title = {Discovering Mathematical Objects of Interest - A Study of
          Mathematical Notations},
12  }
```