

Augmenting Mathematical Formulae for More Effective Querying & Efficient Presentation

vorgelegt von
Diplom-Physiker
Moritz Schubotz
geb. in Offenbach am Main

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
– Dr. rer. nat. –
genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Odej Kao
Gutachter: Prof. Dr. Volker Markl
Gutachter: Prof. Abdou Youssef, PhD
Gutachter: Prof. James Pitman, PhD

Tag der wissenschaftlichen Aussprache: 31. März 2017

Berlin 2017

Abstract

Mathematical Information Retrieval (MIR) is a research area that focuses on the Information Need (IN) of the Science, Technology, Engineering and Mathematics (STEM) domain. Unlike traditional Information Retrieval (IR) research, that extracts information from textual data sources, MIR takes mathematical formulae into account as well.

This thesis makes three main contributions:

1. It analyses the strengths and weaknesses of current MIR systems and establishes a new MIR task for future evaluations;
2. Based on the analysis, it augments mathematical notation as a foundation for future MIR systems to better fit the IN from the STEM domain; and
3. It presents a solution on how large web publishers can efficiently present mathematics to satisfy the INs of each individual visitor.

With regard to evaluation of MIR systems, it analyses the first international MIR task and proposes the Math Wikipedia Task (WMC). In contrast to other tasks, which evaluate the overall performance of MIR systems based on an IN, that is described by a combination of textual keywords and formulae, WMC was designed to gain insights about the math-specific aspects of MIR systems. In addition to that, this thesis investigates how different factors of similarity measures for mathematical expressions influence the effectiveness of MIR results.

Based on the aforementioned evaluations, this thesis proposes to rethink the fundamentals of MIR systems. MIR systems should elevate the internal representation of mathematics and use a more semantic rather than syntactic representation for the retrieval algorithms. This approach simplifies MIR research by defining three orthogonal MIR research challenges: (1) Augmentation; (2) Querying; and (3) Efficient Execution. As augmentation target, this thesis proposes the concept of context-free formulae visualized by the idea of Formula Home Page (FHP). By visiting a FHP, a mathematically literate person can fully understand the formula semantics without

context or additional resources. As a first step towards unsupervised formula augmentation, this thesis introduces Mathematical Language Processing (MLP). MLP extracts knowledge about individual formulae from their surrounding text. To achieve that, it borrows concepts from Natural Language Processing (NLP) and adapts them to the specifics of mathematical language.

To finally satisfy the user's mathematical IN, formulae (i.e., data representing mathematical semantics) need to be presented to the user. Given the large variety of users and information systems, delivering math in a robust, scalable, fast and accessible way, was an open research problem. This thesis investigates different approaches to solve this problem and demonstrates the feasibility of a service-oriented multi-format approach which was implemented and is known as the Mathoid math rendering service. This implementation improves the math rendering for all Wikimedia sites including Wikipedia in production.

Kurzfassung

Die digitale Revolution hat die Informationsbeschaffung grundlegend verändert. Das Internet ist zum ersten Anlaufpunkt zur Befriedigung des täglichen Informationsbedarfs avanciert - sowohl im privaten, als auch im professionellen Leben. Dies gilt auch für die Disziplinen Mathematik, Ingenieurwissenschaften, Natur und Technik (MINT). Der hohe Anteil mathematischer Ausdrücke, die in MINT-Fächerin integraler Bestandteil der Schriftsprache sind, stellt eine besondere Herausforderung für Systeme wie Suchmaschinen und Literaturempfehlungsdienste dar. Mit dieser Thematik beschäftigt sich das Forschungsgebiet Mathematical Information Retrieval (MIR). Einige Probleme, wie beispielsweise die Disambiguierung, können durch Adaption korrespondierender Methoden aus der Computerlinguistik gelöst werden. Viele Aspekte erfordern jedoch auch vollständig neue Lösungen.

Die Anwendungsszenarien für bessere Verarbeitungs- und Analyseverfahren von Texten mit einem hohen Anteil mathematischer Notation sind vielfältig und reichen von der Literaturrecherche wissenschaftlicher Texte, über die Vermeidung von Plagiaten bis zur Verbesserung von Lernsoftware für die MINT-Fächer in Schulen und Universitäten.

Die vorliegende Dissertation leistet die folgenden Beiträge zur MIR-Forschung:

1. Analyse der Stärken und Schwächen bereits bestehender MIR-Systeme und Entwicklung eines standardisierten Evaluationssystems zur Quantifizierung der Effektivität von MIR- Systemen.
2. Erforschung von Verfahren zur automatischen, semantischen Anreicherung mathematischer Ausdrücke.
3. Entwicklung eines Lösungsvorschlags für die effiziente und skalierbare Darstellung mathematischer Inhalte.

Basierend auf der Analyse bereits bestehender MIR-Systeme, wird in dieser Arbeit eine Dreiteilung der MIR-Forschung vorgeschlagen: (1) Augmentierung; (2) Anfragengenerierung und (3) Effiziente Ausführung.

Es wird ein Evaluationsverfahren zur Quantifizierung der Effektivität von MIR-Systemen entwickelt, bestehend aus einem auf Wikipedia basierenden Testkorpus, einer Aufgabenliste und einem vollautomatischen Auswertungssystem der Messergebnisse. Im Gegensatz zu herkömmlichen Evaluationsverfahren, bei denen die Aufgaben aus Schlagwortenlisten bestehen, verwendet das hier vorgestellte Verfahren Formelmuster. Das Evaluationsverfahren war Teil des ersten offiziellen, internationalen Wettbewerbs für MIR-Systeme. Darüber hinaus wurde das Evaluationsverfahren auch außerhalb des Wettbewerbs zur Evaluation von MIR-Systemen verwendet und von anderen Wissenschaftlern weiterentwickelt.

In einem Prozess, der als "Mathematical Language Processing" (MLP) bezeichnet wird, werden mathematische Bezeichner durch Informationen aus dem umgebenden Text semantisch angereichert. In einem zweiten Schritt wird nicht nur der umgebende Text eines Bezeichners betrachtet, sondern die Gesamtheit der Texte aus ähnlichen Themengebieten analysiert, um die Bedeutungen einzelner Bezeichner zu identifizieren und die Effektivität der semantischen Anreicherung weiter zu verbessern.

In einem weiteren Schritt wird die Darstellung und Verarbeitung von mathematischen Formeln in Wikipedia grundlegend verbessert. Dazu werden die mathematischen Ausdrücke, die bis zu diesem Zeitpunkt in Bilddateien dargestellt wurden, in HTML5-Code umgewandelt. Dies ermöglicht eine schnellere und skalierbare Verarbeitung der mathematischen Inhalte in Wikipedia. Seit Mai 2016 wird dieses Verfahren weltweit auf allen Wikipediaseiten mit mathematischen Ausdrücken verwendet.

Acknowledgements

This thesis would not have been possible without the collaboration and support of numerous individuals and institutions. I am especially grateful to my doctoral advisor, Professor Volker Markl. Moreover, I gratefully acknowledge Professor Abdou Youssef for collaboration and advise in different projects and Professor James Pitman for his support for this thesis and the research field as a whole. I wish to thank Dr. Howard Cohl for the open exchange of ideas and our collaboration on the National Institute of Standards and Technology (NIST) Digital Repository of Mathematical Formulae (DRMF) project. Furthermore, I wish to thank Marcus Leich, Dr. Howard Cohl and Norman Meuschke for their valuable feedback and proofreading of the manuscript. I also wish to thank Professor Akiko Aizawa, Professor Michael Kohlhase and Professor Bela Gipp for fruitful discussions at different stages of my thesis. I also thank collaborating researchers for their input including Alan Sexton, Dr. Bruce Miller, Deyan Ginev, Juan Soto, Dr. Peter Krautzberger, Frédéric Wang and Professor Volker Sorge. Furthermore, I wish to thank my students Alexey Grigorev, Robert Pagel, David Veenhuis, André Greiner-Petter, Malte Schwarzer, Julian Hilbigson, Duc Linh Tran, Tobias Uhlich and Thanh Phuong Luu.

I especially acknowledge all the people, that I collaborated with during code development, program design, code review, testing and infrastructure setup, especially Gabriel Wicke, Dr. Marko Obrovac, Terry Chay, Matthew Flaschen, Goran Topic, Bryan Davis, Dr. Scott Ananin, Daniel Kinzler, Derk-Jan Hartmann, Ed Sanders, Andrew Otto, Andrew Bogott, Erik Moeller, James Forrester, Lydia Pintscher, Quim Gil, Raimond Spekking, Alexandros Kosiariis, Frédéric Wang, Dr. Bruce Miller and Deyan Ginev.

I also gratefully acknowledge the NIST in Gaithersburg for inviting me as a foreign guest researcher in 2014, 2015 and 2016 as well as the National Institute for Informatics in Tokyo for my research stay in 2014. Furthermore, I wish to thank the Wikimedia Foundation for their travel grants for my participation in the Wikimedia Hackathon 2016 and 2017 as well as their financial support during my research visit in 2013. I thank ACM and SIGIR first for their conference travel grants.

Furthermore, I thank my colleagues at Technische Universität Berlin, my family and friends for their support during my work on this thesis.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Definition	3
1.3	Outline of the Thesis	4
2	Foundations of Formula Data Management	7
2.1	Content Augmentation	9
2.1.1	Introducing Context-Free Formulae	10
2.1.2	Digitization	12
2.1.3	Layout Description	14
2.1.4	Structure Description	20
2.1.5	Entity Linkage	21
2.2	Content Querying	22
2.2.1	Excursus on Information Retrieval Tasks	22
2.2.2	Query Formulation	23
2.2.3	Advanced Indexes	24
2.2.4	Hit Ranking	24
2.2.5	Result Presentation	25
2.3	Efficient Execution	25

2.3.1	Foundations of Parallel Data Processing Systems	25
2.3.2	Approaches Adapted from Other Disciplines	26
2.3.3	Math-Specific Approaches	26
2.4	Conclusions	27
3	Augmentation Methods	29
3.1	Digital Repository of Mathematical Formulae	30
3.1.1	Implementation Goals	31
3.1.2	Seeding with Generic LaTeX Sources	33
3.1.3	KLS Seeding Project	36
3.1.4	Future Outlook	39
3.2	Mathematical Language Processing	40
3.2.1	Problem and Motivation	41
3.2.2	Preliminary Study	43
3.2.3	Methods	48
3.2.4	Evaluation	58
3.2.5	Results	60
3.2.6	Conclusion and Outlook	67
3.3	Augmenting Presentation of Mathematical Expressions	69
3.3.1	Introduction: Browsers are Becoming Smarter	69
3.3.2	Bringing MathML to Wikipedia	71
3.3.3	Making Math Accessible to MathML Disabled Browsers	72
3.3.4	A Global Distributed Service for Math Rendering	74
3.3.5	Performance Analysis	75
3.3.6	Conclusion, Outlook and Future Work	77
4	Evaluation	79

4.1	Developing a Test Suite for MIR Systems	80
4.1.1	Dataset and Feedback to Wikipedia	81
4.1.2	Topic Design	82
4.1.3	Evaluation Process	84
4.1.4	Participants and Evaluation Results	85
4.1.5	Conclusion	85
4.2	MIR Systems vs. a Human Brain	87
4.2.1	Methods	88
4.2.2	Results	91
4.2.3	Discussion	94
4.2.4	Conclusion and Outlook	96
4.3	Analysis of Similarity Measure Factors	96
4.3.1	Formula Similarity Search	97
4.3.2	Similarity Measure Factors	97
4.3.3	Evaluation	99
4.3.4	Conclusion and Outlook	102
5	Conclusion and Greater Impact	107
5.1	Summary	107
5.2	Impact and Prospectives	110
A	Applications and Case Studies	117
A.1	Combining Text and Formula Search	118
A.1.1	Excursus: Math Parsing in MediaWiki	118
A.1.2	Combining Math Expressions and Text	120
A.1.3	Experimental Evaluation	123
A.1.4	Conclusion	123

A.2 Mathosphere: No Index Mathematical Information Retrieval	123
A.2.1 System Description	125
A.2.2 Results	133
A.2.3 Conclusion	137
A.3 Mathoid: Formulae Processing for Wikipedia	138
A.3.1 Mathoid’s Improvements over texvc Rendering	139
A.3.2 Internals of the Mathoid Rendering	141
A.3.3 Measuring Mathoid’s Performance	142
A.3.4 Comparing Different Rendering Engines with Mathpipe	144
A.3.5 Image Comparison	144
A.3.6 Further Work	151
A.3.7 Conclusion	151
B Data Tables	153
List of Figures	166
List of Tables	168
Bibliography	169
Acronyms	195

Chapter 1

Introduction

1.1 Motivation

Let's start with a 'Gedankenexperiment' (experiment of thought): Assume there is a global state of knowledge ψ which is materialized by the set of publications at large. Let ψ_0 be the state of a world without any publications. Every publication \mathcal{P} is an incremental update of the previous state of knowledge $\psi_i \rightarrow \psi_{i+1}$. We call a publication **essential**, if $|\psi_{i+1}| > |\psi_i|$ with respect to the measure $|\cdot|$, and $|\psi_0| = 0$. The goals of the authors is to maximize the **novelty** $|\psi_{i+1}| - |\psi_i|$. Two publications $\mathcal{P}, \mathcal{P}'$ are **independent**, if their novelty is invariant under permutation, i.e., $|\mathcal{P}'\mathcal{P}\psi| - |\mathcal{P}\psi| = |\mathcal{P}\mathcal{P}'\psi| - |\mathcal{P}'\psi|$.

For sufficiently small i , it is safe to assume that the authors state of knowledge $\hat{\psi}_i$ is similar to ψ_i and therefore they are able to maximize the novelty. However, in Schwarzer, Schubotz, et al. [213], we estimate the number articles published in 2015 to be approximately 1.9 million using a regression model [35]. This means that the average waiting time for a state change of $\psi_i \rightarrow \psi_{i+1}$ is currently between 40 and 50 ms which is significantly less than the reaction time of the human brain to visual stimuli [230, 212]. Consequently, authors have to develop smarter ways to update $\hat{\psi}$. While in the future, artificial intelligence might provide sophisticated updating methods, the principal of selection pushdown has been the method of choice for scientists in the past. That means that a partition function

$$\sigma_{\mathcal{P}}(\mathcal{P}') = \begin{cases} 1 & \text{for } \mathcal{P}' \text{ independent of } \mathcal{P} \\ \mathcal{P}' & \text{otherwise,} \end{cases} \quad (1.1)$$

filters independent publications, so that $\sigma_{\mathcal{P}}(\mathcal{P}')$ can be used to update $\hat{\psi}$ rather than \mathcal{P}' . Note that the partition function would need to have a selectivity of about 1 in 5000 to raise the average waiting time for a state change of $\hat{\psi}$ to one day.

The ensemble averaged stationary current is obtained from the moment generating function (cf. Appendix B, Eq. (B1)) and evaluates to

$$\langle I_\infty \rangle = \frac{1}{\langle \tau_L \rangle + \langle \tau_R \rangle (\epsilon^2 \langle \tau_T^2 \rangle + 2) + \frac{1}{4} \langle \Gamma_R \rangle \langle \tau_T^2 \rangle}, \quad (38)$$

where we defined

$$\langle \tau_L \rangle \equiv \left\langle \frac{1}{\Gamma_L} \right\rangle, \quad \langle \tau_R \rangle \equiv \left\langle \frac{1}{\Gamma_R} \right\rangle, \quad \langle \tau_T^2 \rangle \equiv \left\langle \frac{1}{T_C^2} \right\rangle. \quad (39)$$

Some interesting observations can be made from the explicit expressions for $\langle I_\infty \rangle$. First, Eq. (38) reduces to the known result [11, 26, 28] in the clean, non-random limit for the parameters Γ_L , Γ_R , and T_C .

Second, we recognize that in the expression for $\langle I_\infty \rangle$, the random coupling to the right reservoir enters in the form of two independent averages, i.e., the mean values $\langle \Gamma_R \rangle$ and $\langle \tau_R \rangle$ of the right tunnel rate and its inverse, respectively. In other words, when fixing the averages Eq. (39) there is in general no coincidence between $\langle I_\infty \rangle$ and the corresponding

current I_∞ without random fluctuations in the parameters. For a random ensemble of right tunnel rates Γ_R , the ensemble averaged current $\langle I_\infty \rangle$ is always *smaller* than the corresponding clean result I_∞ ,

$$\frac{\langle I_\infty \rangle}{I_\infty} = \frac{1}{1 + \alpha \frac{\langle \tau_T^2 \rangle}{4 \langle \tau_R \rangle} I_\infty} < 1, \quad \alpha \equiv \langle \tau_R \rangle \langle \Gamma_R \rangle - 1. \quad (40)$$

Here, the parameter α depends on the probability distribution, but it is always non-negative owing to Jensen's inequality $g(\langle x \rangle) \leq \langle g(x) \rangle$ for convex functions $g(x)$ in the special case $g(x) = \frac{1}{x}$.

For example, if we consider a uniform probability density distribution

$$f_{\sigma, \langle \tau_R \rangle}(x) = \begin{cases} \frac{1}{\sigma} & -\frac{\sigma}{2} + \langle \tau_R \rangle \leq x \leq \frac{\sigma}{2} + \langle \tau_R \rangle \\ 0 & \text{else} \end{cases} \quad (41)$$

with average $\langle \tau_R \rangle$ and width σ for the inverse right tunnel rate $\tau_R \equiv 1/\Gamma_R$, we find $\langle \Gamma_R \rangle = \frac{1}{\sigma} \log \left(1 - \frac{2\sigma}{\sigma - 2\langle \tau_R \rangle} \right)$. For

Figure 1.1: Excerpt from a STEM publication [203]: Mathematical expressions (blue) are inaccessible to today's Information Retrieval (IR) Systems.

Unfortunately, in the real world σ_P can only be estimated. While simple selection predicates, like for instance buying certain journal subscriptions, were state of the art in the past, the application of Literature Recommender Systems (LRS) became recently more common. A literature survey [32] analyzed more than 200 publications on LRS and identified that the majority (55%) apply content-based (e.g., keywords, n -grams) document similarity measures. While keywords and n -gram based recommendations provide benefits for many scientists [190], research recommender systems miss a lot of the significant content for publications from the Science, Technology, Engineering and Mathematics (STEM) area. There, a significant portion of the knowledge is encoded in mathematical formulae alone (cf. Figure 1.1).

My personal motivation for making mathematical formulae accessible to LRS and other IR systems emerged during my work on my diploma thesis [201], where I was searching for a theorem that I could apply to prove equation (40) in Figure 1.1. While I was lucky and found ‘my’ theorem by chance, scientists should be able to retrieve relevant mathematics deterministically. To achieve that, this thesis presents approaches to augment mathematical formulae for machines, so that they can be effectively queried and presented as needed by scientists.

Especially, augmented mathematical content that is accessible to Mathematical Information Retrieval (MIR) systems, provides the basis for a wide range of future application driven research. For example, a well-designed math aware search engine for scientific publications helps to avoid reinventions. An example for a reinvention is the Viterbi algorithm [237] (involving dynamic programming equations), developed in 1967, which was rediscovered in 1970 by Needleman and Wunsch [170] (the

Needleman-Wunsch algorithm) for the alignment of protein or nucleotide sequences, and by Wagner and Fischer [241] in 1974 (the Wagner-Fischer algorithm) for computing the edit distance between two strings¹. In addition, the applications include the following topics:

- More effective literature research;
- Plagiarism prevention;
- Fertilization for international cooperation by a better identification of research in a similar scientific domain;
- Improvement of process quality in STEM research by better queryable digital mathematical libraries;
- Better tutoring systems by improved tools to support the student learning process;
- Patent search by incorporating mathematical formulae;
- Enterprise search for technology companies;
- Search engines for mathematics i.e., applicable theorem search;
- Definition lookup; and
- Application lookup; etc.

1.2 Problem Definition

The Formula Centric View Point The first workshop on Mathematical Knowledge Management (MKM) was held in 2001, but the field has a much longer history [43]. For example, Gottlob Frege expressed significant parts of mathematics using predicate logic in 1879 [69], a century before the relationship between computer science and mathematics was described by Knuth [104]. While there is much to say about MKM (see for [116, 189, 243]) and the newer term MIR [259], I will keep this section brief and focus on data management of context-free mathematical formulae.

MIR is an interdisciplinary domain including research questions in the fields of mathematics, humanities and informatics. While concepts like *wisdom*, *knowledge* and *information* [193] play an important role in the motivation of my research, I neither define nor use those as technical terms in this thesis. In particular, deriving the Information Need (IN) for mathematically literate users via user studies [108, 262] is not in the scope of this informatics thesis. Instead, I approximate the IN from the

¹Example provided by Abdou Youssef.

literature or assume a certain IN, if user studies are unavailable. Moreover, the area of computer algebra systems and (semi)-automated theorem proving system for fully formalized mathematics is not considered in this thesis.

Based on this distinction, I coin the term Mathematical Formula Data Management (FDM) which I subdivide into three mutually orthogonal research challenges:

Content Augmentation How to augment existing, potentially unstructured formula data, to make them processable by IR systems?

Content Querying How to translate fuzzy INs into deterministic queries and retrieval units?

Efficient Execution How to efficiently execute formula related data augmentation and querying tasks on potentially large datasets?

This research area provides a large variety of research questions, suitable to serve content for computer science research groups for many years, if not even decades. For this doctoral thesis the scope is limited to the following research objective:

Development of a mechanism to automatically augment mathematical formulae in large datasets to make querying more effective with regard to the IN of mathematically literate users and to make the presentation of mathematical formulae more efficient.

To achieve this research objective, I define the following research tasks:

1. Research about existing MIR technology and classify their contributions regarding the three FDM research challenges.
2. Develop a representation for semantic formulae that might serve as augmentation target.
3. Augment mathematical formulae with regard to the identified extraction targets.
4. Develop measures and benchmarks that allow for objective evaluation of MIR systems performance.
5. Identify key components in MIR systems and evaluate the impact of each individual building block.

1.3 Outline of the Thesis

Chapter 1 describes the general need for math aware IR systems, divides the complex problem of MIR into three distinct informatics research areas and defines the research

objective of this thesis. Moreover, fundamental assumptions are made, the contribution of this research objective to other research areas is described, and the research tasks to archive the research objective are mentioned.

Chapter 2 provides an overview about the field of MIR and classifies past and current approaches with regard to the three areas **Content Augmentation**, **Content Querying**, and **Efficient Execution**. Moreover, the classification schema to different degrees of formality for mathematical formulae is introduced and compared to other approaches.

Chapter 3 develops mechanisms for Content Augmentation. This includes the development of the Digital Repository of Mathematical Formulae (DRMF) project with the concept of context-free formulae materialized by Formula Home Page (FHP). Moreover, the chapter presents the idea of Mathematical Language Processing (MLP). Semantics regarding mathematical identifiers are extracted from the surrounding text and subsequently indicators for the existence of implicit namespaces in the mathematical language derived. The chapter develops a mechanism to present formulae more effectively. This leads to the development of the Mathoid software and its application by the Wikimedia foundation to provide math rendering to Wikipedia user's.

Chapter 4 contributes to the development of standardized test and evaluation environments for MIR systems in the context of international MIR tasks. This satisfies the need for an overview about existing MIR technologies and its performance as well as provides a basis for objective evaluations of current and future MIR systems. Moreover, it derives the main factors for formula similarity and discusses their independence and fundamental properties.

Chapter 5 summarizes the results and their impact to the international research community as well as the benefit for possible applications. The thesis concludes with a summary of future works and an overview about my ongoing research projects.

Appendix A describes applications and case studies of augmentation methods. This includes two implementations of MIR systems and the description of a new framework for a more efficient method to test new math rendering services.

I will use ‘we’ rather than ‘I’ in the subsequent chapters of this thesis, since I published and discussed my ideas with others including my advisors Professor Volker Markl, Professor Abdou Youssef, and Professor James Pitman, the hosts during research visits Professor Akiko Aizawa, Dr. Howard Cohl and Professor Bela Gipp, fellow researchers Marcus Leich, Norman Meuschke, Alan Sexton, and Juan Soto, and the students Alexey Grigoriev, Malte Schwarzer, David Veenhuis, Robert Pagel, Jimmy Li, André Greiner-Petter, Julian Hilbigson, Duc Linh Tran, Tobias Uhlich and Thanh Phuong Luu.

Chapter 2

Foundations of Mathematical Formula Data Management

This chapter introduces our classification schema for different degrees of mathematical formality. In addition, we classify past and current approaches with regard to the three research areas of Mathematical Formula Data Management (FDM), thereby addressing Research Task 1: ‘Research about existing Mathematical Information Retrieval (MIR) technology and classify their contributions regarding the three FDM research challenges.’ More specifically, we report on Content Augmentation in Section 2.1, on Content Querying in Section 2.2 and Efficient Execution in Section 2.3.

We included all publications from a MIR survey from 2015 [84] in our analysis. We extended our analysis with additional publications of recent research from the area of MIR. Furthermore, previous surveys and theses [43, 19, 134, 135] on this topic were considered.

Guidi and Sacerdoti Coen [84] (and later republished as [85]) clustered the papers with regard to the following six aspects

1. Purpose

- (a) Document retrieval [106, 262, 3, 12, 11, 256, 136, 166, 5, 259, 102, 136, 162, 246, 254, 253, 252, 255, 138, 136, 254, 254, 253]
- (b) Formula retrieval [176, 73, 20, 27, 28, 29, 86, 101, 188]
- (c) Document synthesis [31, 129, 128, 42, 42, 129, 127, 108, 262, 128, 156]

2. Encoding

- (a) Presentation [138, 136, 162, 173, 184]

(b) Content [27, 28, 29, 73, 87, 89, 117, 134, 138, 137, 136, 163, 162, 166, 174, 176, 188, 204, 247, 261]

(c) Semantics [40, 184, 122]

3. Techniques

(a) Segmentation [262, 7]

(b) Normalization [160, 67, 91, 160, 10, 67, 160, 160, 176, 10, 162, 162, 176, 215, 219, 257, 10, 160, 59, 73, 176, 195]

(c) Approximation [71, 160, 219, 89]

(d) Enrichment [20, 27, 28, 29, 40, 122, 246, 90]

(e) Query Reduction

(f) Full text search [3, 71, 184, 91, 122, 128, 138, 136, 156, 158, 160, 162, 162, 181, 166, 219, 254, 252, 255, 122, 231, 174]

(g) Substitution Trees [89, 117, 101, 129]

(h) SQL [18, 86, 27, 28, 29, 18, 87]

(i) XQuery [12, 11, 256, 42, 129]

4. Ranking [252, 261, 209, 219, 252, 103, 174, 247, 102, 3, 262]

5. Evaluation [103, 134, 6, 7, 71, 184, 87, 89, 117, 122, 125, 132, 137, 181, 195, 204, 209, 231, 122, 7, 181, 108, 262]

6. Availability [181]

In contrast to the approach of [85], we structured the analysis and presentation of previous work according to the specific contributions to the FDM challenges introduced in 1.2. This structure allows MIR researchers to find relevant research regarding individual FDM challenges. Especially systems papers, that typically present a math search engine and its evaluation, make it sometimes difficult to identify the research contributions to the underlying data management research questions. Since the FDM classification and the associated set of research questions was a helpful guide for the research carried out in this thesis, we assume that this classification is beneficial for the whole research community in MIR.

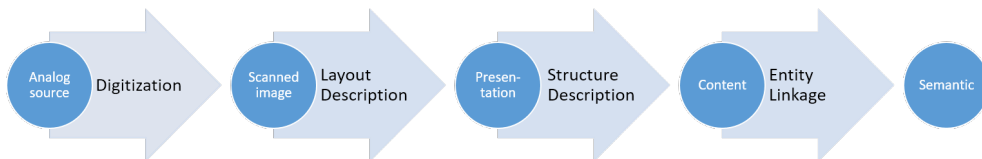


Figure 2.1: Overview of the Content Augmentation process

2.1 Content Augmentation

The first challenge, **Content Augmentation**, describes the process of collecting the full semantic information about an individual formula from a given input. Most fundamentally, this process starts with digitization of analogue mathematical content (Section 2.1.2) and the closely related task of mathematical Optical Character Recognition (OCR) (Section 2.1.3.2). It captures the conversion from imperative typesetting instructions to declarative layout descriptions (Section 2.1.3), but also deals with inferring the syntactical structure of a formula (Section 2.1.4). In addition, this first challenge involves the discovery of mathematical concepts (Section 2.1.5.1) and the association of formula metadata with individual formulae (Section 2.1.5.2).

As quoted in [97], there is a long-lasting interest in building a Digital Mathematics Library (DML). While collecting and indexing documents is an essential first step towards a DML, the vision reaches much farther. For instance, Schröder stated in 1898 that pasigraphy will be an important topic in future mathematics conferences:

Zur Diskussion auf einem internationalen Mathematiker-Kongress dürfte sich kaum ein Thema in höherem Maße empfehlen als das der Pasigraphie, Und ich bin überzeugt, daß der Gegenstand von der Tagesordnung künftiger Kongresse nicht mehr verschwinden wird. Ist doch das Ziel dieser neuen Disziplin kein geringeres als die endgültige Festlegung einer wissenschaftlichen Universalsprache, die völlig frei von nationalen Eigentümlichkeiten, und bestimmt ist durch ihre Konstruktion, die Grundlage zur wahren, nämlich exakten Philosophie zu liefern! [198]

Anticipating our efforts on context-free semantic Formula Home Page (FHP) (cf. Section 3.1) and the discovery of implicit namespaces in mathematical notation (cf. Section 3.2), the establishment of a universal mathematical notation is still an open problem that is been actively investigated. From a data perspective, Content Augmentation, the process of ascending the formality hierarchy, might be regarded as information preserving compression and entropy reduction process. For instance the Mathematical Markup Language standard has a much larger vocabulary for presentation markup compared to strict form content markup.

2.1.1 Introducing Context-Free Formulae

In Science, Technology, Engineering and Mathematics (STEM) research, mathematical expressions serve - like words - as constituents of sentences (cf. Figure 5.1). Neither ‘words’, nor ‘mathematical expressions’ carry semantics without each other. To extract semantics, words and mathematical expressions need to be considered at the same time. While there is much technology available, like for instance ‘word sense disambiguation’, to analyze words and specify their meaning, equivalent technology for mathematical formulae is less well-established. In the following, we describe how mathematical formulae can be isolated from their context. Therefore, we propose a threefold classification with regard to formality.

While there is a continuous spectrum of formality [113, 109], we identify three main levels of formality (cf. Figure 2.2) to facilitate the orientation in the spectrum of formality:

Presentation describes the arrangement of (mathematical) characters on a two-dimensional surface.

Content describes the interaction between objects, such as symbols, identifiers, numbers, etc. in the mathematical abstract syntax tree.

Semantic links items in the content tree to entities with properties and relations among each other that are described elsewhere.

Note that the content and semantics levels differ slightly from the definition of [84]. According to our terminology, the content level only represents the syntactic level corresponding to the abstract syntax tree in compilers. In this chapter, ‘libraries of formalized mathematical knowledge’ were not considered. Thus, their ‘semantic’ level has no corresponding level in our classification. However, from a data perspective, the association of elements in the content tree to a definition in a formally described logic, is only an attribute.

While we differentiate between different augmentation levels (cf. Section 2.1.1) and separate the tasks from each other, it has to be noted that in some cases the success of an individual augmentation target only becomes obvious on the next higher level. For example, a higher quality of a scan might have only little effect on a human subject while reading the scan, but significantly improves the OCR effectiveness. Moreover, we do not define explicit interfaces to pass data between the abstraction levels. For instance, a probability density function representing different possible results of an OCR result, might actually be better than just a single result. For performance benchmarks it has to be noted that systems, that process the content to a higher level of formality and then back-propagate the information to the lower level, might perform better in benchmarks, even though they might not contribute to a better technology for that particular augmentation task. One example of an approach that

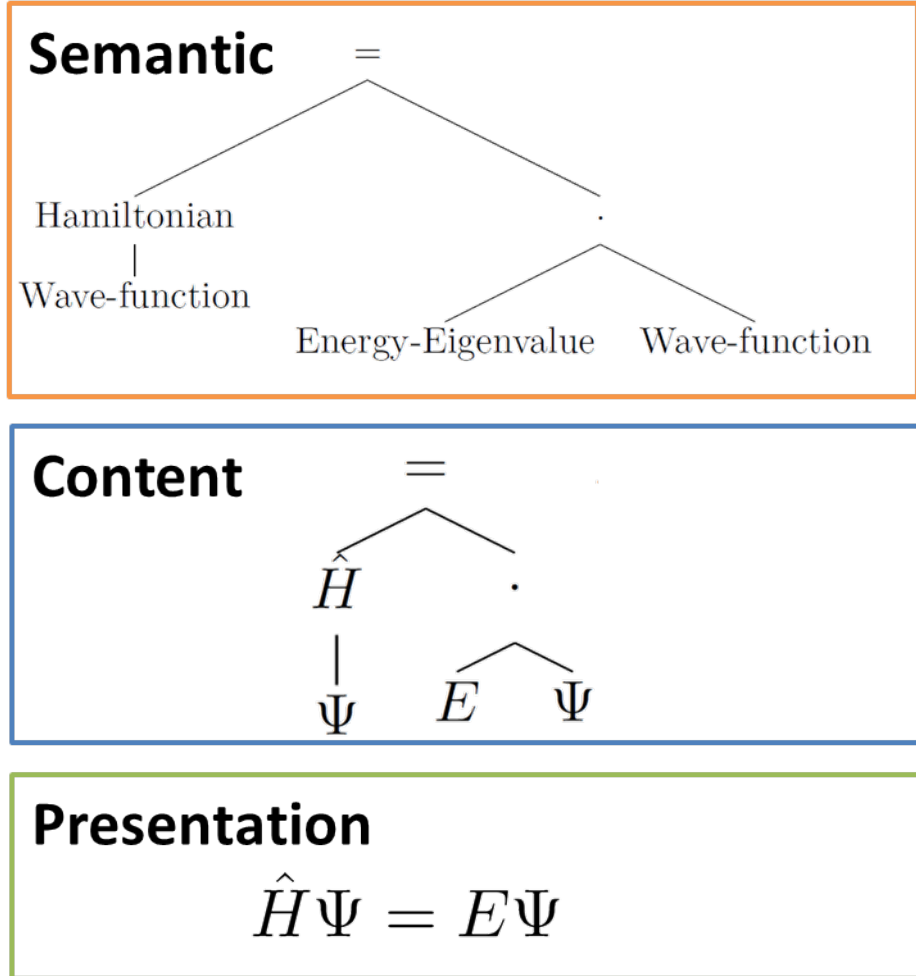


Figure 2.2: Different levels of formality for a mathematical formula visualized for the example of the time-independent Schrödinger equation. The lowest level (**Presentation**) describes the arrangement of (mathematical) characters on a two-dimensional surface supposed to be interpreted by a human reader. The middle level (**Content**) describes how the mathematical symbols interact with each other and form something like an abstract operator tree. The top level (**Semantic**) links to the entities, which are referred in the mathematical formulae. Note that the content tree visualization is algorithmically generated from the prefix-based Content MathML markup. However, it omits the invisible function application and uses the first child element as root node. For example, $\mathcal{Q}(\cdot, E, \Psi)$ is visualized as $\cdot(E, \Psi)$.

uses back-propagation was proposed by Ahmadi and Youssef [5], who compensate symbol-recognition errors by lexical rules.

2.1.2 Digitization

Digital libraries collect digital objects such as scientific publications and normally store those objects together with metadata describing the objects. While in most research areas keyword-based Information Retrieval (IR) leads to satisfactory results, scientific publications from STEM areas require additional and specific strategies due to the high percentage of mathematical notations. An average of 380 mathematical formulae per arXiv paper [138], reflects the significance of mathematical formulae for STEM publication and especially mathematical publications. While a general introduction to digital libraries can be found in [96], we focus on Digital Mathematical Libraries. In 2014, the US National Research Council published the vision of a global DML [52, 185]. The need for digital mathematical library was also recognized in Europe, where the <https://initiative.eudml.org/> project was launched [227]. Modern Digital Mathematics Libraries include mathematical formulae as well as metadata like abstracts, key words, mathematical subject classification (MSC) codes and links [67, 37].

The Digital Library of Mathematical Functions (DLMF) project revises and extends the ‘Handbook of Mathematical Functions’ [2] in a handbook and a website [158, 139]. In Section 3.1, we provide further details on the DLMF project and introduce the Digital Repository of Mathematical Formulae (DRMF) which was built based on the DLMF expertise.

In the field of digitization of mathematics there are two sub challenges:

1. Retrodigitization of mathematical content; and
2. (Online) recognition of hand written mathematics

There has been a large effort on **retrodigitization of mathematical content** [151] as summarized in [99]. According to Ulf Rehmanns list¹, which was featured as the most comprehensive list of retrodigitized mathematical contents [52], there are about 5M pages of so called ‘non-digitally born’ mathematics. However, even this area, which appears to be comparably less challenging, is not completed. Furthermore, the quality of the scans is not always sufficient for math OCR. Unlike black and white scans, that provide a good basis for OCR of Latin based natural language texts (cf. Figure 2.3), ongoing research from National Institute of Standards and Technology (NIST), based on the BMP first dataset (cf. Section 3.1.2), indicates that the recognition performance (w.r.t precision and recall) for mathematical notation

¹https://www.math.uni-bielefeld.de/~rehmann/DML/dml_links.html, accessed December 22, 2016

1.5. The beta function

The beta function is defined by the integral

$$(1) \quad B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt \quad \operatorname{Re} x > 0, \quad \operatorname{Re} y > 0.$$

Substituting $t = v/(1+v)$, the relation

$$(2) \quad B(x, y) = \int_0^\infty v^{x-1} (1+v)^{-x-y} dv \quad \operatorname{Re} x > 0, \quad \operatorname{Re} y > 0$$

is obtained, and from this (break into $\int_0^1 + \int_1^\infty$ and let $v = \frac{1}{t}$ for latter)

$$(3) \quad B(x, y) = \int_0^1 (v^{x-1} + v^{y-1}) (1+v)^{-x-y} dv \quad \operatorname{Re} x > 0, \quad \operatorname{Re} y > 0.$$

can be deduced. It follows that

$$(4) \quad B(x, y) = B(y, x).$$

Figure 2.3: Excerpt of a scan of the book ‘Higher Transcendental Functions’ by Erdélyi, Magnus, et al. [65], which is by permission of the copyright owner, publicly available.

increases, if gray scale scans are used. Moreover, it turned out that the recognition performance increases, if different copies of the same book are used. Despite technical challenges, economic interests hinder the dissemination of those sources [37].

Regarding recognition of hand written mathematics, Zanibbi and Blostein provided an extensive review of the state of the art in 2012 [259]. Note that the digitization of a time series of pen strokes or similar human input devices slightly differs from the retrodigitization part. In particular, Hidden Markov Models have been successfully applied to those time series [259]. Phan, Nguyen, et al. developed an incremental method for digitization which is more efficient with regard to a reduction of the waiting time for the user [183]. However, despite the large number of publications that included the digitization of hand written mathematical expression [259], few information can be found on the hardware and software setup used for the digitization.

To summarize the digitization section, we conclude that much research has been done on digitization of documents in general. However, we could not find related work that addresses math specific issues of retrodigitization such as

- Which scan setup (light, resolution, color, number of iterations) is best for documents containing mathematical formulae?
- Which hardware performs best for digitalizing mathematical documents?
- What are the minimal requirements (resolution, data formats, available meta-data) for storing digitalized version of mathematical documents in order to perform subsequent augmentation and retrieval tasks?

The same holds for recognition of handwritten mathematics. Especially online recognition systems that digitize mathematical content produced on physical devices such as chalk blackboards, electronic blackboards, touch screens or even on paper, provide a ground for many open research questions. The digitization of text with a high percentage of mathematical formulae with online recognition systems is more complex than the standard digitization task, but offers several additional opportunities as well. Beside considering the user feedback, analysis of the time aspect might improve digitization and more specifically the OCR disambiguation process. For example, we hypothesize that mathematicians tend to stop for a little while after writing an \int -sign, but do not pause after writing the Latin letter S. This information could be used to disambiguate between the visually similar \int -sign and S. Moreover, having recorded the audio track in addition to the video of a math lecture might help to disambiguate between similar characters such as the typically co-occurring identifiers u , v and ν . With regard to user feedback, systems like E-Chalk [228] from 2005 that are being used in production at universities for many years, provide Math OCR and might even be connected to a computer algebra system. However, they do not yet support the users while writing mathematics. However, this would be a major contribution to the research task of online digitization of mathematical content. Furthermore, interactive math digitization methods represent an enormous potential, since they allow capturing additional information from the human, which is only available at the moment of usage, as well as a flexible assistance of the user. Most recently, [229] proposed a box-based system to enter mathematical expressions, which allows a flexible adaption of size and arrangement of the constituents of a formula due to complexity. Their user study indicates that their method might be the preferred way of entering mathematics on a digital device in the future. For example, if Fermat would have had access to the boxed-based input system of Taranta, Vargas, et al., the proof of his ‘Last Theorem’ might have been available more than 350 years earlier.

2.1.3 Layout Description

After the digitization step, all successive steps are pure data management tasks and do neither involve physical devices nor human beings. Note, that there is some influence from the efficient execution challenge described in Section 2.3, where the physical characteristics of the hardware, used to perform the operations on the data, play an essential role.

Describing the layout has several disjoint challenges:

- Differentiate between mathematical expressions and other elements such as text images or tokens.
- Describe math-specific layout of documents.
- Identify individual atomic mathematical characters.

- Describe the spatial relations between the characters.

Blostein and Zanibbi [33] describe the final outcome of the layout analysis, the ‘Symbol Layout Tree’, which they claim contains the same information as Presentation MathML or LaTeX. While the name intuitively describes the goal of the layout description on an abstract level, without relying on one particular data encoding such as Presentation MathML or LaTeX, the term ‘symbol’ does not fit to the terminology used in this thesis. According to the MathML standard, a (Content) symbol ‘is used to refer to a specific, mathematically-defined concept with an external definition’. However, the ‘Symbol Layout Tree’ describes the spatial arrangement of printable mathematical characters, also referred as mathematical alphanumeric symbols by the MathML standard (cf. Section 2.1.3.1). To avoid confusion between content symbol and mathematical alphanumeric symbol, we refer to the ‘Character Layout Tree’ as the augmentation target of this subsection.

Moreover, we recognize differences between the LaTeX and the Presentation MathML encoding from a data management perspective. In contrast to the declarative description of mathematical expressions in the XML schema Presentation MathML, LaTeX is a Turing complete programming language to draw curves on paper. The repetitive execution of the same LaTeXcode, given a constant hard- and software setup, guarantees to produce the same visual impression. The declarative description using Presentation MathML, which eliminates side effects and separates the specification of the desired result from the actual implementation, is preferable. Following this paradigm, we have devoted Section 2.1.3.3 to the conversion to a declarative data format.

Before describing the related work with regards to the aforementioned challenges, we will provide a brief introduction to data formats for mathematical expressions.

2.1.3.1 Excursus on Data Formats for Mathematical Formulae

Key ingredients of digital (mathematical) libraries are the data formats used to store and manage mathematical expression. In this subsection, we will summarize the significant key features of data formats for this thesis. In the following, we will (1) introduce the standard markup language for mathematics MathML; (2) describe its Presentation and Content part; (3) introduce OpenMath as well as the concept of Content Dictionaries.

Mathematical Markup Language (MathML) Mathematical Markup Language (MathML) is a Extensible Markup Language (XML)-based language for describing mathematical notations and part of HTML5. It was published as World Wide Web Consortium (W3C) Recommendation in 1998 for the first time [159]. Furthermore, MathML is an ISO standard (ISO/IEC DIS 40314) since 2015. It contains two vocab-

ularies, namely **Presentation MathML** and **Content MathML**. While Presentation MathML focuses on the layout of mathematical notation, Content MathML deals with the semantics or meaning of the mathematical notation. These two vocabularies are preserved separately and can either be used alone or in conjunction. Presentation MathML requires a tag for every element (token) of a mathematical notation that specifies the role of the tokens in the mathematical expression (e.g., `<mi/>` for identifier or `<mo/>` for operators) [159]. Content MathML similarly requires a markup (e.g., `<ci/>` for identifiers). In contrast, operators in Content MathML are represented by elements from a core set that specify their function. These application of an operator is represented by `<apply/>`. The following listings show the MathML encoding of the mathematical expression $f(a+b)$ in Presentation MathML (Listing 2.1.) and Content MathML (Listing 2.2) in contrast to \TeX (Listing 2.3).

```

1 <math xmlns="http://www.w3.org/1998/Math/MathML">
2   <semantics>
3     <mrow id="r1">
4       <mi id="i1">f</mi>
5       <mo id="o1">(</mo>
6       <mrow id="r2">
7         <mi id="i2">a</mi>
8         <mo id="o2">+</mo>
9         <mi id="i2">b</mi>
10      </mrow>
11     <mo id="o3">)</mi>
12   </mrow>

```

Listing 2.1: MathML encoding of the mathematical expression $f(a+b)$. Part 1: Presentation MathML

```

13 <annotation-xml encoding="MathML-Content">
14   <apply xref="r1">
15     <ci xref="b">f</ci>
16     <apply xref="r2">
17       <plus xref="o2"/><!-- <csymbol cd="arith1">plus</csymbol> in
18           strict encoding -->
19       <ci xref="i2">a</ci>
20       <ci xref="i3">b</ci>
21     </apply>
22   </apply>

```

Listing 2.2: MathML encoding of the mathematical expression $f(a+b)$. Part 2: Content MathML

```

24 <annotation encoding="application/x-tex">f(a+b)</annotation>

```

Listing 2.3: MathML encoding of the mathematical expression $f(a+b)$. Part 3: LaTeX

OpenMath and Strict Content MathML MathML is tightly coupled to OpenMath². OpenMath provides the opportunity to encode the semantics of mathemat-

²<http://www.openmath.org/>

ical expressions via extensible content dictionaries. In contrast to MathML, which provides a presentation form (Presentation MathML), there is no presentation form for OpenMath encodings. Nevertheless, OpenMath encoding can be transformed into Presentation MathML and $\text{T}_{\text{E}}\text{X}$. Listing 2.4 show the mathematical expression $f(a+b)$ in OpenMath XML encoding.

```

25 <annotation-xml encoding="application/openmath+xml">
26   <OMOBJ xmlns="http://www.openmath.org/OpenMath">
27     <OMA>
28       <OMV name="f" />
29       <OMA>
30         <OMS cd="arith1" name="plus" />
31         <OMV name="a" />
32         <OMV name="b" />
33       </OMA>
34     </OMA>
35   </OMOBJ>
36 </annotation-xml>
37 </semantics>
38 </math>

```

Listing 2.4: MathML encoding of the mathematical expression $f(a+b)$. Part 4: OpenMath

Content MathML elements point to individual entries in the content dictionaries. Traditional Content MathML, has a predefined list of mathematical operators such as `<plus />`. In contrast, **strict** Content MathML requires that each mathematical symbol links to a content dictionary entry. For example, the MathML standard defines the symbol ‘plus’ in the content dictionary ‘arith1’ to be the strict equivalent of the traditional `<plus/>` tag.

Example 1: plus

Content dictionary entry ‘**arith1**’: **plus**: The symbol representing an n-ary commutative function plus.

and requires that the symbol has the ‘formal mathematical property’ $\forall a, b : a + b = b + a$. This formal property is also available in Content MathML.

```

1 <math xmlns="http://www.w3.org/1998/Math/MathML">
2   <bind><csymbol cd="quant1">forall</csymbol>
3     <bvar><ci>a</ci></bvar>
4     <bvar><ci>b</ci></bvar>
5     <apply>
6       <csymbol cd="relation1">eq</csymbol>
7       <apply>
8         <csymbol cd="arith1">plus</csymbol>
9         <ci>a</ci>
10        <ci>b</ci></apply>
11      </apply>
12      <csymbol cd="arith1">plus</csymbol>
13      <ci>b</ci>
14      <ci>a</ci></apply></apply></bind></math>

```

Listing 2.5: MathML encoding of the commutativity of ‘plus’ $\forall a, b : a + b = b + a$. from the ‘arith1’ content dictionary <http://www.openmath.org/cd/arith1.xhtml#plus>.

In contrast to strict Content MathML, canonical Content MathML has a default corresponding Presentation MathML representation. In our example (Listing 2.1 and Listing 2.2) the presentation and content elements are linked using pointers. Further details on OpenMath are available in the respective publications [44, 51, 60, 186, 216, 224, 232, 55].

2.1.3.2 Mathematical OCR

OCR is the conversion of images of text, either typed, handwritten or printed, into machine-encoded text. Mathematical OCR is the conversion of text including mathematical expression into a machine-readable format. Zanibbi and Blostein [259] provide an extensive overview of the state of the art Math OCR. They elaborate on the relation between MIR and the structure detection in STEM documents. Since 2012, several approaches have been developed [167, 21, 225, 70, 131]. Most notably are the ongoing efforts of the team of scientists and engineers that have developed the commercial product Infty [226]. To our knowledge, Infty is the only commercial product on the market. While their original main focus was to identify characters in retrodigitized material, their software is applied to digitally born material with increasing frequency. Baker and others [24, 23, 95, 22, 26, 130, 9, 131, 25] developed a promising approach for OCR in PDF documents, MaxTract. Nevertheless, their open source system was never turned from a research prototype to a commercial product. We were able to reproduce the high precision and recall values of MaxTract that suggest a significant improvement compared to the performance of Infty. However, MaxTract requires a certain type of PDF documents, i.e., those using Type 1 fonts.

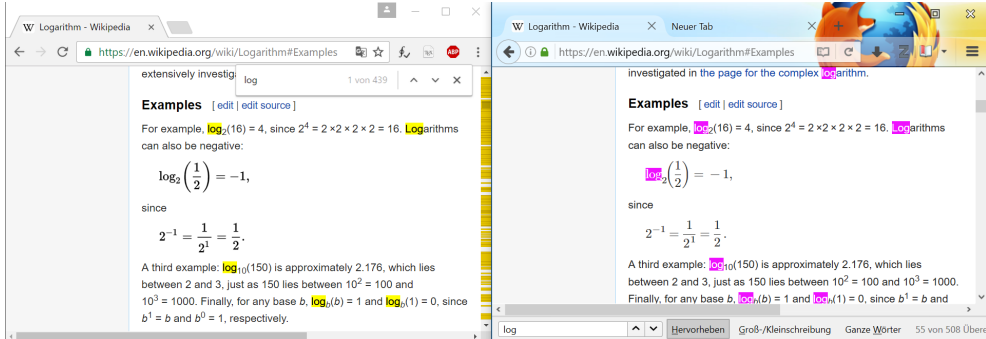


Figure 2.4: Highlighting ‘log’ in Chrome using Scalable Vector Graphics (SVG) (left) vs. Firefox using MathML (right). Highlighting in formulae displayed as SVG (like in the example $\log_2\left(\frac{1}{2}\right) = -1$) is currently not possible.

2.1.3.3 Declarative Layout Description

Many mathematical documents use T_EX or L^AT_EX which are turing complete programming language. Moreover, many mathematical expressions are available from PDF documents or SVG. While all the aforementioned options are capable to produce a visually perfect display of mathematical formulae on a two-dimensional, typically plane surface (such as a screen or a piece of paper), they do not provide descriptive information on the displayed content. For instance, for visually impaired people this information is of little help. Furthermore, these types of content are hard to access for information retrieval systems (for instance image search on web sites, cf. Figure 2.4). Furthermore, the reuse or a remix of mathematical expressions is also difficult, if no declarative layout description is available. Especially exchanging and modifying L^AT_EX involves some difficulties that should not be underestimated, such as custom macros, special packages, redefinition of syntax. The most prominent examples of obfuscated L^AT_EX code are the ctan-L^AT_EX-packages xii which encodes a Christmas Song with 762 character, and reverxii which implements a L^AT_EX Reversi game in 938 characters. These examples show that the reimplementations of L^AT_EX parsers to perform IR is non-trivial.

As not all browsers support MathML, MathJax converts mathematical notation from several input formats (including MathML, T_EX and L^AT_EX) to either MathML or non-standard HTML with special CSS rules that make mathematical notation look similar to the L^AT_EX rendering, but does not require a browser that supports HTML5, i.e., MathML [45]. The internal format for MathJax is essentially MathML. Beside MathJax [45], there are several tools for converting L^AT_EX to MathML like LaTeXML [155], Tralics [83] or SnuggleTex [146]. Beside other L^AT_EX to MathML converters, LaTeXML [155] has proven to successfully convert large portions of the arXiv papers [222]. A more detailed investigation of LaTeXML and our new development Mathoid [207] is described in Section A.3. An alternative to the semantic augmentation

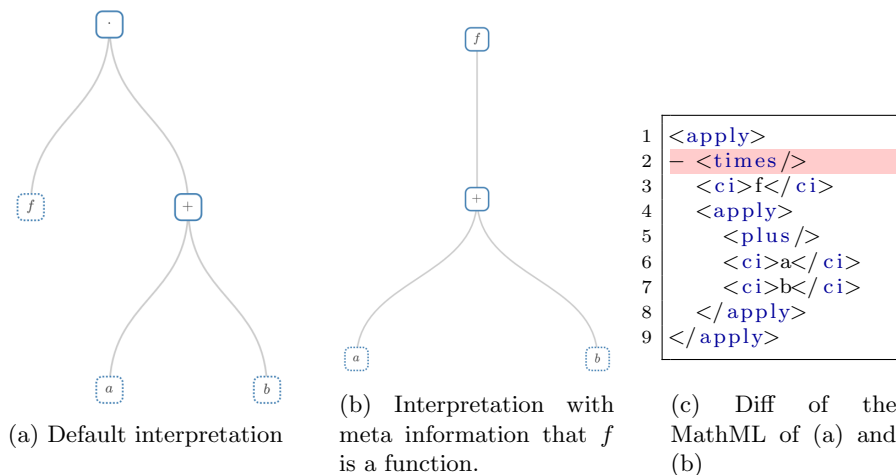


Figure 2.5: Visualizations of different possible syntactical structures of the expression $f(a+b)$. The visualizations were generated from the LaTeXsource (a) $\$f(a+b)\$$ and (b) $\backslash\text{Declare[role=FUNCTION]}\{f\}\ \$f(a+b)\$$ respectively. The LaTeXsource was converted to MathML code by LaTeXML. Thereafter, the resulting MathML was displayed using our visualization from math.citeplag.org.

of non declarative math input is the use of graphical formula editors that internally represent the entered symbols as Presentation MathML such as MathFind [166].

2.1.4 Structure Description

While a declarative description of the visual impression of a mathematical expression is a good starting point for determining the syntactical structure, this task is not straight forward. For instance, the visual representation of function f applied to the sum of a and b is indistinguishable from the visual representation of the product of f and the sum of a and b (cf. Figure 2.5). When converting the mathematical character layout tree, one faces several challenges:

- Subexpression splitting,
- Disambiguation of operators and
- Identification of invisible operators.

For example, computer algebra systems require knowledge about the mathematical abstract syntax tree. While computer algebra systems use their own tree structures, the Content MathML standard serves as common communication standard. It is supported for instance by Wolfram Mathematica, Sage, Maple and others. However,

the authors are not aware of a comparison of the Content MathML output of those systems.

The open research questions for this task include:

- To which degree can MathML be used to encode the syntactic structure of mathematical formulae?
- To which degree can the syntactic structure be interchanged between systems?
- How do current software systems (such as LaTeXML) perform in representing the syntactical structure?
- How do different methods, heuristics and training effects influence the structure detection task?
- How can the user be involved in the process of adding metadata to represent the correct syntactical structure?

2.1.5 Entity Linkage

While there is a large variety of definitions of semantic data models, we do not aim for a particular meta- or meta-meta model respectively. Our goal for the semantic enrichment process is to assign attributes and annotations to mathematical expressions and their constituents. Our main focus is to annotate (thereafter called semantically enrich) mathematical formulae in a way that is useful to both, users and information retrieval systems, at the same time.

While we will report in detail on semantic enrichment of identifiers in mathematical expressions in Section 3.2 and about the extraction of metadata, such as constraints, proofs and substitutions regarding mathematical formulae in Section 3.1, we will focus on the classification of previous work in this section. However, there will be a slight topical overlap with the above mentioned chapters.

2.1.5.1 Concept Discovery

Several approaches extract information from the surrounding text to retrieve information about mathematical expressions [121, 248, 172, 180, 140, 81]. Quoc et al. [172] extract entire formulae and link them to natural language descriptions from the surrounding text. Yokoi et al. [248] train a support vector machine to extract mathematical expressions and their natural language phrase. Note that the term in this context also includes function words, etc.

2.1.5.2 Metadata extraction

While there is a large body of research on metadata extraction on the document level, the problem of extracting metadata such as constraints, identifier definitions, related keywords or substitutions on a formula level is a relatively untouched field [49]. Refer to Section 3.1 for details.

2.2 Content Querying

The second challenge is **Content Querying**. This ranges from query formulation (Section 2.2.2) over query processing (Section 2.2.3) to hit ranking (Section 2.2.4) and result presentation (Section 2.2.5).

2.2.1 Excursus on Information Retrieval Tasks

In practice, an information retrieval task begins with a user having an actual Information Need (IN). This user interacts in some way with an information system for a certain period of time. After this interaction period his IN is satisfied or not. IR research tries to formalize, analyze and eventually improve this process. One widely used method is the definition of IR tasks. Since there is no consistent definition of the terms used in the literature, we define basic terminology with regards to IR tasks used in this thesis. Traditional IR tasks consist of a list of topics, a dataset and an evaluation measure. We here define the terms **topic** and **query** for this thesis:

Definition 2.2.1 (topic) *a topic in an information retrieval task is a data record that represents a user's information need in a machine-readable form.*

The topic-record consists of public and private fields. The public fields are exposed to the task participants and the private fields are available to the assessors of the IR task only. The private fields usually contain additional information for the assessors to judge the relevance of result. This is for instance a relevance judgment description or an example hit [239]. The public fields might include a list of keywords and a short topic description. Moreover, IR tasks usually provide one or more data sets that shall be used by information systems to answer the aforementioned IN. These datasets are often derived from real world data after potential data cleaning and other preprocessing steps. Developers participate with their IR systems in these tasks by submitting results to the IR task organizers. The IR task organizers organize the evaluation of the results submitted by the participants. Typically, assessors are employed for this tasks. The assessors try to model an actual IN of a users, described by the topic. Their judgment is supposed to quantify the degree of fulfillment of the user's IN.

A topic must not be confused with a query.

Definition 2.2.2 (query) *A query (string) is a well-defined term in a formal syntax.*

A query must certify the following conditions: (1) arbitrage-freeness and (2) reproducibility. Note, that this definition excludes some SQL functions such as `RAND()` and other functions that are not compatible with statement-based replication.

2.2.2 Query Formulation

There are different forms of formula queries. One type are standard ad-hoc retrieval queries, for which a user defines the IN and the MIR system returns a ranked list given a particular data set, such as the NII Testbeds and Community for Information access Research(NTCIR) test dataset [6, 7]. Similarly are interactive formula filter queries, where a user filters a data set interactively until she derives at the result set [61, 227] which is relevant to her needs. Different are unattended queries that run in the background to assist authors during editing or readers to identify related work while viewing a certain formula. This scenario is mostly a vision for future works [259].

However, the NTCIR tasks, that use the α -equality based MathWebSearch (MWS) query language [110], were not the first to invent a query language for mathematical formulae. Youssef and Altamimi proposed a simple query language for mathematical expressions [12, 11, 256].

Beside the implicit query language in MWS, there were a few approaches on query Youssef and Altamimi propose a simple query language for mathematical expressions [12, 11, 256]. Beside the implicit query language in MWS, there are a few approaches on query languages for mathematical formulae [27, 28, 29, 86, 101, 188]. While Bancerek et al. focus on query languages for formalized mathematical libraries [27, 28, 29, 188], Kamil et al. focuses on type-aware wild cards [101]. Moreover, specialized query languages for sub-areas have been developed such as a query language for geometric figures [90].

Another important aspect of query formulation is the field of similarity search. Zhang and Youssef [261] were the first who systematically addressed the problem of similarity measurement for mathematical formulae and identified factors of similarity. In [209] (cf. Section 4.3), we evaluate their measures based on the NTCIR corpus. According to Kamali and Tompa, the presentation layout is the level of choice to define formula similarity [102]. However, this was not confirmed by others.

In the area of formal digital mathematics libraries, Delahaye [59] proposes a method to find a lemma in a Coq prove library. Moreover, Asperti, Guidi, et al. develop a content-based math search engine for Coq proves and provide an unambiguous

question to the user. In an interactive disambiguation step, the user is supposed to transform the question into a real query [20]. Normann and Kohlhase uses ϵ - as extension to α -equivalence based search [176]. Cairns apply Latent Semantic Indexing to find related documents in formal mathematical libraries [40].

A popular approach is to reduce the problem to regular full text search. Guidi and Sacerdoti Coen enumerate 19 publications that reduce to full text search [3, 71, 184, 91, 122, 128, 138, 136, 156, 158, 160, 162, 181, 166, 219, 254, 252, 255].

Libbrecht [127] uses query expansion techniques by fetching related terms from Wikipedia to broaden the search space. However, his system does not allow for formula input, yet.

2.2.3 Advanced Indexes

Slightly related to query formulation, in particular query expansion, are unification and related advanced indexing techniques. While query expansion is performed at querying time, unification is done in the indexing phase. One could argue

- That unification is only one particular view on the data,
- That the computation could also be done at querying time, and
- That the optimizer should decide, when the unification operations should be executed.

However, we decided to separate this section from the previous one to keep a better overview over the large body of previous works.

Most recently Růžička, Sojka, and Láška presented their Lucene-based unification and indexing system which has evolved over many years [137, 219, 194]. Their findings are based on previous work such as the work from Altamimi and Youssef that discusses canonicalization rules for Content MathML. They use equivalence rules to transform Content MathML expressions into a more canonical form [10] and other approaches that use associative commutativity rules such as [257, 215].

With regard to formal mathematical libraries, Formánek, Láška, et al. remove notational inherent ambiguity from the contents in order to improve MIR research [67].

2.2.4 Hit Ranking

Ahmadi and Youssef address the problem of combining and ranking results from different queries, generated from ambiguous formulae, due to errors in the recognition

process [5]. Nguyen, Chang, and Hui propose a question answering task based on learning to rank [133] [174].

2.2.5 Result Presentation

Miller and Youssef proposes a method to highlight Presentation MathML according to content features that were significant for the search result [157]. But also others, such as [138, 136, 162, 246, 254, 253, 252, 255], elaborate on aspects of result presentation.

Libbrecht et al. propose an automatic document summarization method for their application in education [31, 129, 128]. Later, Libbrecht develops a web framework to collect interactive geometry documents. The web interface provides various meta information on the resources in the result view [127]. Bancerek and Rudnicki generate automatic summaries and statistics for DMLs [28]. Líska, Sojka, and Ruzicka presents the WebMiaS Math Search frontend [136] that provides result highlighting and displays the score if requested.

2.3 Efficient Execution

Since the size of digital (mathematical) libraries grows exponentially, scalable data processing methods are required to manage the large data size. Especially interactive MIR application require reasonable response rates and consequently elaborated techniques to process the data. Therefore, the third challenge is **Efficient Execution** for growing data sets. This challenge includes the scalable execution of the solutions to the two aforementioned challenges. While well-established concepts and methods from other computer science disciplines might be applicable (cf. Section 2.3.2), math-specific (complexity) problems require individual solutions (cf. Section 2.3.3).

The theoretical maximal speedup factor S is given by Amdahl's law [13]

$$S(s) = \frac{1}{(1 - p) + \frac{p}{s}}, \quad (2.1)$$

where s is the scaling factor and p the fraction of tasks that benefit from the speedup. Thus, the goal is to maximize p . Therefore, systematically parallel programming paradigms have been established which will be explained in the following section.

2.3.1 Foundations of Parallel Data Processing Systems

To process big data sets efficiently, parallel data processing systems have been designed. Most prominent is MapReduce. In contrast to regular programs, the program design is split into two parts. One part is the modeling of the data flow. While

modern frameworks like Apache Flink, formerly known as Stratosphere, allow a user defined data flow, the data flow is fixed for MapReduce programs. In that case the data-flow reads data from a source, applies the map operator that creates a key value pair, groups all records with the same key and applies the reduce operator on each of the groups. Finally, the data is written to a sink. The fixed operators map and reduce are called second order functions, whereas the user code that is executed with the map and reduce phase respectively is called a first order function.

Flink adds additional second order function such as Join, Co-Group or Cross. However, recent versions of Apache Flink provide much more functionality including stream processing, support for iterations, and stateful operations which reach beyond the functionality that was required for the research carried out in this thesis. Please refer to **Introduction to Apache Flink** [64] for a comprehensive introduction to Apache Flink.

2.3.2 Approaches Adapted from Other Disciplines

Given the data structures mathematical formulae are stored in, it seems quite natural to adapt well-established methods from other disciplines. First and foremost, the area of database systems, i.e., XML processing and indexing, is relevant. For example, Hashimoto, Hijikata, and Nishida present an XML-based math search engine [91]. Kamali and Tompa use a tree edit distance-based similarity approach. In their paper, they evaluate correctness and efficiency of their approach, but do not achieve significantly better results in either of both fields [103]. Lipani, Andersson, et al. use simple L1- and L2-based tokenization methods for their NTCIR 11 contribution [132].

But also technology from the area of traditional text search is used. Larson, Reynolds, and Gey try to adapt traditional text search techniques for MIR. They coin the result “The Abject Failure of Keyword IR for Mathematics Search: Berkeley at NTCIR-10 Math” [125]. Mišutka and Galamboš extend a full text search engine to support mathematics [163].

The well-established ‘Term Frequency Inverse Document Frequency (TFIDF)’ approach from text processing is also applied to formula search. Hagino and Saito flatten the structure of mathematical expressions to bi-grams and use customized rules to eliminate frequent MathML tags [87].

2.3.3 Math-Specific Approaches

Not all MIR systems adapt methods from other areas of informatics. For example, Hambasan, Kohlhase, and Prodescu develop a α -unification-based search engine for mathematical expressions and rely on the text search engine Lucene only for the keywords contained in the search topics from the NTCIR task [89, 117, 118].

Kohlhase and Prodescu describe how to distribute their α -equality based math search engine MWS [115]. MWS [118] is one example of a search engine that is based on substitution tree indexing [80] and the concept of α -equivalence from First Order Logic (FOL). More formally α -equivalence is defined as follows:

Definition 2.3.1 (α -equivalence) *Two λ -terms M, N are α -equivalent, if M can be converted to N with an arbitrary number of α -conversions.*

Here, an α -conversion is defined as:

Definition 2.3.2 (α -conversion) *Let $\lambda x_1 \dots x_n.M$ be a λ -term with n bound variables x_i . An α -conversion $[x_i/y]$ renames the bound variable $x_i \rightarrow y$, i.e., $[x_i/y]\lambda x_1 \dots x_n.M = \lambda x_1 \dots x_{i-1}yx_{i+1} \dots x_n.M$.*

For similarity search Zhang and Youssef propose to use a greedy approximation to find similarity of two formula sub-trees with commutative functions. The evaluation and improvement of this approximation is subject to future research [261].

Asperti, Guidi, et al. develop a performant method to query formal libraries of mathematical knowledge, such as MIZAR, by considering so called meta data. Meta data in this context is a list of formal symbols and the position, where the symbol occurs in the expression [20]. In subsequent work [18], this approach is improved. The meta data-based filtering is reduced to the relational model that can be executed using a relational database.

2.4 Conclusions

In this chapter, we have analyzed about 150 resources including research paper, online resources and software systems and analyzed their contribution to the FDM research. As continuation of the recent survey from Guidi and Sacerdoti Coen, we identified specific research gaps. While there is a good foundation of the first Content Augmentation steps up to the presentation level, especially the conversion from declarative presentation format to semantics provides sufficient room for several research groups. It should to be noted that not only concrete methods, but also testbeds and evaluation measures are absent in those fields.

Different techniques exist for content querying. However, a universal query language that is capable to formally describe the IN of mathematically literate users could not be identified. Neither of existing approaches, like MathQL and the MWS dialect, can be evaluated using more than one implementation. Thus, the area of Query Generation also provides room for future research. However, it should be noted that a current effort such as the OpenDreamKit project³ target on improving open source

³<http://opendreamkit.org/>

computational mathematics ecosystem. If they succeed, and more data in the content and semantic augmentation levels will be available, the need for a universal query language to make better use of that data will be more urgent than ever before.

In the field of efficient execution, there is a significant amount of related works. However, the absence of actual usage data of MIR systems, makes it hard to predict which optimizations are required in the future.

Overall, it has to be noted, that the field of MIR has recently grown significantly. It was reasonable that the first pioneers in that area targeted the whole set of FDM challenges as a whole. However, given the large complexity of the problem, future research projects should focus on individual aspects of the aforementioned FDM challenges.

Chapter 3

Augmentation Methods

After having introduced different semantics levels and the FDM challenges in Chapter 2, this chapter starts with addressing Research Task 2 ‘Develop a representation for semantic formulae that might serve as augmentation target’. This is done in a joint project with NIST. Together with mathematicians, we create the DRMF. For the DRMF a perfect data quality is required and the augmentation speed or the amount of human interaction is less important.

Following the desire for semantic augmentation, we elaborate our idea of Mathematical Language Processing (MLP) from Section A.2 [204]. With respect to Research Task 3 ‘Augment mathematical formulae with regard to the identified extraction targets.’, we did already derive the extraction targets in Section 3.1. Given the large variety and complexity of augmentation needs, we restrict ourselves to mathematical identifiers.

Beside augmenting formulae on the semantic level, this chapter develops a mechanism to improve the presentation level. For instance, we investigate technology to present formulae more effectively. This leads to the developments of the Mathoid software and its application by the Wikimedia foundation to provide math rendering to Wikipedia users. To address the issues with poor presentation of mathematical content in the MediaWiki platform, which we uncovered at the MIR happening, Schubotz and Wicke present “Mathoid: Robust, Scalable, Fast and Accessible Math Rendering for Wikipedia” [207]. This section was accepted as full paper at Conference on Intelligent Computer Mathematics (CICM) 2014.

3.1 Digital Repository of Mathematical Formulae

The purpose of the Digital Repository of Mathematical Formulae (DRMF) is to create a digital compendium of mathematical formulae for Orthogonal Polynomials and Special Functions (OPSF) and of associated mathematical data. The DRMF addresses needs of working mathematicians, physicists and engineers: providing a platform for publication and interaction with OPSF formulae on the web. Whereas Wikipedia and other web authoring tools manifest notions or descriptions as first class objects, the DRMF does that with mathematical formulae. Using MediaWiki extensions and other existing technology (such as software and macro collections developed for the National Institute of Standards and Technology (NIST) Digital Library of Mathematical Functions (DLMF)), the DRMF acts as an interactive web domain for OPSF formulae. In this thesis, the work on the DRMF project serves as fulfillment of research goal 4 ‘Develop a representation for semantic formulae that might serve as augmentation target.’ Parts of this chapter have been published as a short paper “Digital Repository of Mathematical Formulae” by Cohl, McClain, Saunders, Schubotz, and Williams¹ at the Conference on Intelligent Computer Mathematics (CICM) 2014, and as a full paper at CICM 2015 titled “Growing the DRMF with Generic LaTeX Sources” by Cohl, Schubotz, et al.

Compendia of mathematical formulae have a long and rich history. Many scientists have developed such repositories as books and these have been extremely useful to scientists, mathematicians and engineers over the last several centuries (see [38, 39, 65, 66, 79, 142, 187] for instance). While there may be some overlap of formulae in different compendia, one often needs to be familiar with many different compendia to find a specific desired formula. Online compendia of mathematical formulae exist, such as the NIST , subsets of Wikipedia and the Wolfram Functions site. We hope to take advantage of the best aspects of these online efforts, while also incorporating powerful new features that a community-arm of scientists should find beneficial. Our strategy is to start with validated and trustworthy special function data from the NIST DLMF, while adding Web 2.0 capabilities which will encourage community members to discuss mathematical data associated with formulae. These discussions will include internally hyperlinked proofs as well as mathematical connections between formulae in the repository.

Formula Home Page (FHP)s are the principal conceptual objects for the DRMF project. These should contain the full context-free semantic information concerning individual orthogonal polynomial and special function (OPSF) formulae. The DRMF is designed for a mathematically literate audience and should

1. Facilitate interaction among a community of mathematicians and scientists interested in compendia formulae data for OPSF;

¹In addition to the authors of this paper, we honor the contributions of Bruce Miller and Deyan Ginev to the concept and the first prototype.

2. Be expandable, allowing the input of new formulae from the literature;
3. Represent the context-free full semantic information concerning individual formulae;
4. Have a user friendly, consistent, and hyperlinkable viewpoint and authoring perspective;
5. Contain easily searchable mathematics; and
6. Take advantage of modern Mathematical Markup Language tools for easy-to-read, scalably rendered content driven mathematics.

It is the goal of the NIST's DRMF group to build a platform that brings the above features together in a public website for mathematicians, scientists and engineers. We refer to this web platform as the DRMF.

The DRMF project was motivated by the goal of creating an interactive online compendium of mathematical formulae. This need was addressed in SIAM Activity Group OPSF-Net discussions, such as Dmitry Karp (OPSF-Net 18.4, Topic #5). In that OPSF-Net edition, there were two related posts (OPSF-Net 18.4, Topics #6,#7) with a follow-up post in OPSF-Net 18.6, Topic #3. We also found inspiration from the Planetary system [119, 123].

3.1.1 Implementation Goals

In the DRMF project, we have taken advantage of the free and open source MediaWiki software as well as tools developed within the DLMF project [178], such as LaTeXML and the DLMF LaTeX macros [154]. DLMF macros (and extensions as necessary) tie specific character sequences to unique mathematical objects such as special functions, orthogonal polynomials, or to other mathematical symbols associated with these. The DLMF macros are hence used to define OPSF within DRMF and through LaTeXML, their corresponding rendered mathematical symbols. Furthermore, the use of DLMF macros as linked to their definitions within the DLMF, allows for easy access to precise OPSF definitions for the symbols used within the LaTeX source for OPSF formulae. The committed use of DLMF macros guarantees a mathematical and structural consistency throughout the DRMF.

As a web platform, the DRMF provides

1. Formula interactivity,
2. FHP,
3. Centralized bibliography, and

4. Mathematical search.

The DRMF shares the core DLMF component, LaTeXML, which (through the MediaWiki Math extension) processes Wikitext math markup written in LaTeX to produce HTML. For formula interactivity and menus linked to formulae, we have customized the MathJax [45] menu extension. We have also incorporated the MediaWiki: Math and MathSearch [202] (cf. Section A.1) extensions. Within the DRMF, we will develop technology for users to interact with formulae using a clipboard, which allows for easy copy/paste of formula source representations (to include LaTeX with DLMF macros; Presentation or Content MathML; as well as input formats for computer algebra systems such as Mathematica, Maple, and Sage).

The DRMF treats formulae as first class objects, describing them in FHPs that currently contain:

1. A rendered description of the formula itself (required);
2. Bibliographic citation (required);
3. Open section for proofs (required);
4. List of symbols used and links to their definitions corresponding to the DLMF macros (required);
5. Open section for notes relevant to the formula (e.g., formula name, if the formula is a generalization or specialization of some other formula, growth or decay conditions, links to errata pages, etc.);
6. Open section for external links;
7. Substitutions with definitions required to understand the formula; and
8. Constraints the formula must obey.

For each FHP there is a corresponding talk page, and we are incorporating a strategy for handling the insertion of formula errata. A major resource in our ability to implement effective and precise OPSF search will be the use of the DLMF macros in building the LaTeX source for OPSF formulae and related mathematical data.

Next, we present an overview of the seed resources which we plan to incorporate within DRMF. We have been given permission and are seeding the DRMF with data from the NIST DLMF [178]. Moreover, we would also like to extend the DLMF list of formulae by including all relevant formulae which are cited but not listed within the DLMF. We have also been given permission to and are seeding LaTeX formulae data from Koekoek, Lesky, and Swarttouw (KLS) [105]. We will also incorporate Tom Koornwinder's companion of recent arXiv published addendum to KLS [120]. We have also been given permission to incorporate seed formula data from the BMP first

[65, 66]: Higher Transcendental Functions, and Tables of Integral Transforms. Efforts to upload BMP data, as well as any book data without existing LaTeX source, will prove extremely difficult, since this effort will rely on the use of mathematical Optical Character Recognition (OCR) software such as InftyReader to produce LaTeX source for these formulae. Our investigations of state of the art technology showed, that more research is required to extract sufficiently well typesetting information from ORC images. To address this problem, we have started a joint effort together with Alan Sexton to address this problem. Our research on Mathematical OCR is still in its nascence. However, following the test driven development approach we already set up a testing environment (cf. Section A.3). Moreover, we started with initial seeding projects for the aforementioned BMP dataset. We are in communication with other authors and publishers to gain access and permission for other proven sources of mathematical OPSF formulae such as [16, 72, 79, 98] and we are excited about the prospect of seeding proof data (see for instance [75]).

Our current focus is on seeding the DRMF with DLMF data, and we have completed this for Chapter 25 entitled Zeta and Related Functions. Future near-term efforts will focus on seeding the rest of the DLMF data as well as the KLS data including Koornwinder's addendum with DLMF macros incorporated. For LaTeX source where DLMF macros are not present (such as KLS), we are developing tools which automate DLMF macro replacements. Seeding and generating symbol lists are accomplished by converting LaTeX source into MediaWiki Wikitext, in an automated fashion which will be described in the following section.

We are actively in preparation for development of an upload check MediaWiki extension which will verify that community-arm input data conforms to our strict CSS requirements and to ensure that uploaded LaTeX source data has incorporated DLMF macros. Our group, and a team of students, are currently working on development.

3.1.2 Seeding with Generic LaTeX Sources

In this section, we will discuss the DRMF seeding projects whose goal is to import data, for example, from traditional print media (cf. Figure 3.1).

We are investigating various sources for seed material in the DRMF [48]. We have been given permission to use a variety of input resources to generate our online compendium of mathematical formulae. The current sources that we are incorporating into the DRMF are given as follows:

1. NIST DLMFs (DLMF²) [178, 61];
2. Chapters 1, 9, and 14 (a total of 228 pages with about 1800 formulae) from

²We use the typewriter font in this chapter to refer to our seeding datasets.

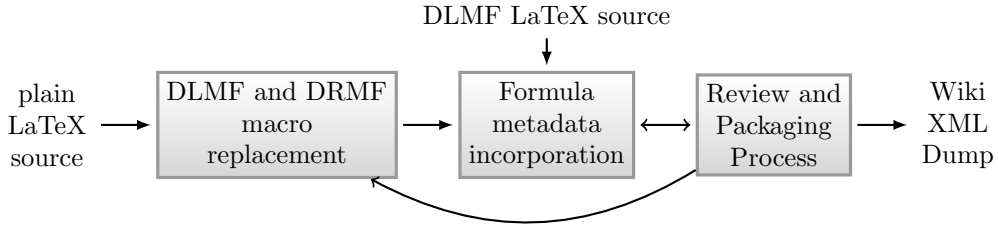


Figure 3.1: Data flow of seeding projects. For most of the input LaTeX source distributions, DLMF and DRMF macros are not incorporated. For the DLMF LaTeX source, the DLMF macros are already incorporated.

the Springer-Verlag book *‘Hypergeometric Orthogonal Polynomials and their q -Analogues’* (2010) by Koekoek, Lesky and Swarttouw (KLS) [105];

3. Tom Koornwinder’s *Additions to the formula lists in ‘Hypergeometric orthogonal polynomials and their q -Analogues’* by Koekoek, Lesky and Swarttouw (KLSadd) [120];
4. Wolfram Computational Knowledge of Continued Fractions Project (eCF);
5. Annie Cuyt’s Continued Fractions package for Special Functions (CFSF);
6. and the BMP first [65, 66] (see Table 3.1)

Note that the DLMF, KLS, KLSadd and eCF datasets are currently being processed within our pipeline. For the BMP dataset, we have furnished high-quality print scans to Alan Sexton and are currently waiting on the math OCR generated LaTeX output for this dataset which is currently being generated. In this section, we focus on DRMF seeding of generic LaTeX sources, namely those which do not contain explicit semantic information.

DRMF seeding projects collect and stream OPSF mathematical formulae into FHPs. FHPs are classified into those which list formulae in a broad category, and the individual FHPs for each formula. Generated FHPs are required to contain bibliographic information and usually contain a list of symbols, substitutions and constraints required by the formulae, proofs and formula names if available, as well as related notes. Every semantic formula entity (e.g., function, polynomial, sequence, operator, constant or set) has a unique name and a link to its definition or description.

For LaTeX sources which are extracted from the DLMF project, the semantic macros are already incorporated [158]. However, for generic sources such as the KLS dataset, the semantic macros need to be inserted in replacement for the LaTeX source which represents that mathematical object.

In Table 3.2 we give representative examples for the trigonometric sine function, gamma function, Jacobi polynomial and little q -Laguerre/Wall polynomials which are

Table 3.1: Overview of the first three stages of the DRMF project. Even though, all three stages were launched sequentially, none of the stages was completed until 2016. Note that the numbers which are given are rough estimates.

	STAGE 1	STAGE 2	STAGE 3
STARTED IN	2013	2014	2015
DATASET	DLMF, semantic LaTeX	KLS, plain LaTeX	eCF: Mathematica BMP: book images
SEMANTIC ENRICHMENT	identify constraints, substitutions, notes, names, proofs, ...	add new semantic macros	image recognition macro suggestion
TECHNOLOGIES	manual review, rule-based approaches	improved rules	natural language processing and machine learning
NUMBER OF FORMULA HOME PAGES	500	1500	5000
HUMAN TIME PER FORMULA HOMEPAGE	10 minutes	5 minutes	1 minute
TEST CORPORA CONTRIBUTION	gold standard for constraint and proof detection	gold standard for macro replacement	evaluation metrics

rendered respectively as $\sin z$, $\Gamma(z)$, $P_n^{(\alpha, \beta)}(x)$, and $p_n(x; a|q)$. These functions and orthogonal polynomials have LaTeX presentations given respectively as indicated in the table. The semantic representations for these functions and orthogonal polynomials are given in the table. The arguments before the @ or @@ symbols are parameters and the arguments after the @ or @@ symbol are in the domain of the functions and orthogonal polynomials. The difference between the @ or @@ symbols indicates a specified difference in presentation, such as the inclusion of the parentheses or not in our trigonometric sine example. For the little q -Laguerre polynomials, one has three arguments within parentheses. These three arguments are separated by a semi-colon and a vertical bar. Our macro replacement algorithm identifies these polynomials, and then extracts the information about what the contents of each argument is. Furthermore, there are many ways in LaTeX to represent open and close parentheses and our algorithm identifies these. Also, since the vertical bar in LaTeX can be represented by ‘|’ or ‘\mid’, we search for both of these patterns. Our algorithm, for instance, also searches for and removes all LaTeX white-space characters such as those given

Table 3.2: Semantic LaTeX macros – Some Examples

Name	Type	Rendering	LaTeX	semantic LaTeX
Trigonometric sine	f	$\sin z$	<code>\sin z</code>	<code>\sin@@{z}</code>
Euler gamma	f	$\Gamma(z)$	<code>\Gamma(z)</code>	<code>\EulerGamma@{z}</code>
Jacobi polynomial	p	$P_n^{(\alpha, \beta)}(x)$	<code>P_n^{(\alpha, \beta)}(x)</code>	<code>\Jacobi{\alpha}{\beta}{n}@{x}</code>
little q -Laguerre polynomial	p	$p_n(x; a q)$	<code>p_n(x;a q)</code>	<code>\littleqLaguerre{n}@{x}{a}{q}</code>

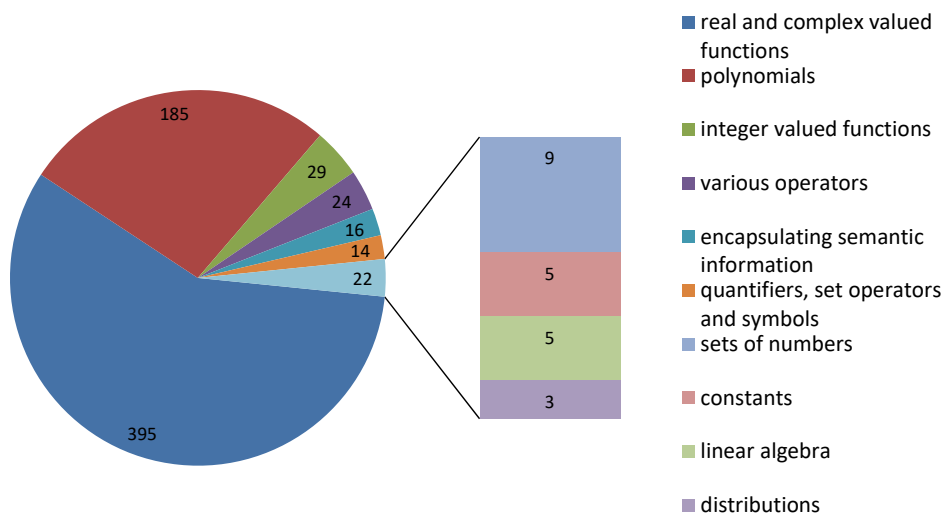


Figure 3.2: Semantic macro breakdown

by `\`, `\!` or `\hspace{}`. There are many other details about making our search and replace work which we will not mention here.

3.1.3 KLS Seeding Project

In this section, we describe how we augment the input KLS LaTeX source in order to generate formula pages (cf. Figure 3.1). We are developing software which processes input LaTeX source to generate output LaTeX source with semantic mathematical macros incorporated. The semantic LaTeX macros that we are using (664 total with 147 currently being used for the DRMF project) are being developed by NIST for

use in the DLMF and DRMF projects. Whenever possible, we use the standardized definitions from the NIST DLMF [178]. If the definitions are not available on the DLMF website, then we link to definition pages in the DRMF with included symbols lists. One main goal of this seeding project is to incorporate mathematical semantic information directly into the LaTeX source. The advantage of incorporating this information directly into the LaTeX source is that mathematicians are capable of editing LaTeX whereas human editing of MathML is not feasible. This enriched information can be further modified by mathematicians using their regular working environment.

For the 3 chapters of the KLS dataset plus the KLSadd dataset, a total number of 89 semantic macros were replaced a total of 3308 times. That is an average of 1.84 macros replaced per formula. Note that the KLSadd dataset is actively being maintained, and when a new version of it is published, in an automated fashion, we incorporate this new information into the DRMF. This fraction will increase when more algebraic substitution formulae are included as formula metadata. The most common macro replacements are given as follows. The macro for the cosine function, Racah polynomial, Pochhammer symbol, q -hypergeometric function, Euler gamma function, and q -Pochhammer symbol were converted a total number of times equal to 117, 205, 237, 266, and 659. Our current conversions, which use a rule-based approach, can be quite complicated due to the nature of the variety of combinations of LaTeX input for various OPSF objects. In LaTeX, there are many ways of representing parentheses which are usually used for function arguments. Also, there are many ways to represent spacing delimiters which can mostly be ignored as far as representing the common semantic information for a mathematical function call. Our software canonicalizes these additional meaningless degrees of freedom and generates easy-to-read semantic LaTeX source and improves the rendering. Developing automatic software which performs macro replacements for OPSF functions in LaTeX is a challenging task. The current status of our rule-based approach is highly tailored to our specific KLS and KLSadd input LaTeX source.

Historically, the desired need for formal consistency has driven mathematicians to adapt consistent and unique notations [16]. This is extremely beneficial in the long run. We have interacted on a regular basis with the authors of the KLS and KLSadd datasets. They agree that our assumptions about consistent notations are correct and they consider using semantic LaTeX macros in future volumes. Certainly, the benefit of using these macros in communicating with different computer systems is clear.

Once semantic macros are incorporated, the next task is to identify formula metadata. Formula metadata can be identified within and must be associated with formulae. One must then identify semantic information for the formula within the surrounding text to produce formula annotations which describe this semantic information. There are annotations which can be summarized as constraints, substitutions, proofs and formula names if available, as well as related notes. The automated extraction of formula metadata is a challenging aspect of the seeding project and future computer implementations might use machine learning methods to achieve this goal (cf. Section 3.2).

However, we have built automated algorithms to extract formula metadata. We have for instance identified substitutions by associating definitions for algebraic or OPSF functions which are utilized in surrounding formulae. The automation process continues by merging these substitution formulae as annotations in the original formulae which use them. Another extraction algorithm we have developed is the identification of related variables, understanding their dependencies and merging corresponding annotations with the pre-existing formula metadata. We have manually reviewed the printed mathematics to identify formula metadata. After we have exhausted our current rule-based approach for extracting the formula annotations, we will perform the manual insertion of the missing identified annotations into the LaTeX source. This will then be followed by careful checking and expert editorial review. This also evaluates the quality of our rule-based approach and creates a gold standard for future programs.

Once the formula metadata has been completely extracted from the text, then the remainder of the text should be removed and one is left with a list of LaTeX formulae with associated metadata. From this list (at the current stage of our project), we use this semantic LaTeX source to generate Wikitext. One of the features of the generated Wikitext is that we use a glossary that we have developed of DLMF and DRMF macros to identify semantic macros within a formula and its associated metadata. Presentation and meaningful Content MathML is generated from the DLMF and DRMF macros using a customized LaTeXXML server (<http://gw125.iu.xsede.org>) hosted by the XSEDE project that includes all generated semantic macros. From this glossary, we generate symbols lists for each formula, which uses recognized symbols. The generated Wikitext is converted to the MediaWiki Extensible Markup Language (XML)dump format which is then bulk imported to our Wiki instance. Our DRMF Wiki has been optimized for MathML output. Because we are using Mathoid to render mathematical expressions [207] (cf. Section A.3), browsers without MathML support can display DRMF formulae within MediaWiki. However, some MathML-related features (such as copying parts of the MathML output) are not available on these browsers.

At the moment, there are 1282 KLS and KLSadd Wikitext pages. The current number of KLS and KLSadd FHPs is 1219 and the percentage of non-empty symbols lists in FHPs is given by 98.6 percent. This number will increase as we continue to merge substitution formulae into associated metadata and as we continue to expand our macro replacement effort. We have detected 208 substitutions which originally appeared as formulae. We inserted these in an automated fashion into 515 formulae. The goal of our learning is to obtain a mostly unambiguous content representation of the mathematical OPSF formulae which we use.

3.1.4 Future Outlook

The next seeding projects which we will focus on are those which correspond to image and Mathematica or Maple inputs (see Table 3.1). We have been given permission from Caltech to use the BMP dataset within the DRMF. In the BMP dataset, the original source for data are printed pages of books. We are currently collaborating on the development of mathematical OCR software [214] with Alan Sexton (cf. Section A.3) for use in this project. We plan to utilize this math OCR software to generate LaTeX output which will be incorporated with the DLMF and DRMF semantic macros using our developed macro replacement software. We have also been given permission by Springer-Verlag to incorporate the KLS dataset into the DRMF. For about one year, we have been developing software which has become part of a pipeline which starts from the original LaTeX source and generates Wikitext for use in the MediaWiki software which is used by Wikipedia.

We are already developing for our next sources, namely the incorporation of the Wolfram eCF dataset, and the Maple CSFS dataset into the DRMF. We have been furnished the Mathematica source (also known as Wolfram language) for this dataset and we are currently developing software which translates in both directions from the Wolfram language to our semantic LaTeX source with DRMF and DLMF macros incorporated (cf. Table 3.1).

For the DLMF source, due to the hard efforts of the DLMF team for more than the past ten years, we already have semantic macros implemented, and all that remains is to extract the metadata from the text associated with formulae, removing the text after the content has been transferred, converting formulae information in tables to lists of distinct formulae, and generating FHPs. We already have mostly achieved this for DLMF Chapter 25 on the Riemann Zeta function and are currently at work on Chapters 5 (gamma function), 15 (hypergeometric function), 16 (generalized hypergeometric functions), 17 (q -hypergeometric and related functions) and 18 (orthogonal polynomials) which will ultimately be merged with the KLS and KLSadd datasets. Then we will continue to the remainder of the DLMF chapters.

Once semantic information has been inserted into the LaTeX source, there is a huge number of possibilities on how this information can be used. Given that our datasets are collections of OPSF formulae, we plan on taking advantage of the incorporated semantic information as an exploratory tool for symbolic and numerical experiments. For instance, one may use this semantic content to translate to computer algebra system, computer languages such as those used by Mathematica, Maple or Sage. One could then use the translated formulae while taking advantage of any of the features available in those software packages. We should also mention that the DRMF seeding projects generate real Content MathML. This has been a huge problem for Mathematical Information Retrieval (MIR) research for many years [118, 173]. One major contribution of the DRMF seeding projects is that they offer quite reasonable Content MathML.

From a methodological point of view, we are going to develop evaluation metrics that measure the degree of semantic formula enrichment. These should be able to evaluate new approaches such as Mathematical Language Processing (MLP) [180] (cf. Section 3.2) and/or machine learning approaches based on the created gold standard. Additionally, we are considering the use of sTeX [114], in order to simplify the definition of new macros. Eventually, we can also develop a heuristic which suggests new semantic macros based on statistical analysis.

3.2 Mathematical Language Processing and Namespace Discovery

In this section, we present our **Mathematical Language Processing (MLP)** approach as one of the major augmentation methods. In a first step we extract natural language description regarding individual identifiers from the surrounding text. Thereafter, we expand this approach to mathematical concepts. Moreover, we use clustering methods to discover namespaces in mathematical language. This opens a large field for future research opportunities. Parts of that section have been published as “Mathematical Language Processing Project” by Pagel and Schubotz at the Conference on Intelligent Computer Mathematics (CICM) in [180] and “Semantification of Identifiers in Mathematics for Mathematical Information Retrieval (MIR)” by Schubotz, Grigorev, et al. at SIGIR first 2016 conference [211].

Mathematical formulae are essential in science, but face challenges of ambiguity due to the use of a small number of identifiers to represent an immense number of concepts. Corresponding to word sense disambiguation in Natural Language Processing (NLP), we disambiguate mathematical identifiers. By regarding formulae and natural text as one monolithic information source, we are able to extract the semantics of identifiers in a process which Pagel and Schubotz termed (MLP) [180]. As scientific communities tend to establish standard (identifier) notations, we use the document domain to infer the actual meaning of an identifier. Therefore, we adapt the software development concept of namespaces to mathematical notation. Thus, we learn namespace definitions by clustering the MLP results and mapping those clusters to subject classification schemata. In addition, this gives fundamental insights into the usage of mathematical notations in Science, Technology, Engineering and Mathematics (STEM). Our gold standard-based evaluation shows that MLP extracts relevant identifier definitions. Moreover, we discover that identifier namespaces improve the performance of automated identifier definition extraction, and elevate it to a level that cannot be achieved within the document context alone.

3.2.1 Problem and Motivation

Current MIR approaches perform well in identifying formulae that contain the same set of identifiers or have a similar layout tree structure [7]. Refer to Section 4.1.

However, the ambiguity of mathematical notation decreases the retrieval effectiveness of current MIR approaches. Since the number of mathematical concepts by far exceeds the number of established mathematical identifiers, the same identifier often denotes various concepts [118]. For instance, ‘ E ’ may refer to ‘energy’ in physics, ‘expected value’ in statistics or ‘elimination matrix’ in linear algebra. Analyzing the identifier-based and structural similarity of formulae without considering the context of a formula can therefore lead to the retrieval of non-relevant results.

Ambiguity is a problem that mathematical notation and natural language have in common. Since words are also often ambiguous [58, 76, 118], **Word Sense Disambiguation** [100], i.e., identifying the meaning of an ambiguous word in a specific context [100], is an integral part of NLP. Typical approaches for Word Sense Disambiguation replace a word by its meaning [223] or append the meaning to the word. For example, if the ambiguous word ‘man’ has the meaning ‘human species’ in a specific context, one can replace it by `man_species` to contrast it from the meaning ‘male adult’, replaced by `man_adult`. We transfer this idea to ambiguous mathematical identifiers. If the identifier E has the meaning **energy** in the context of physics, one could replace E by E_{energy} given one can determine that E is indeed used as energy in this context.

In this section, we propose a method to semantically enrich mathematical identifiers by determining and assigning the context (namespace) in which the identifier is used, e.g., mathematics or physics. We determine the **namespace** of an identifier by analyzing the text surrounding mathematical formulae using NLP techniques. In software development, a namespace refers to a collection of terms that is grouped, because it shares functionality or purpose. Typically, namespaces are used to provide modularity and to resolve name conflicts [63]. We extend the concept of namespaces to mathematical identifiers and present an automated method to learn the namespaces that occur in a document collection.

Today’s MIR systems treat formulae and natural language as separate information sources [7] (cf. Section 4.1). While current systems offer retrieval from both sources (formulae and text), they typically do not link them. For example, math-aware search systems allow to search in formulae by specifying a query using mathematical notation or specialized query languages. To search in the text, MIR systems support traditional keyword search [7] (cf. Section 4.1).

We deem the MLP approach promising for two reasons. First, a large-scale corpus study showed that around 70 percent of the symbolic elements in scientific papers are explicitly denoted in the text [246]. Second, although almost all identifiers have multiple meanings, mathematical notation obeys conventions for choosing identifiers

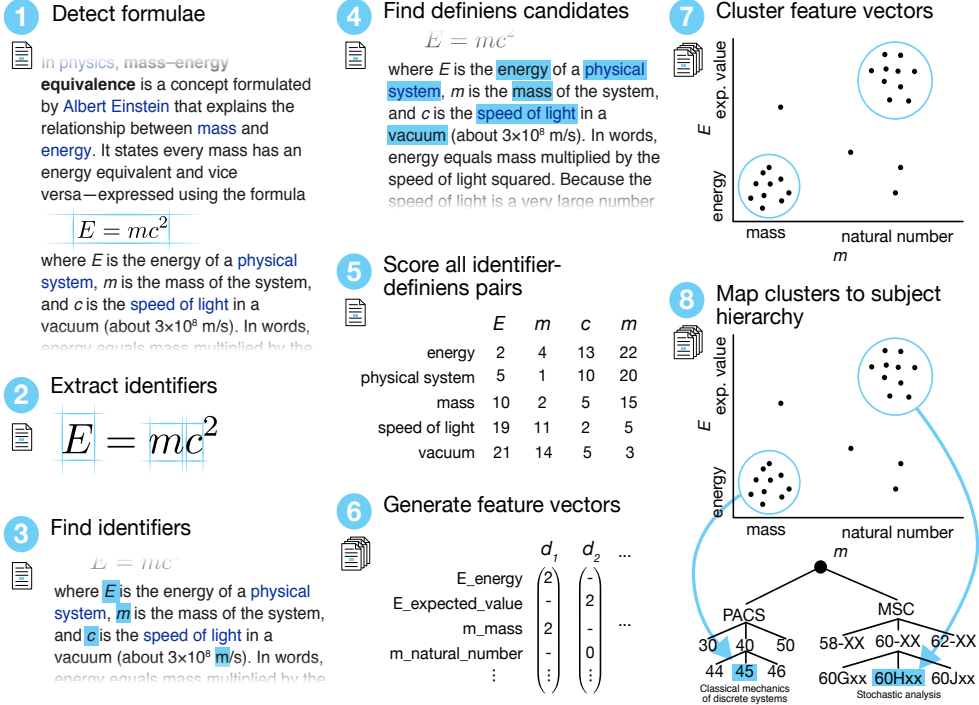


Figure 3.3: Overview of the document based Mathematical Language Processing pipeline (steps 1-5), and the corpus based namespace discovery pipeline (steps 6-8). For each step, a detailed description is available in the corresponding subsection of Chapter 2.

[41, 118]. Therefore, we propose that identifying the namespace of identifiers can improve their disambiguation and the capabilities for machine processing mathematics in general.

In summary, the contributions we make in this section are:

1. A method to extract the semantic meaning of mathematical identifiers from the text surrounding mathematical formulae;
2. A method to learn the set of mathematical namespaces occurring in a collection;
3. A method that utilizes identified mathematical namespaces to improve the disambiguation of mathematical identifiers; and
4. A large scale analysis of identifier use as part of the mathematical notation in different scientific fields.

Related Work Several approaches extract information from the surrounding text to retrieve information about mathematical formulae [121, 248, 172, 180, 140, 81]. Quoc et al. [172] extract entire formulae and link them to natural language descriptions from the surrounding text. Yokoi et al. [248] train a support vector machine to extract mathematical expressions and their natural language phrase. Note, that this phrase also includes function words, etc. In [180], we suggest a Mathematical Language Processing framework - a statistical approach for relating identifiers to definitia, in which we compare a pattern-based approach and the MLP approach with Part Of Speech (POS) tag-based distances. We find the MLP approach to be more effective. The findings of Kristianto et al. [121] confirm these findings.

Our approach is the first that uses the concept of namespaces to improve the extraction of semantics regarding mathematical identifiers. While other approaches only use one document at a time to extract the description of a specific formulae [121, 248, 140], we use a large scale corpus and combine information from different documents to extract the meaning of a specific identifier. Furthermore, our task is more specific. We limit the extraction of mathematical expressions to identifiers and extract semantic concepts instead of descriptions.

3.2.2 Preliminary Study

In the following subsection we briefly report on our preliminary work on identifier definition extraction, where we compared a pattern-based approach with the MLP approach.

3.2.2.1 Pattern-Based Definition Discovery

At first, we implemented a simple identifier definition extractor that is based on a set of static patterns. As this is a fairly robust approach and easy to implement, it serves as a good reference point in terms of performance. It simply iterates through the text, trying to find word groups, that are matched by a pattern. The patterns being used to discover description terms are depicted in Table 3.3. Due to the fact that we already tokenized and annotated the articles in a previous step in the MLP system, we can make use of POS tags here as well.

Note, determiners not only contain articles, but also quantifiers and distributives. The last pattern in Table 3.3 contains ‘*/DT’. This is a shorthand for every word, that has the POS tag ‘DT’ (determiner). Otherwise this pattern would be rather large, as it needs to contain every possible determiner. **IDENTIFIER** as well as **DESCRIPTION** are placeholder, that mark the positions of the entities from a possible definition relation.

Pattern
<code><description> <identifier></code> <code><identifier> is <description></code> <code><identifier> is the <description></code> <code>let <identifier> be the <description></code> <code><description> is are denoted by <identifier></code> <code><identifier> denotes */DT <description></code>

Table 3.3: Implemented static patterns

3.2.2.2 Statistical Definition Discovery

We detect relations between identifiers and their description in two steps. First, we extract the identifiers from the formulae found in an article, and second, we determine their description from the surrounding text.

Extracting relevant identifiers from the article relies on the assumption that the author will use `<math/>` tags for all formulae. This said, a formula that is written in the running text cannot be recognized, and therefore, cannot be extracted by our system.

The fact that we can estimate all relevant identifiers for an article (cf. Section 3.2.2.3), combined with some common assumptions about definition relations, can be exploited to largely reduce the set of candidates that need to be ranked. Please note that this reduction is essential for retrieving the correct relations for our approach. Otherwise almost any word would be ranked and the precision of the retrieval would drop significantly.

The basic assumption of our approach is that the two entities of a definition relation co-occur in the same sentence. In other words, if we want to retrieve the description for an identifier, only sentences containing the identifier could include the definition relation. Having said this, any other sentences can be ignored. Furthermore, we assume that it is more likely to find the description in first sentences than in the latter. This is based on the idea that authors introduce the meaning of an identifier and then subsequent use the identifier, without necessarily repeating its definition.

Another assumption can be made about the lexical class of the definition relation we want to rank. The descriptions are nouns or even noun phrases (e.g., ‘the effective spring constant k ’ or ‘mass m of something’). We discard all other words (according to their POS tag) except noun phrases and Wikipedia hyperlinks. These are the candidates descriptions for a definition relation. Noun phrases and hyperlinks may consist of multiple words. For all intents and purposes, it is not necessary to treat noun phrases and hyperlinks as a set of words, and therefore, they will be treated subsequently as if they were one. This is important due to the fact that the overall ranking will be greatly influenced by the distance of candidates to the position of the

identifier.

Numerical Statistics Each description candidate is ranked with the weighted sum

$$R(n, \Delta, t, d) = \frac{\alpha R_{\sigma_d}(\Delta) + \beta R_{\sigma_s}(n) + \gamma \text{tf}(t, d)}{\alpha + \beta + \gamma} \mapsto [0, 1]. \quad (3.1)$$

The weighted sum depends on the distance Δ (amount of word tokens) between identifier and the description term t , the distance between formula and sentence n , and the term frequency $\text{tf}(t, d)$. The distance was normalized with $R_{\sigma}(\Delta) = \exp \left[-\frac{1}{2} \frac{\Delta^2 - 1}{\sigma^2} \right]$. We assume that the probability to find a relation at $\Delta = 1$ is maximal. For example, in the text fragment ‘the energy E, and the mass m’, in order to determine the full width at half maximum of our distribution, we evaluated some articles manually and found $R_{\sigma_d}(1) \approx 2R_{\sigma_d}(5)$ and thus $\sigma_d = \sqrt{\frac{12}{\ln 2}}$. The probability to find a correct definition decays to 50% within three sentences. Consequently $\sigma_s = 2(\ln 2)^{-\frac{1}{2}}$.

Robustness. The classic Term Frequency Inverse Document Frequency (TFIDF) [196] statistic reflects the importance of a term to a document. For our task, the inverse document frequency (idf) assigns high penalties to frequent words like ‘length’, as opposed to words seldom seen such as ‘Hamiltonian’. These are both valid definitions for identifiers. As the influence of $\text{tf}(t, d)$ on the sensitivity of the overall ranking (3.1) seems to be very high, we reduce the impact with the tuning parameters $\gamma = 0.1$ and remain $\alpha = \beta = 1$. Please note that the algorithm currently only takes sentences into account which were found in a single article. In the future, the MLP system will examine sets of closely related articles. This will leverage the problem that distributional properties will be volatile on term universes with very few members (e.g., term frequencies in a single sentence).

Implementation. We implemented the MLP processing system [179] as Stratosphere data-flow using Java which allows for scalable execution, application of complex higher order functions, and easy integration of third party tools such as Stanford NLP and the Mylyn framework for mark-up parsing.

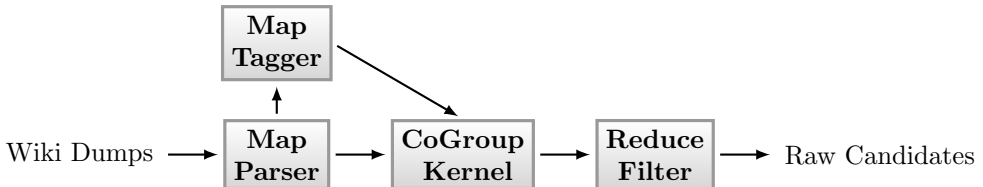


Figure 3.4: Data flow of the Stratosphere program

3.2.2.3 Evaluation

Identifier Retrieval Throughout our experiments, we made some observations that had an impact on the accuracy of retrieving the correct set of identifiers. First of all, people tend to use `texvc` (cf. Section A.3) only as a typesetting language and neglect its semantic capabilities. For example, `\text{log}` is more often used than the correct operator `\log`. Another problem is that sometimes people use indices as a form of ‘in field’ annotation, like T_{before} and T_{after} . The identifier T is defined in the surrounding text, but neither T_{before} nor T_{after} . There are more ambiguities. For example, the superscripted 2 in x^2 and σ^2 can be interpreted as the power or as a part of the identifier. Another ambiguity is that the multiplication sign can be omitted, so that it is undecidable for a naïve program whether ab^2 contains one or two identifiers.

We took a very conservative approach and preprocessed all formulas. The \TeX command `\text{}` blocks along with subscriptions containing more than a single character will be removed before analysis. Superscripts will also be ignored in terms of being a part of the identifier. Moreover, we created a comprehensive blacklist to improve the results further. Identifier like ‘a’, ‘A’, and ‘I’, which are also very common in the English language, could be easily matched by our processor in the surrounding text, and therefore, will also be blacklisted. Additionally, we blacklist common mathematical operators, constants, and functions.

We took a sample of 30 random articles and counted all matches by hand. The resulting estimates for the identifier retrieval performance are **recall: 0.99** and **precision: 0.86**, which satisfy our Information Need (IN)s, as we are mostly interested in recall at this stage.

3.2.2.4 Description Retrieval

We ran our program on a dataset of about 20,000 articles, all containing `<math/>` tags, and retrieved about 550,000 candidate relations. The most common definition relations are listed in Table 3.4.

Identifier	Descriptions	Count
n	number	1709
t	time	1480
M	mass	1042
r	radius	752
T	temperature	666
θ	angle	639
G	group	635

Table 3.4: Most common definition relations

Observations. We observed some poorly ranked relations. For example, in the fragment ‘where $\phi(r_i)$ is the electrostatic potential’, the distance is $\Delta(\phi, \text{electrostatic potential}) = 6$. This is due to counting brackets and function arguments as words. Also wrongly tagged words like ‘Hamiltonian’ as an adjective leads to false negatives.

Comparison of the approaches At the start of our project there were no gold standard datasets available to measure the performance of identifier definition extractors. Thus, we created one on our own. This is a very time consuming job. At the moment, the dataset only contains two large articles (revision ids included) with around 100 identifier definitions. This dataset is also available on the project repository.

As in many articles, the articles in the evaluation dataset contain identifiers whose description cannot be retrieved. This is due to two reasons. First and foremost, the identifier found in a formula is never mentioned in the surrounding text, and therefore, no description can be extracted. Second, the identifier is somehow ambiguous (cf. Section 3.2.2.3) and has been dropped. Most notably, identifiers like I_{xx} will be discarded because of an ambiguous index that contains multiple letters.

Unfortunately 32 out of 99 identifiers from our dataset fall into that category. We have decided to evaluate the performance of the remainder, as those 32 do not convey any conceptual flaws. From the users standpoint, the overall performance (in terms of recall) of such a system would be rather annoying. As we are only interested in evaluating the performance of the **MLP Ranking** algorithm itself, it is safe to ignore those 32 identifiers.

	MLP Ranking ($k = 1$)	MLP Ranking ($k = 2$)	Pattern Matching
Precision	0.872	0.915	0.911
Recall	0.839	0.892	0.733

Table 3.5: Evaluation results. Note: k equals the amount of the top ranked candidate definitions.

Our results show that the unoptimized MLP approach keeps up with the performance of the simple pattern matcher. Furthermore, we observed that it is more robust in terms of recall, as it is less vulnerable to small changes in the sentence structure.

Conclusions from the preliminary experiment The statistical approach outperforms the pattern matching approach, as it has slightly better precision and recall values as well compared, and is more general since it does not rely language specific patterns. Thus, we continue using that approach.

3.2.3 Methods

The goal of MLP is to extract identifier definitions from a text that uses mathematics. Formally, a **definition** consists of three parts: definiendum, definiens and definitor. **Definiendum** is the expression to be defined. **Definiens** is the phrase that defines the definiendum. **Definitor** is the verb that links definiendum and definiens. An identifier definition is a definition where the definiendum is an identifier.

According to ISO/IEC 40314: ‘Content **identifiers** represent ‘mathematical variables’ which have properties, but no fixed value.’ Identifiers have to be differentiated from **symbols**, that refer to ‘specific, mathematically- defined concepts’ such as the operator $+$ or the sin function. Identifier definiens pairs are candidates for identifier definitions. Since we do not use the definitor, we only extract the definiendum (identifier) and the definiens (natural language term), and subsequently, identifier definiens pairs as candidates for identifier definitions. To illustrate, we introduce the following running example:

Example 2: Mass energy equivalence

The relation between energy and mass is described by the mass energy equivalence formula $E = mc^2$, where E is energy, m is mass, and c is the speed of light.

This description includes the formula $E = mc^2$, the three identifiers E , m and c , and the following identifier definitions: (E, energy) , (m, mass) , and $(c, \text{speed of light})$.

In our approach (see Figure 3.3), we divide the MLP pipeline into the following steps:

1. Detect formulae;
2. Extract identifiers;
3. Find identifiers;
4. Find definiens candidates; and
5. Score all identifier definiens pairs.

① Detect formulae

In a first step, we need to differentiate between formulae and text. In this section, we assume that all formulae are explicitly marked as mathematics and that everything marked as mathematics actually is mathematics. However, in real world documents such as conference papers, posters or Wikipedia articles, some formulae are typed using the unicode symbols instead of math mode. As this type of formula is hard to detect, we decided to exclude it from our analysis. Moreover, not all structures marked as formulae are really mathematical formulae. In some cases unmarked text like $\frac{\text{work done}}{\text{heat absorbed}}$ or chemical formulae $2 \text{ H}_2\text{O} \rightarrow 2 \text{ H}_2 + \text{O}_2$ are also marked as mathematics. One might develop heuristics to discover words and chemical structures within mathematical markup, but this is outside of the scope of this research.

② Extract identifiers

After having identified the formulae, we extract the list of identifiers from within the formulae. In the above example, this means to extract the identifiers E , m , and c from the formula $E = mc^2$. Mostly, identifiers (in formulae and text), are not explicitly marked as identifiers. Consequently, we develop a heuristic to extract identifiers by assuming the following characteristics: an identifier consists of one variable or a combination of a variable and one or multiple subscripts.

In the following, we will discuss advantages and limitations of this heuristic. In this process, we delineate four limitations (special notation, symbols, sub-super-script, incorrect markup) which we will quantify in the evaluation section. We observe that more complex expressions are sometimes used on behalf of identifiers, such as σ^2 for the ‘variance’, without mentioning σ and ‘standard deviation’ at all or ΔS for ‘change in entropy’. In this work, we focus on atomic identifiers and thus prefer to extract the pair (S , entropy) instead of (ΔS , change in entropy). The disadvantage of this approach is that we miss some **special notation** such as contra-variant vector components like the coordinate functions x^μ in Einstein notation. In this case, we are able to extract (x , coordinate functions) with our approach, which is not incorrect,

but less specific than $(x^\mu, \text{coordinate functions})$. In addition, we falsely extract several **symbols**, such as the Bessel functions J_α, Y_α , but not all symbols, i.e., we do not extract symbols that use **sub-super-scripts** like the Hankel function $H_\alpha^{(1)}$. Note that especially the superscript is not used uniformly (e.g., it may refer to power, n -th derivative, Einstein notation, inverse function). The most prominent example is the \sin symbol, where $\sin^2 : x \mapsto (\sin(x))^2$ vs. $\sin^{-1} : \sin(x) \mapsto x$, for all $x \in [-1, 1]$. Far less debatable, but even more common is the problem of **incorrect markup**. The one variable assumption tokenizes natural language words like *heat* into a list of four variables h, e, a, t .

Identifiers often contain additional semantic information, visually conveyed by special diacritical marks or font features. Examples of diacritics are hats to denote estimates (e.g., \hat{w}), bars to denote the average (e.g., \bar{X}) or arrows to denote vectors (e.g., \vec{x}). Regarding the font features, bold lower case single characters are often used to denote vectors (e.g., \mathbf{w}) and bold upper case single characters denote matrices (e.g., \mathbf{X}), while double-struck fonts are used for sets (e.g., \mathbb{R}), calligraphic fonts often denote spaces (e.g., \mathcal{H}) and so on. Unfortunately, there is no common notation established for diacritics across all fields of mathematics and thus there is a lot of variance. For example, a vector can be denoted by \vec{x} , \mathbf{x} or \mathbf{x} , and the real line can be denoted either by \mathbb{R} or \mathbf{R} .

To decide if two identifiers are identical, we need a comparison function that eliminates invariants in the input format. For example, the inputs `c_0` and `$c_{\{0\}}$` produce the same presentation c_0 in LaTeX and therefore have to be considered as equivalent. In this work, we compare the identifiers based on abstract syntax trees which eliminates most of the complications introduced by the invariants in the input encoding. We considered to reduce the identifiers to their root form by discarding all additional visual information, such that \bar{X} becomes X , \mathbf{w} becomes w and \mathfrak{R} becomes R . The disadvantage of this approach is the loss of additional semantic information about the identifier that are potentially useful. For instance, \mathbf{E} usually denotes the electric field, compared to E which is often used for energy. By removing the bold font, we would lose this semantic information. Therefore, we decided against using the root form in our approach.

③ Find identifiers

In a next step, all identifiers that are part of the formulae have to be identified in the surrounding text. Therefore, we use mathematical formulae that only consist of a single identifier, or textual elements that are not marked up as mathematics (i.e., words) and are equivalent to one of the identifiers extracted in the formulae. In the above example, the identifiers E, m and c have to be identified in the text: ‘The relation between energy and mass is described by the mass-energy equivalence formula [...], where E is energy, m is mass, and c is the speed of light.

④ Find definiens candidates

We are not only interested in the identifier, but also in its definiens. Therefore, we extract identifier definiens pairs (identifier, definiens) as candidates for identifier definitions. For example, (E , energy) is an identifier definition, where E is an identifier, and ‘energy’ is the definiens. In this step, we describe the methods for extracting and scoring the identifier definitions in three sub-steps:

1. Math-Aware POS Tagging;
2. POS-based distances; and
3. Scoring of definiens candidates.

Pagel and Schubotz [180] (cf. Section 3.2.2) found the MLP method with a POS based distance measure in a probabilistic approach to outclass a pattern based method. Thus, we use the POS based distances methods here to extract identifier definitions. First, we define definiens candidates:

1. Noun (singular or plural);
2. Noun phrases (noun-noun, adjective-noun); and
3. Special tokens such as inner Wiki links.

We assume that successive nouns (both singular and plurals), possibly modified by an adjective, are candidates for definitia. Thus, we include noun phrases that either consist of two successive nouns (e.g., ‘mean value’ or ‘speed of light’) or an adjective and a noun (e.g., ‘gravitational force’).

Authors often use special markup to highlight semantic concepts in written language. For example, in Wikipedia articles, Wiki markup, a special markup language for specifying document layout elements such as headers, lists, text formatting and tables, is used. In the Wikipedia markup processing, we retain inner Wikipedia links that link to another article that describes the semantic concept, which eliminates the ambiguity in the definiens itself. This link is an example for a definiens candidate of type special token. POS assigns a tag to each word in a given text [100]. Although the POS Tagging task is mainly a tool for text processing, it can be adjusted to scientific documents with mathematical expressions [197, 180]. Therefore, we tag math-related tokens of the text with math specific tags [197]. If a math token is only one identifier, an identifier tag is assigned rather than a formula tag. We introduce another tag for inner Wiki-links. For the extraction of definiens candidates, we use common natural language POS tags as well as the following three task specific tags:

1. Identifiers;

2. Formulae; and
3. Special tokens.

Generally, the Cartesian product of identifiers and definiens might serve as identifier definition candidate.

⑤ Score all identifier definiens pairs

To extract the definiens candidates, we make three assumptions, according to [180] (cf. Section 3.2.2):

1. Definiens are noun phrases or a special token;
2. Definiens appear close to the identifier; and
3. If an identifier appears in several formulae, the definiens can be found in a sentence in close proximity to the first occurrence in a formula.

The next step is to select the most probable identifier definition by ranking identifier definition candidates by probability [180] (cf. Section 3.2.2). The assumption behind this approach is that definientia occur closely to their related identifiers, and thus the closeness can be exploited to model the probability distribution over identifier definition candidates. Thus, the score depends on (1) the distance to the identifier of interest and (2) the distance to the closest formula that contains this identifier. The output of this step is a list of identifier definiens pairs along with the score. Only the pairs with scores above the user specified threshold are retained.

The candidates are ranked by the following formula:

$$R(n, \Delta, t, d) = \frac{\alpha R_{\sigma_d}(\Delta) + \beta R_{\sigma_s}(n) + \gamma \text{tf}(t)}{\alpha + \beta + \gamma}.$$

In this formula Δ is the number of tokens between identifier and definiens candidate, $R_{\sigma_d}(\Delta)$ is a zero-mean Gaussian that models this distance, parametrized with the variance σ_d , and n is the number of sentences between the definiens candidate and the sentence in which the identifier occurs for the first time. Moreover, $R_{\sigma_s}(n)$ denotes a zero-mean Gaussian, parameterized with σ_s , and $\text{tf}(t)$ is the frequency of term t in a sentence, and the weights α, β, γ combine these quantities. Therefore, we reuse the values suggested in [180] (cf. Section 3.2.2), namely $\alpha = \beta = 1$ and $\gamma = 0.1$.

We also tested a refined strategy which takes into account that the same definition might be explained multiple times in a document and calculated a refined weighting $R_\Sigma = (\eta - 1)^{-1} \sum_{i=1}^n \eta^{-i} R_i$. Thereby, R_i iterates over all weightings from within one document that lead to one definition. However, this did not lead to a significant

performance increase for the task at hand, so we dropped this approach. Note that the idea is revived in the Namespace Discovery section, where multiple documents are considered at the same time.

3.2.3.1 Namespace Discovery

In this section, we describe the adaptation of the idea of namespaces to identifier disambiguation and the process of namespace discovery to extract identifier-definitions in the following steps:

1. Automatic Namespace Discovery;
2. Document Clustering;
3. Building Namespaces; and
4. Building Namespace Hierarchy.

Automatic Namespace Discovery Namespaces in well-defined software exhibit low coupling and high cohesion [124]. Coupling describes the degree of dependence between namespaces. Low coupling means that the dependencies between classes of different namespaces are minimized. Cohesion refers to the dependence within the classes of the same namespace. High cohesion principle means that the related classes should be put together in the same namespace. We define a notation \mathcal{N} as a set of pairs $\{(i, s)\}$, where i is an identifier and s is its semantic meaning or definiens, such that for any pair $(i, s) \in \mathcal{N}$ there is no other pair $(i', s') \in \mathcal{N}$ with $i = i'$. Two notations \mathcal{N}_1 and \mathcal{N}_2 conflict if there exists a pair $(i_1, s_1) \in \mathcal{N}_1$ and a pair $(i_2, s_2) \in \mathcal{N}_2$ such that $i_1 = i_2$ and $s_1 \neq s_2$.

Thus, we can define a namespace as a named notation. For example, $\mathcal{N}_{\text{physics}}$ can refer to the notation used in physics. For convenience, we use the Java syntax to refer to specific entries of a namespace [78]. If \mathcal{N} is a namespace and i is an identifier such that $(i, s) \in \mathcal{N}$ for some s , then $\mathcal{N}.i$ is a fully qualified name of the identifier i that relates i to the definiens s . For example, given a namespace $\mathcal{N}_{\text{physics}} = \{(E, \text{'energy'}), (m, \text{'mass'}), (c, \text{'speed of light'})\}$, $\mathcal{N}_{\text{physics}}.E$ refers to 'energy' – the definiens of E in the namespace 'physics'. Analogous to definitions in programming language namespaces, one can expect that (a) definiens in a given mathematical namespace come from the same area of mathematics, and (b) definiens from different namespaces do not intersect heavily. In other words, one can expect namespaces of mathematical notation to have the same properties as well-designed software packages, namely low coupling and high cohesion.

To precisely define these concepts for mathematical namespaces, we represent them via a document-centric model. Suppose we have a collection of n documents $\mathcal{D} =$

$\left[\begin{array}{c ccc} & d_1 & d_2 & d_3 \\ \hline E & 1 & 0 & 1 \\ m & 1 & 1 & 0 \\ c & 1 & 1 & 0 \end{array} \right]$	$\left[\begin{array}{c ccc} & d_1 & d_2 & d_3 \\ \hline E & 1 & 0 & 1 \\ m & 1 & 1 & 0 \\ c & 1 & 1 & 0 \\ \text{energy} & 1 & 0 & 1 \\ \text{mass} & 1 & 1 & 0 \\ \text{speed of light} & 1 & 1 & 0 \end{array} \right]$
(a) identifier only.	(b) weak association.
$\left[\begin{array}{c ccc} & d_1 & d_2 & d_3 \\ \hline E_energy & 1 & 0 & 1 \\ m_mass & 1 & 1 & 0 \\ c_speed\ of\ light & 1 & 1 & 0 \end{array} \right]$	
(c) strong association.	

Figure 3.5: Illustration of the identifier-document matrix \mathbf{D} for the analyzed methods to create features from the identifiers and definientia, for the mass-energy equivalence example and three hypothetical documents $d_1 = \{E, m, c\}$, $d_2 = \{m, c\}$, $d_3 = \{E\}$.

$\{d_1, \dots, d_n\}$ and a set of K namespaces $\{\mathcal{N}_1, \dots, \mathcal{N}_K\}$. A document d_j can use a namespace \mathcal{N}_k by implicitly importing identifiers from it. Note that real-life scientific documents rarely explicitly use import statements. However, we assume that these implicit namespace imports exist. In this document-centric model, a namespace exhibits low coupling, if only a small subset of documents uses it and high cohesion if all documents in this subset are related to the same domain.

We use the extracted identifier definitions (cf. Section 3.2.3) to discover the namespaces. Since manual discovery of mathematical namespaces is time consuming and error prone, we use machine learning techniques to discover namespaces automatically.

We utilize **clustering methods** to find homogeneous groups of documents within a collection. Comparable to NLP, identifiers can be regarded as ‘words’ in the mathematical language and entire formulae as ‘sentences’. We use cluster analysis techniques developed for text documents represented via the ‘bag-of-words’ model for documents with math formulae that are represented by ‘bag-of-identifiers’. Some definientia are used only once. Since they do not have any discriminative power, they are not very useful and are excluded. Once the identifiers are extracted, we discard the rest of the formula. As a result, we have a ‘bag-of-identifiers’. Analogue to the bag-of-word approach, we only retain the counts of occurrences of identifiers, but do not preserve any structural information.

⑥ Generate feature vectors For clustering, documents are usually represented using the Vector Space Models [4, 177]. We apply the same model, but use identifiers instead of words to represent documents. As the vocabulary, we use a set of identifier-definiens pairs $V = I \otimes F$ which is an element of the vector product space of the identifier space I and the definiens space F . We represent documents as m -dimensional vectors $\mathbf{d}_j = (w_1, \dots, w_m)$, where w_k is the weight of an identifier-definiens pair i_k in the document \mathbf{d}_j and $m = \dim(I) \dim(F)$. We define an identifier-document matrix \mathbf{D} as a matrix where columns represent document vectors and rows represent identifier-document co-occurrences. We evaluate three ways to incorporate the extracted definiens into the model: (1) we use only identifiers without definiens, which reduces the vocabulary to $V_1 = \mathcal{P}_I V$, where the projection operator $\mathcal{P}_I : I \otimes F \rightarrow I$ reduces the dimensions $\dim V_1 = \dim I$; (2) we use ‘weak’ identifier-definiens associations that include identifiers and definiens as separate dimensions, formally $V_2 = \mathcal{P}_{I \oplus F} V$ where the projector $\mathcal{P}_{I \oplus F} : I \otimes F \rightarrow I \oplus F$ reduces the dimension to $\dim V_2 = \dim I + \dim F$; and (3) we use ‘strong’ identifier-definiens associations that append a definiens to each identifier and thus $V_3 = V$.

There is some variability in the definiens: for example, the same identifier σ in one document can be assigned to ‘Cauchy stress tensor’ and in another to ‘stress tensor’ which is almost the same thing. To reduce this variability, we perform the following preprocessing steps: we tokenize the definiens and use individual tokens to index dimensions of the space. For example, suppose we have two pairs $(\sigma, \text{‘Cauchy stress tensor’})$ and $(\sigma, \text{‘stress tensor’})$. In the ‘weak’ association case, we will have dimensions $(\sigma, \text{‘Cauchy’}, \text{‘stress’}, \text{‘tensor’})$, while for the ‘strong’ association we only use the last term, i.e., (σ_tensor) as additional features.

⑦ Cluster feature vectors At this stage, we aim to find clusters of documents that are reasonable namespace candidates. We vectorize each document using the following weighting function $\log(\text{tf})/(z\text{df})$, where tf denotes the term frequency, df the document frequency and z the normalization parameter, such that the length of each document vector is 1. In addition, we discard all identifiers with $\text{DF} < 2$. We further reduce the dimensionality of the resulting dataset via Latent Semantic Analysis (LSA) [58] which is implemented using randomized Singular Value Decomposition (SVD) [88], see [82]. After the dimensionality reduction, we apply Mini-Batch K -Means with cosine distance, since this algorithm showed the best performance in our preliminary experiments (refer to [82] for further details).

⑧ Building namespaces Once a cluster analysis algorithm assigns documents from our collection to clusters, we need to find namespaces among these clusters. We assume that clusters are namespace-defining, meaning that they are not only homogeneous in the cluster analysis sense (e.g., in the case of K -Means it means that the within-cluster sum of squares is minimal), but also contain topically similar documents.

To assess the purity of the clusters, we use the Wikipedia category information which was not used for clustering in the first place. Since each Wikipedia article might have an arbitrary number of categories, we find the most frequent category of the cluster, and thus define its **purity** C as

$$\text{purity}(C) = \frac{\max_i \text{count}(c_i)}{|C|},$$

where the c_i 's are **cluster categories**. Thus, we can select all clusters with purity above a certain threshold and refer to them as namespace-defining clusters. In our experiments, we achieved best results with a threshold of 0.6.

Afterwards, we convert these clusters into namespaces by collecting all identifiers and their definiens in the documents of each cluster. Therefore, we first collect all identifier definiens pairs, and then group them by identifiers. During the extraction, each definiens candidate is scored. This score is used to determine which definiens will be assigned to an identifier in the namespace. We group the pairs by identifier. If an identifier has two or more identical definiens, we merge them into one. Thus, the score of an identifier definiens pair is the sum of scores. There is some lexical variance in the definiens. For example, 'variance' and 'population variance' or 'mean' and 'true mean' are closely related definiens. Thus, it is beneficial to group them to form one definiens. This can be done by fuzzy string matching (or approximate matching) [169]. We group related identifiers and calculate the sum of their scores. Intuitively, the closer a relation is, the higher is the score. A high score increases the confidence that a definiens is correct.

In the last step of our pipeline, we label our namespace defining clusters with categories from well-known classifications, effectively naming the namespaces we identified. We thus achieve two goals. First, we indirectly evaluate our dataset. Second, we ease the use of our dataset to improve MIR. We use the following official classifications:

1. Mathematics Subject Classification (MSC2010) [14] [American Mathematical Society];
2. Physics and Astronomy Classification Scheme (PACS) [15]; and
3. ACM Computing Classification System [192] available as a Simple Knowledge Organization System (SKOS) ontology [153].

We processed the SKOS ontology graph with RDFLib. All categories can be found on our website [200]. After obtaining and processing the data, the three classifications are merged into one. We map namespaces to second-level categories by keyword matching. First, we extract all keywords from the category. The keywords include the top level category name, the subcategory name and all third level category names. From each namespace, we extract the namespace category and names of the articles that form the namespace. Finally, we perform a keyword matching, and compute the

cosine similarity between the cluster and each category. The namespace is assigned to the category with the largest cosine score. If the cosine score is below 0.2 or only one keyword is matched, the cluster is assigned to the category ‘others’.

Improve identifier definition extraction We used POS Tagging-based distance measures (cf. Section 3.2.3) to extract identifier definiens pairs from the text surrounding the formula. In a second step, we build namespaces of identifiers. These namespaces allow us to study the usage of identifiers in different scientific fields. Many, but not all definiens can be found in the text surrounding the formulae. Thus, the namespaces can additionally be used to identify the definiens in cases where the definiens is not mentioned in the text.

3.2.3.2 Implementation Details

We use the Big Data framework Apache Flink, which is capable of processing our datasets in a distributed shared nothing environment, leading to short processing times. Our source code, training, and testing data is openly available from our website [200].

For the MLP part, our implementation follows the open source implementation of the Mathematical Language Processing Project [180] (cf. Section 3.2.2), with the following improvements. Rather than converting the Wikipedia formulae via LaTeXXML, we now directly extract the identifiers from the LaTeX parse tree via mathoid [207] (cf. Section 3.3). Second, we include a link to Wikidata, so that Wikipedia links can be replaced by unique and language independent Wikidata identifiers (ids). These ids are associated with semantic concepts which include a title, and in many cases a short description that simplifies disambiguation. For the POS Tagging, we use the Stanford Core NLP library (Stanford NLP) [143] for POS Tagging of natural language as well as additional math-aware tags (cf. Section 3.2.3). In summary, we use the following tags:

1. Identifiers (‘ID’);
2. Formulae (‘MATH’);
3. Inner Wiki link (‘LINK’);
4. Singular noun (‘NN’);
5. Plural noun (‘NNS’);
6. Adjective (‘JJ’); and
7. Noun phrase (‘NOUN_PHRASE’).

For the Namespace Discovery step in our pipeline (Section 3.2.3.1), we use the following implementation to discover clusters that are suitable namespace candidates. Using ‘TfidfVectorizer’ from scikit-learn [182], we vectorize each document. The experiments are performed with $(\log \text{TF}) \times \text{IDF}$ weighting. Therefore, we use the following parameters: ‘use_idf=False’, ‘sublinear_tf=True’. Additionally, we discard identifiers that occur only once by setting ‘min_df=2’. The output of ‘TfidfVectorizer’ is row-normalized, i.e., all rows have unit length.

The implementation of randomized SVD is taken from [182] – method ‘randomized_svd’. After dimensionality reduction, we apply Mini-Batch K -Means (class ‘MiniBatchKMeans’) from [182] with cosine distance. In our preliminary experiments, this algorithm showed the best performance. To implement it, we use the Python library FuzzyWuzzy. Using fuzzy matching we group related identifiers and then sum over their scores.

3.2.4 Evaluation

3.2.4.1 Dataset

As test collection, we use the collection of Wikipedia articles from the NII Testbeds and Community for Information access Research (NTCIR) 11 Math Wikipedia Task (WMC) [208] (cf. Section 4.1) in 2014. We choose this collection instead of the latest version of Wikipedia to be able to compare our results to previous experiments.

After completing the MLP pipeline, we exclude all documents containing less than two identifiers. This procedure results in 22 515 documents with 12 771 distinct identifiers that occur about 2 million times. Figure 3.6 shows that identifiers follow a power law distribution, with about 3 700 identifiers occurring only once and 1 950 identifiers occurring only twice.

The amount of identifiers per document also appears to follow a long tail power law distribution ($p < 0.001$ for KS test) as only a few articles contain a lot of identifiers, while most of the articles do not. The largest number of identifiers in a single document is an article with 22 766 identifiers, the second largest has only 6 500 identifiers. The mean number of identifiers per document is 33. The distribution of the number of distinct identifiers per document is less skewed than the distribution of all identifiers. The largest number of distinct identifiers in a single document is 287 (in the article ‘Hooke’s law’), followed by 194 (in the article ‘Dimensionless quantity’). The median of identifiers per document is 10. For 12 771 identifiers, the algorithm extracted 115 300 definitia. The number of found definitia follows a long tail distribution as well, with the median of definitia per page being 4. Moreover, we list the most common identifier -definiens pairs in Figure 3.6.

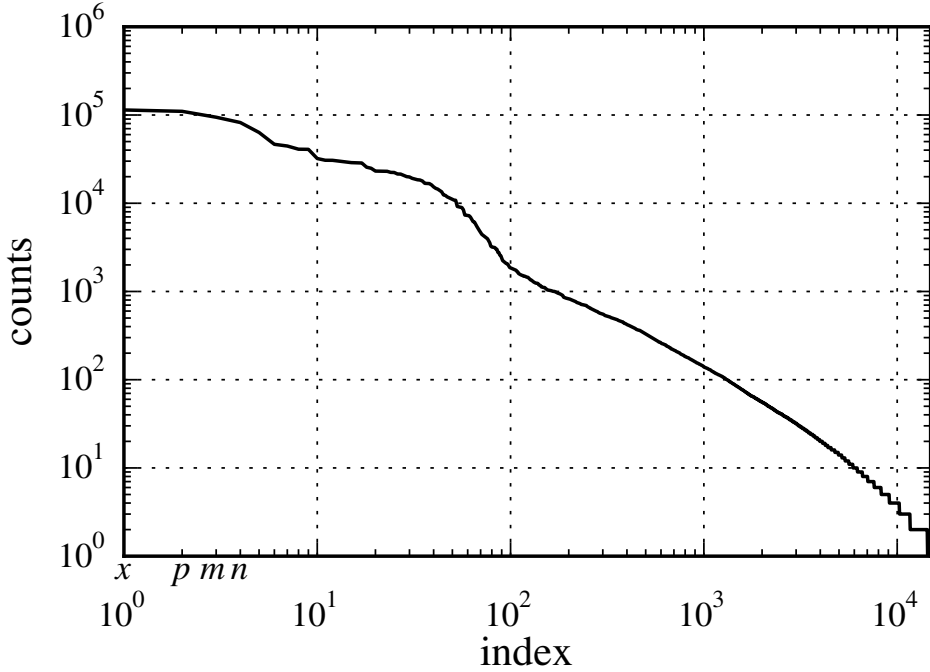


Figure 3.6: Distribution of identifier counts. The most frequent identifiers are x (125k), p (110k), m (105k), and n (83k).

3.2.4.2 Gold Standard

We created a gold standard from the 100 formulae patterns included in the NTCIR 11 Wikipedia task [208] (cf. Section 4.1) and the following information:

1. Identifiers within the formula;
2. Definitions of each identifier; and
3. Links to semantic concepts on Wikidata.

We compared our results with that gold standard and calculated the three measures: precision, recall, and F1-score, to evaluate the quality of our identifier definitions. In a first step, we evaluated the results acquired with the POS Tagging based distance measures (see Section 3.2.3). In a second step, we evaluated the results acquired by combining the POS Tagging based distance measures and the results of the namespaces (see Section 3.2.3.1)

The gold standard (cf. Figure 3.7) consists of 310 identifiers, with a maximum of 14

identifiers per formula. For 174 of those identifiers, we could assign the corresponding semantic concept in Wikidata. For 97, we assigned an individual phrase that we could not relate to a Wikidata concept. For an additional 27, we assigned two phrases. For example, for Topic 32 (cf. Figure 3.7), we assigned critical temperature in addition to the semantic concept of the critical point, since the critical temperature is more specific. The full list of assignments is available from our website [200]. Note, that the identification of the correct identifier definition, was very time consuming. For several cases, the process took more than 30 minutes per formulae, since multiple Wikipedia pages and tertiary literature had to be consumed. The gold standard was checked by a mathematician from the Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, Maryland, USA.

3.2.5 Results

In this section, we describe the results of our evaluation. First, we describe the quality of the MLP process in Section 3.2.5.1. Afterwards, we describe the dataset statistics and the results of the namespace evaluation in Section 3.2.5.2.

- (1) Van der Waerden's theorem: $W(2, k) > 2^k / k^\varepsilon$

W **Van der Waerden number**

k **integer** : number that can be written without a fractional or decimal component

ε **positive number** (real number...)

...

- (31) Modigliani-Miller theorem: T_c

T_c **tax rate** : ratio (usually expressed as a percentage) at which a business or person is taxed

- (32) Proximity effect (superconductivity): T_c

T_c critical temperature, **critical point** : critical point where phase boundaries disappear

...

- (69) Engine efficiency: $\eta = \frac{\text{work done}}{\text{heat absorbed}} = \frac{Q_1 - Q_2}{Q_1}$

η **energy efficiency**

Q_1 **heat** (energy)

Q_2 **heat** (energy)

...

- (86) Lagrangian mechanics: $\frac{\partial L}{\partial q_i} = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i}$

L **Lagrangian**

q_i **generalized coordinates**

t **time** (...)

\dot{q}_i generalized velocities, **generalized coordinates**

Figure 3.7: Selected entries from the gold standard. Bold font indicates that the entry is linked to a language-independent semantic concept in Wikidata. The descriptions in brackets originate from the English Wikidata label and have been cropped to optimize the layout of this figure.

Table 3.6: Identifier definitions for selected identifiers and namespaces extracted from the English Wikipedia, the accumulated score s and the human relevance rankings confirmed (🟢), partly confirmed (🟡), not sure (🟡) and incorrect (🔴). Discovered semantic concepts are printed using bold font. The descriptions were fetched from Wikidata. To improve readability of the table, we manually shortened some long description texts.

Classical mechanics of discrete systems 45.00 (PACS)		
Categories: Physics, Mechanics, Classical mechanics		
Purity: 61%, matching score: 31%, identifiers 103, semantic concepts 50, 🟢 58,🟡 4, 🟡 42, 🔴 1		
Identifier-definitions:		
m mass (quantitative measure of a physical object's resistance to acceleration by a force ...)	$[s \approx 29]$	🟢
F force (influence that causes an object to change)	$[s \approx 25]$	🟢
v velocity (rate of change of the position of an object... and the direction of that change)	$[s \approx 24]$	🟢
t time (dimension in which events can be ordered the past through the present into the future)	$[s \approx 19]$	🟢
a acceleration (rate at which the velocity...)	$[s \approx 17]$	🟢
r position (Euclidean vector ...)	$[s \approx 14]$	🟢
i particle	$[s \approx 12]$	🟡
E energy (physical quantity representing the capacity to do work)	$[s \approx 11]$	🟢
v speed (magnitude of velocity)	$[s \approx 10]$	🟢
a acceleration	$[s \approx 10]$	🟢
V velocity	$[s \approx 9]$	🟡
u flow velocity	$[s \approx 8]$	🟢
r radius	$[s \approx 8]$	🟡
...		
E electric field (... representing the force applied to a charged test particle)	$[s \approx 6]$	🟢
...		
c speed of light (speed at which all massless particles and associated fields travel in vacuum)	$[s \approx 3]$	🟢

Table 3.6: Identifier definitions for selected identifiers and namespaces extracted from the English Wikipedia, the accumulated score s and the human relevance rankings confirmed (🟢), partly confirmed (🟡), not sure (?) and incorrect (🔴). Discovered semantic concepts are printed using bold font. The descriptions were fetched from Wikidata. To improve readability of the table, we manually shortened some long description texts.

Stochastic analysis 60Hxx (MSC) Categories: Stochastic processes, Probability theory Purity: 92%, matching score: 62%, identifiers 54, semantic concepts 32, 🟢 18, 🟡 0, ? 30, 🔴 0 Identifier-definitions:	
a stochastic process (... random variables)	$[s \approx 12]$ 🟢
X stochastic process (... random variables)	$[s \approx 10]$ 🟢
...	
E expected value	$[s \approx 2]$ 🟢
...	
\mathbb{E} expected value $s < 1$	
v function $s < 1$	
Theory of data 68Pxx (MSC) Categories: Information theory, Theoretical computer science Purity: 86%, matching score: 35%, identifiers 58, semantic concepts 10 Identifier-definitions:	
R rate	$[s \approx 12]$ 🟢
X posterior probability	$[s \approx 10]$ 🟢
n length	$[s \approx 8]$ 🟢
...	
H Information entropy (expected value of the amount of information delivered by a message)	$[s \approx 5]$ 🟢
I mutual information	$[s \approx 5]$ 🟢
a program	$[s \approx 5]$ 🟢
...	
a codeword $s < 1$	
\mathbb{E}_X expected value $s < 1$	

3.2.5.1 Mathematical Language Processing

Identifier extraction Our gold standard consists of 310 identifiers to be extracted from the aforementioned 100 reference formulae. We were able to extract 294 identifiers (recall 94.8%) from the gold standard correctly. We obtained only 16 false negatives, but overall 57 false positives (precision 83.7%, F_1 89.0%). Falsely detected identifiers affect 22% of the reference formulae, showing that often several falsely extracted identifiers belong to one formula. In the following, we explain why the errors can be attributed to the shortcomings of the heuristics explained in Section 3.2.3.

Incorrect markup. Errors relating to 8 formulae (33 false positive and 8 false negative identifiers) were caused by the incorrect use of LaTeX, especially the use of math mode for text or the missing usage of math mode for part of the formula. An identifier Q_1 that is falsely marked as $Q1$ (cf. Figure 3.7, Topic 69) in a formula, can easily be identified correctly by a human since it looks very similar in the output. As obviously Q_1 is meant in the formula, we took Q_1 as gold standard for this identifier. But in the MLP process it is impossible to extract the identifier correctly, as $Q1$ implies Q times 1.

Symbols mathoid. For 8 formulae (9 false positive identifiers), mathoid [207] (cf. Section 3.3) misclassified symbols as identifiers, such as d in $\frac{d}{dx}$. Two formulae (2 false positive identifiers) are substitutions (abbreviations that improve the readability of formulae without specific meaning).

Sub-super-script. Two formulae (3 false positive, 2 false negative identifiers), used sub- super-script such as σ_y^2 .

Special notation. For 2 formulae (10 false positive, 2 false negative identifiers), use special notation like the Einstein sum convention.

We excluded incorrectly extracted identifiers from the following processing steps. Thus, the upper bound for recall and precision are set by the identifier extraction step.

Definition extraction In a first step, we only assess the definitions that matched exactly the semantic concepts materialized as Wikidata item in the gold standard. Thus, we found 88 exact matches (recall 28.4%), but also obtained 337 false negatives which results in a precision of 20.7% (F_1 23.9%).

In addition, we evaluated the performance of partially relevant matches by manually deciding the relevance for each entry. For example, **integer** (number that can be written without a fractional or decimal component) would be classified as highly relevant, but the string **integers** was classified as relevant. Although this classification is mathematically incorrect, it provides valuable information for a human regarding the formulae. With this evaluation, we obtain 208 matches (recall 67.1%) and 217 false negatives (precision 48.9%, F_1 56.6%). To interpret these results, we differentiate

between definitions that have not been extracted, although all necessary information is present in the information source, and definitions that do not completely exist in the information source. Wolska and Grigore [246] found that around 70% of objects denoting symbolic expressions are explicitly denoted in scientific papers. Since in our data source only 73% of the identifiers are explained in the text, 73% represents the highest achievable recall for systems that do not use world knowledge to deduce the most likely meaning of the remaining identifiers. Considering this upper limit, we view a recall of 67.1% that was achieved when including partly relevant results, as a good result. These results also confirm the findings of Kristianto et al. [121]. Although these overall results match with the results of Wolska and Grigore [246], we found major differences between different scientific fields. In pure mathematics, the identifiers usually do not link to a specific concept and the formulae do not relate to specific real-life-scenarios. In contrast, in physics the definitia of the identifiers are usually mentioned in the surrounding text, like in the mass-energy-equivalence example.

3.2.5.2 Namespace Discovery

The evaluation of the namespace discovery performance is twofold. First, we apply the same procedure as in the evaluation of the MLP process. In a second step, we perform a manual quality assessment of the final namespaces.

We obtain the following results with regard to the extraction performance. For the strict relevance criterion, the recall improved by 18% (0.048) to 33.2% (103 exactly correct definitions), and the precision declined only slightly with 420 false positives to 19.7% (F_1 24.7%). In the end, 30 identifiers (9.6%) reached the ultimate goal and were identified as a semantic concept on Wikidata. For the non-strict relevant criterion, we could measure a recall performance gain of 19.4%, while maintaining the precision level. This exceeds the upper limit for recall achievable by exclusively analyzing the text of a single document of (73%) and extracts 250 definitions correctly (recall 80.6%) with only 273 false positives (precision 47.8%, F_1 60.0%).

The second part of the evaluation assesses the quality of the discovered namespaces. While a detailed performance evaluation of the clustering methods was already carried out in [82], we focus on the contents of the discovered namespaces here. For evaluating the Namespace Discovery, we evaluated six randomly sampled subject classes. Two independent judges rated the categorized identifier definiens pairs regarding their assignment to subject classes using the four categories: ‘confirmed’ (✔), ‘partly confirmed’ (✓), ‘not sure’ (?) and ‘incorrect’ (✗), regarding their assignment to the subject class by two independent raters. All cases of disagreement (mostly ? vs. ✔) could be resolved in consensus.

With strong coupling and a minimal purity of 0.6, 250 clusters were obtained of which 167 could be mapped to namespaces in the official classification schemes (MSC 135, PACS 22, ACM 8). The purity distribution is as follows: 0.6- 0.7: 98, 0.7- 0.8: 57, 0.8-

0.9: 44, 0.9- 1.0: 51.

Those namespaces contain 5 618 definitions with an overall score >1 , of which 2 124 (37.8%) link to semantic concepts. We evaluated the recall of 6 discovered namespaces exemplary. The purity of the selected namespaces ranged from 0.6 to 1 with an average of 0.8. They contained between 14 and 103 identifiers (with a score >1). Here, relevance means that the definition is commonly used in that field. This was decided by domain experts. However, since this question is not always trivial to judge, we introduced an unknown response (🔍). In total, 129 (43%) of the 278 discovered definitions matched the expectation (🟢) of the implicit namespaces expected by the domain experts. For 7 definitions (3%), they were clearly wrong (🔴), for 8 (3%), the definitor was not specific enough and for the remaining 144 (52%), the reviewers could not assess the relevance (🔍). Note that the quality of namespaces varied. For example, cluster (33Cxx, Hypergeometric functions) had significantly more clearly wrong results, because symbols were classified as identifiers, compared to the investigated clusters in physics where the definition of specific symbols is less common.

In general, this result was expected, since it is hard to assess the namespaces that have not been spelled out explicitly before. Especially, the recall could not be evaluated, since to the best of our knowledge, there is no reference list with typical identifiers in a specific mathematical field. For details regarding implementation choices, visit our website [200], browse the namespaces, investigate our gold standard, and contribute to our open source software Mathosphere.

3.2.6 Conclusion and Outlook

We investigated the semantification of identifiers in mathematics based on the NTCIR 11 Math Wikipedia test collection using MLP and Namespace Discovery. Previous approaches have already shown good performance in extracting descriptions for mathematical formulae from the surrounding text in individual documents.

We achieved even better performance (80% recall, while maintaining the same level of precision) in the extraction of relevant identifier definitions. In cases where identifier definitions were absent in the document, we used our fall back mechanism of identifier definitions from the namespace that we learned from the collection at large. This way, we could break the theoretical limit for systems (about 70% recall cf. Section 3.2.5) that take into account only one document at a time. Moreover, the descriptions extracted by other systems are language dependent and do not have a specific data structure.

In contrast, we organized our extracted identifier definitions in a hierarchical data structure (i.e., namespaces) which simplifies subsequent data processing tasks such as exploitative data analysis.

For about 10% of the identifiers, we were able to assign the correct semantic concept on

the collaborative knowledge base Wikidata. Note, that this allowed extracting even more semantics beside a natural language description as spelled out in Table 3.6. Namely, one can find labels and descriptions in multiple languages, links to relevant Wikipedia articles in different languages, as well as statements. For example, for the identifier **speed of light**, 100 translations exist. As statements one can, for example, retrieve the numeric value ($3 \times 10^8 \text{m/s}$), and the fact that the speed of light is a unit of measurement. We observed that identifier clusters in physics and computer science are more useful in the sense that they more often link to real-world semantic objects than identifier clusters in pure mathematics, which often solely specify the type of the variable.

During the construction of the gold standard, we noticed that even experienced mathematicians often require much time to manually gather this kind of semantic information for identifiers. We assume that a significant percentage of the 500 million visitors to Wikipedia every day face similar problems. Our approach is a first step to facilitate this task for users.

The largest obstacle for obtaining semantic information for identifiers from Wikidata is the quality of the Wikidata resource itself. For 44% of the identifiers in the gold standard, Wikidata contains only rather unspecific hypernyms for the semantic concept expressed by the identifier. We see two options to remedy this problem in future research. The first option is to use a different semantic net containing more fine-grained semantic concepts. The second option is to identify unspecific semantic concepts in Wikidata and to split them into more specific Wikidata items related to mathematics and science.

Our identifier extraction has been rolled out to the Wikimedia production environment. However, at the time of writing, incorrect markup is still a major source of errors. To overcome this problem, the implementation of procedures that recognize and highlight incorrect markup for Wikipedia editors is scheduled and will encourage editors to improve the markup quality. In addition, symbols falsely classified as identifiers have a noticeable negative impact on the quality of the clustering step. Improving the recognition of symbols is therefore an issue that future research should address. Moreover, in the future our method should be expanded to other datasets beside Wikipedia.

With regard to MIR applications i.e., math search, we have shown that the discovered namespaces can be used to disambiguate identifiers. Exposing namespaces to users is one application of identifier namespaces. Using them as internal data structure for math information retrieval applications, such as math search, math understanding or academic plagiarism detection is another. Regarding MIR tasks, identifier namespaces allow for quiz like topics such as ‘At constant temperature, is volume directly or inversely related to pressure?’. This simplifies comparing traditional word based question and answering systems with math aware methods.

In conclusion, we regard our namespace concept as a significant innovation which will

allow users to better express their mathematical INs, and search engines to disambiguate identifiers according to their semantics. However, more research needs to be done to better understand the influence of each individual augmentation step of our presented pipeline for MIR applications.

This opens a large field for future research opportunities. For instance in “Getting the units right” by Schubotz, Veenhuis, and Cohl published in 2016 [206], we present our vision to check the dimensionality of physical formulae automatically.

3.3 Augmenting Presentation of Mathematical Expressions

Wikipedia is the first address for scientists who want to recap basic mathematical and physical laws and concepts. Today, formulae in those pages are displayed as Portable Network Graphics (PNG). Those images do not integrate well into the text, can not be edited after copying, are inaccessible to screen readers for people with special needs, do not support line breaks for small screens and do not scale for high resolution devices. Mathoid improves this situation and converts formulae specified by Wikipedia editors in a \TeX -like input format to Mathematical Markup Language, with Scalable Vector Graphics (SVG)s as a fallback solution.

3.3.1 Introduction: Browsers are Becoming Smarter

Wikipedia has supported mathematical content since 2003. Formulae are entered in a \TeX -like notation and rendered by a program called `texvc`. One of the first versions of `texvc` announced the future of MathML support as follows:

‘As of January 2003, we have \TeX markup for mathematical formulas on Wikipedia. It generates either PNGs or simple HTML markup, depending on user prefs and the complexity of the expression. In the future, as more browsers are smarter, it will be able to generate enhanced HTML or even MathML in many cases.’ [145]

Today, more than 10 years later, less than 20% of people visiting the English Wikipedia, currently use browsers that support MathML (e.g., Firefox) [258]. In addition, `texvc`, the program that controls math rendering, has made little progress in supporting MathML. Even in 2011, the MathML support was ‘rather pathetic’ (see [171]). As a result, users expected MathML support within Wikipedia to be a broken feature. Ultimately, on November 28, 2011, the user preference for displaying MathML was removed [235].

Annoyed by the PNGs, in December 2010, user Nageh published a script, User:Nageh/mathJax.js, that enables client-side MathJax rendering for individual Wikipedia users [233]. Some effort and technical expertise was required to use the script. The user had to install additional fonts on his system manually, to import the script, into his Wikipedia account settings and to change the math setting in his Wikipedia user account page to ‘Leave it as \TeX ’. With client-side MathJax rendering, the visitor was able to choose from the context menu of each equation with the PNG being replaced by either:

- A SVG,
- An equivalent HTML + CSS representation, or
- MathML markup (this requires a MathML capable browser).

Depended on operating system, browser and hardware profile MathJax needs a significant amount of time to replace the \TeX -code on the page with the replacements. For instance, we measured 133.06s to load the page **Fourier transform** in client side MathJax mode, as compared to 14.05s for the page loading without math rendering (and 42.9s with PNGs) on a typical Windows laptop with Firefox.³

However, improvements in the layout motivated many users to use that script, and in November 2011, client-side MathJax became an experimental option for registered users [236].

As of today, there are almost 21M registered Wikipedia users, of which 130k have been active in the last 30 days. Of these users, 7.8k use the MathJax rendering option which causes long waiting times for pages with many equations. Also 47.6k users chose the (currently disabled) HTML rendering mode, which if possible, tries to use HTML markup to display formula, and the PNG otherwise. Furthermore, 10.1k users chose the MathML rendering mode (disabled in 2011). Thus, the latter 57.7k users are temporarily forced to view PNGs, even though they explicitly requested against this. This demonstrates that there is an significant demand for math rendering, other than for the use of PNGs.

Currently, the MediaWiki Math extension is version 1.0. Our efforts have been to make an improvement on that extension. We refer to our update of the extension as version 2.0. Furthermore, we refer to Mathoid as all the ingredients mentioned in this section which go into developing Math 2.0. One essential ingredient in Mathoid, is what we refer to as Mathoid server. This is a tool, which we describe in this section, which converts the \TeX -like math input used in MediaWiki to various formats that we describe in this section. This section is organized as follows:

In Section 3.3.2, we list the requirements for math rendering in Wikipedia, explain how one may contribute to these requirements, and elaborate on how one may make math

³The measurement was done on a Lenovo T420 Laptop with the following hardware: 8GB RAM, 500GB HDD, CPU Intel Core i7-2640M, Firefox 24.0 on Windows 8.1, download speed 98.7 MB/s upload speed 9.8 MB/s, ping to en.wikipedia.org was 25(\pm 1)ms.

accessible for people with special needs. In this section, we introduce the Mathoid server. In Section 3.3.3, we explain how by using Mathoid server, math can be displayed in browsers that do not support MathML. In Section 3.3.4, we discuss how math rendering can be offered as a globally distributed service. In Section 3.3.5, we discuss and compare the performance of reviewed rendering tools, in regard to layout and speed. Finally, in Section 3.3.6, we conclude with results from our comparison and give an overview of future work.

3.3.2 Bringing MathML to Wikipedia

For Wikipedia, the following requirements and performance measures are critical.

Coverage: The converter must support all commands currently used in Wikipedia.

Scalability: The load for the converter may vary significantly, since the number of Wikipedia edits heavily depends on the language. Thus, a converter must be applicable for both small and large Wikipedia instances.

Robustness: Bad user input, or a large number of concurrent requests, must not lead to permanent failure for the converter.

Speed: Fast conversion is desirable for a good user experience.

Maintainability: A new tool for a global site of the size of Wikipedia must be able to handle tasks with a large management overhead. Therefore, active development over a long period of time is desirable.

Accessibility: Providing accessible content to everyone is one of the key goals of Wikipedia.

There is a large variety of \TeX to MathML converters [36]. However, most of them are no longer under active development, or their licenses are not compatible with MediaWiki. In 2009, [221] showed that LaTeXXML has the best coverage (but not a very high conversion speed) as compared to the LaTeX converters which were analyzed in that paper. Since 2009, a new converter, MathJax [45], has become popular. After negotiations with Peter Krautzberger (of MathJax) and his team, we regard MathJax (executed in a headless browser on a web server), as a useful alternative to LaTeXXML. One strong point about MathJax with regard to coverage, is that it is already used by some Wikipedia users on the client-side (as described in Section 3.3.1). Therefore the risk of unexpected behavior is limited. For LaTeXXML, Bruce Miller has written a set of custom macros for MediaWiki specific \TeX commands which supports the MediaWiki \TeX markup. A test using these macros based on a Wikipedia dump, has shown very good coverage of the mathematics commands currently used in Wikipedia. LaTeXXML is maintained by the United States Department of Commerce Laboratory, the National Institute of Standards and Technology (NIST) (NIST) and

MathJax is maintained by the MathJax project which is a consortium of the American Mathematical Society and the Society for Industrial and Applied Mathematics. We analyze and compare MathJax and LaTeXML in detail, since these are the most promising tools we could discover.

In regard to accessibility, we note that Wikipedia has recently made serious efforts to make the site more accessible. However, images which represent equations are currently not accessible. The only available information from PNGs (which is not very helpful) is the alt-text of the image that contains the \TeX code. Based upon recent unpublished work of Volker Sorge [220], we would like to provide meaningful semantic information for the equations. By providing this, more people will be able to understand the (openly accessible) content [50]. One must also consider that there is a large variety of special needs. People with blindness are only a small fraction of the target group which can benefit from increased accessible mathematical content. Between 1 and 33% [53, 54] of the population suffer from dyslexia. Even if we calculate with the lower boundary of 1%, 90,000 people per hour visit the English Wikipedia and some of them could benefit from improvements of the accessibility while reading articles that contain math. However, our main goal with regard to accessibility is to make Wikipedia accessible for blind people that have no opportunity to read the formulae in Wikipedia today.

Furthermore, the information provided in the tree structure of mathematics by using MathML, helps one to orientate complex mathematical equations which is useful for general purpose use as well. With regard to accessibility, a screen reader can repeat only part of an equation to provide details that were not understood. PNG do not give screen readers detailed related mathematical information [47, 159]. In 2007, [141] states that MathML is optimal for screen readers. **The Faculty Room**⁴ website, lists four screen readers that can be used in combination with MathPlayer [217] to read Mathematical equations. Thus, Mathoid server and LaTeXML server [74] that generate MathML output, contribute towards better accessibility within the English Wikipedia.

3.3.3 Making Math Accessible to MathML Disabled Browsers

For people with browsers that do not support MathML, we would like to provide high quality images as a fallback solution. Both LaTeXML and MathJax provide options to produce images. LaTeXML supports PNGs only, which tend to look rasterized, if they are viewed using large screens. MathJax produces SVGs. For high traffic sites like Wikipedia with 9 million visitors per hour, it is crucial to reduce the server load generated by each visitor. Therefore rendered pages should be used for multiple visitors. Rendering of math elements is especially costly. This is related to the nested structure of mathematical expressions. As a result, we have taken care that the output of the MediaWiki math extension should be browser independent.

⁴<http://www.washington.edu/doit/Faculty/articles?404>

Since MathJax was designed for client-side rendering, our goal is to develop a new component. We call this new component the **mathoid server**. Mathoid server, a tool written in JavaScript, uses MathJax to convert math input to SVGs. It is based on svgtex [1] which uses nodejs and phantomjs to run MathJax in a headless browser. It exports SVG. Mathoid server improves upon the functionality of svgtex while offering a more robust alternative. For instance, it provides a restful interface which supports json input and output as well as the support of MathML output. Furthermore, Mathoid server is shipped as a Debian package for easy installation and maintenance. Many new developments in MediaWiki use JavaScript. This increases the probability of finding volunteers to maintain the code and to fix bugs. For general purpose, Mathoid server can be used as a stand-alone service which can be used in other content management platforms such as Drupal or Wordpress. This implies that Mathoid server will have a larger interest group for maintenance in the future. The fact that Mathoid server automatically adapts to the number of available processing cores, and can be installed fully unattended via tools like Puppet, indicates that the administrative overhead for Mathoid server instances should be independent of the number of machines used. In the latest version, the Mathoid server supports both LaTeX and MathML input and is capable of producing MathML and SVG output.

To support MathML disabled browsers, we deliver both MathML markup, and a link to the SVG fallback image, to the visitor's browser. In order to be compatible with browsers that do not support SVG, in addition, we add links to the old PNGs. In the future those browsers will disappear and this feature will be removed.

To prevent users from seeing both rendering outputs, the MathML element is hidden by default, and the image is displayed. For Mozilla based browsers (these support MathML rendering), we invert the visibility by using a custom CSS style, hide the fallback images and display the MathML markup. This has several advantages. First, no browser detection, neither on the client-side (e.g., via JavaScript) nor on server-side is required. This eliminates a potential source of errors. Our experiments with client-side browser detection showed that the user will observe a change in the math elements if pages with many equations are loaded. Second, since the MathML element is always available on the client-side, the user can copy equations from the page, and edit it visually with tools such as Mathematica. If the page content is saved to disk, all information is preserved without resolving links to images. If afterwards the saved file is opened with a MathML enabled browser, the equations can be viewed off-line. This feature is less relevant for users with continuous network connections or with high-end hardware and software. However, for people with limited resources and unstable connections (like in some parts of India [17]), they will experience a significant benefit.

The current⁵ Firefox mobile version (28.0) passes the MathML Acid-2 test, indicating that there is generally good support for MathML on mobile devices. This allows for customized high quality adaptable rendering for specific device properties. The W3C

⁵as of 2014

MathML specification⁶ discusses the so called best-fit algorithm for line breaking. According to our experiments, Firefox-mobile (28.0) does not pass the W3C line break tests. However, as soon this issue is fixed, mobile users will benefit from the adjusted rendering for their devices. Note that there is active development in this area by the Mathematics in ebooks project⁷ lead by Frédéric Wang.

3.3.4 A Global Distributed Service for Math Rendering

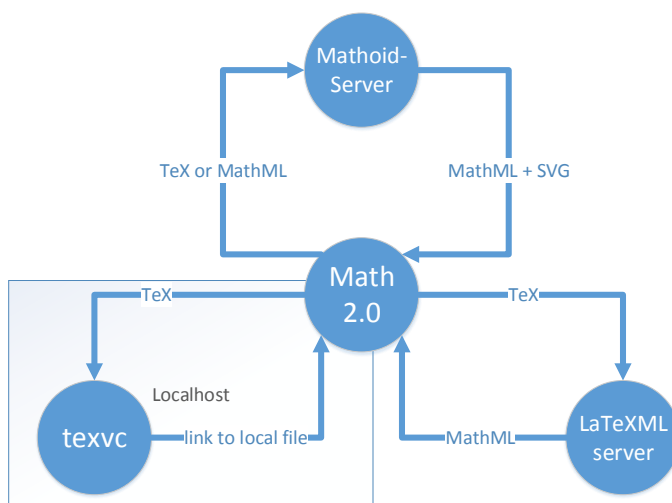


Figure 3.8: System overview. The 2.0 release of the MediaWiki Math extension offers new ways to render math input in MediaWiki. It communicates to LaTeXML servers and to instances of our MathJax- based development of the Mathoid server. Note that we preserve backwards compatibility to the MediaWiki Math extension 1.0.

To use LaTeXML and Mathoid server for math rendering within Wikipedia, we have changed the MediaWiki Math extension (see Fig. 3.8). While preserving backwards compatibility, we pursue our current development [202] by integrating LaTeXML and Mathoid server. System administrators can choose which rendering back-ends are selectable in a MediaWiki instance by logged in users. All rendering modes can be active at the same time, and it is possible to use Mathoid server to generate SVG based on the output of the LaTeXML server.

For texvc, rendering requires one to install a full LaTeX distribution (about 1GB) on each web server. This is a huge administrative overhead and the management of files and permissions has caused a lot of problems. These problems were hard to solve and

⁶<http://www.w3.org/TR/MathML/chapter3.html>

⁷<http://www.ulule.com/mathematics-ebooks>

resulted in inconsistent behavior of the website from a user perspective [147, 148, 165, 168]. Furthermore, for MediaWiki instances run by individuals, it is difficult to set up math support. Therefore, one of our major requirements is that future versions of the MediaWiki Math extension should not need to access the file system. Also, one should not need to use shell commands to directly access the server. This has the potential to introduce major security risks. With our separation approach, rendering and displaying of mathematics no longer needs to be done on the same machine. However, for instances with a small load, this would still be possible. Furthermore, small MediaWiki instances can now enable math support without any configuration or shell access. By default, public LaTeXML and Mathoid server instances are used. With this method, no additional security risk is provided for individuals. For Mathoid server, the security risk for the host is limited as well. This is because the Mathoid process runs on a headless browser on the server without direct access to the file system.

3.3.4.1 Caching

There are two main caching layers. In the first layer, the database caches the rendering result of the math conversion, i.e., the MathML and SVG output in the database.⁸ The second caching layer is browser based. Similar to the SVG rendering process for ordinary SVG, the MediaWiki Math extension uses cacheable special pages to deliver SVG. On the server side, those pages are cached by squid servers. In addition, even if images occur on different pages, the browser will only load that image once.

3.3.5 Performance Analysis

As a first step towards a visual analysis, we compared our impressions of output from LaTeXML and MathJax using Firefox 24.0. Except for additional mrow elements in LaTeXML, the produced Presentation MathML is almost identical. The differences that we did notice had no influence on the rendering output. In rare cases, MathJax uses `mi` elements, whereas LaTeXML uses `mo` elements. In contrast to LaTeXML which uses UTF-8 characters to represent special symbols, MathJax uses HTML entities. However, there still remain some minor differences (see Fig. 3.9).

To illustrate performance differences, we chose a random sample equation, namely

$$\frac{(x-h)^2}{a^2} - \frac{(y-k)^2}{b^2} = 1. \quad (3.2)$$

With 10 subsequent runs⁹, the following conversion times were measured:

⁸To keep the lookup time for equations constant, the key for the cache entry is a hash of \TeX input.

⁹All measurements were performed using virtual Wikimedia labs instances, with the following hardware specifications: number of CPUs 2, RAM size 4096 Mb, allocated Storage 40 Gb

texvc	LaTeXML	LaTeXML-SVG	Mathoid	Mathoid-SVG
$(\begin{array}{cc} a & b \\ c & d \end{array})$				
\overline{abc}	\overline{abc}	\overline{abc}	\overline{abc}	\overline{abc}
$\operatorname{d}t$				
$\mathfrak{U}, \mathfrak{U}$	$\mathfrak{U}, \mathfrak{U}$	$\mathfrak{U}, \mathfrak{U}$	$\mathfrak{U}, \mathfrak{U}$	$\mathfrak{U}, \mathfrak{U}$
Ω_3^4				
\mathcal{ABC}	\mathcal{ABC}	\mathcal{ABC}	\mathcal{ABC}	\mathcal{ABC}
$a \, b, a \, b$				
$\begin{array}{ c c } \hline a & b \\ \hline 0 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$	$\begin{array}{cc} a & b \\ 0 & 0 \\ 0 & 1 \end{array}$	$\begin{array}{cc} a & b \\ 0 & 0 \\ 0 & 1 \end{array}$	$\begin{array}{ c c } \hline a & b \\ \hline 0 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$	$\begin{array}{ c c } \hline a & b \\ \hline 0 & 0 \\ \hline 0 & 1 \\ \hline \end{array}$
$\frac{123}{456}$				
$\mathbf{123}$	$\mathbf{123}$	$\mathbf{123}$	$\mathbf{123}$	$\mathbf{123}$
$\big \backslash \Big \backslash \bigg \backslash \Bigg \backslash \dots \Bigg \backslash \bigg \backslash \Big \backslash \big \backslash$				
$\big / \Big / \bigg / \Bigg / \dots \Bigg / \bigg / \Big / \big /$	$\big / \Big / \bigg / \Bigg / \dots \Bigg / \bigg / \Big / \big /$	$\big / \Big / \bigg / \Bigg / \dots \Bigg / \bigg / \Big / \big /$	$\big / \Big / \bigg / \Bigg / \dots \Bigg / \bigg / \Big / \big /$	$\big / \Big / \bigg / \Bigg / \dots \Bigg / \bigg / \Big / \big /$
Inline Formulae				
Lorem ipsum sum $\sum_{i=0}^{\infty} 2^{-i}$ con setetur sadipscing	Lorem ipsum sum $\sum_{i=0}^{\infty} 2^{-i}$ con setetur sadipscing	Lorem ipsum sum $\sum_{i=0}^{\infty} 2^{-i}$ con setetur sadipscing	Lorem ipsum sum $\sum_{i=0}^{\infty} 2^{-i}$ con setetur sadipscing	Lorem ipsum sum $\sum_{i=0}^{\infty} 2^{-i}$ con setetur sadipscing

Figure 3.9: This figure displays a comparison of possible rendering outputs for the Media-Wiki math extension rendered with Firefox 24.0. Mathoid server allows one to use either a LaTeXML or MathJax renderer to generate MathML output with SVG fallback.

- LaTeXML: $\text{T}_{\text{E}}\text{X} \rightarrow \text{MathML}$ (319ms/220ms);
- Mathoid server : $\text{T}_{\text{E}}\text{X} \rightarrow \text{SVG} + \text{MathML}$ (18ms/18ms);
- Texvc : $\text{T}_{\text{E}}\text{X} \rightarrow \text{PNG}$ (99ms/71ms).

Thus, compared to the baseline (texvc), Mathoid server produced a speedup of 5.5 whereas LaTeXML is 3.2 times slower. LaTeXML and PNG seem to benefit from multiple runs, whereas the rendering time with Mathoid server stays constant.

We also converted all of the English Wikipedia articles and measured the conversion times for each equation therein. The most time consuming equation was the full form of the Ackermann function $A(4, 3)$. LaTeXML¹⁰ needs 147s to answer the HTTP request for $A(4, 3)$. According to the self-reported LaTeXML log, approximately 136.18s was used for parsing. The same request was answered by Mathoid server in 0.393s which is approximately 374 times faster. The old rendering engine needed 1.598s to produce the image. This does not include the 41ms it took to load the image from the server.

3.3.6 Conclusion, Outlook and Future Work

In the scope of Mathoid, we updated the retrograde MediaWiki math extension, and developed Mathoid server which replaces the texvc program. This enhanced the security of MediaWiki as proposed in [202] (Section A.1). It is no longer necessary to pass user input via shell access using the command-line. Nor is it necessary to move files on the server via PHP. The MediaWiki math extension is now capable of using LaTeXML and Mathoid server to convert $\text{T}_{\text{E}}\text{X}$ -like input to MathML + SVG. Based on the requirements, the user can choose if he prefers to use LaTeXML for the MathML generation (this has the advantage of Content MathML output), or he can use Mathoid server which is much faster (but does not produce Content MathML). Mathoid server takes advantage of LaTeXML since it produces MathML. The MediaWiki math extension, through Mathoid server, converts MathML to fallback SVG.¹¹ For Wikipedia itself, with only a few semantic macros, and no real applications for Content MathML produced by LaTeXML, Mathoid server alone seems to be the best choice.

We did exhaustive tests to demonstrate the power of Mathoid server with regard to scalability, performance and enhancement of user experiences. Those test results are summarized in table 3.7. Our implementation was finished in October 2013 and is currently being reviewed by the Wikimedia foundation for production use¹². Our work on the MediaWiki math extension and Mathoid server establishes a basis for further

¹⁰ltxpsgi (LaTeXML version 0.7.99; revision ef040f5)

¹¹This integral feature of Math 2.0 does not require additional source modifications, and is demonstrated for example at <http://demo.formulasearchengine.com>.

¹²Our new Mathoid based rendering was finally enabled globally on 31th of May 2016.

Table 3.7: Overview: comparison of different math rendering engines.
Values based on articles containing mathematics in the English Wikipedia.

	texvc	LaTeXML	Mathoid
relative speed	1	0.3	5
image output	PNG	PNG	SVG
Presentation MathML coverage	low	high	high
Content MathML output	no	no	yes
webservice	no	yes	yes
approximate space required on webserver	1GB	0	0
language	OCaml	Perl	JavaScript
maintained by	nobody	NIST	MathJax

math related developments within the Wikimedia platform. Those developments might be realized by individuals or research institutions in the future. For example, we have considered creating an OpenMath content dictionary [191] that is based on Wikidata items [240]. This will augment mathematical formulae with language independent semantic information. Moreover, one can use gathered statistics about formulae currently in use in Wikipedia to enhance the user interface for entering new formulae¹³. This is common for text input associated with mobile devices.

In regard to future potential work, one may note the following. There is a significant amount of hidden math markup in Wikipedia. Many of these have been entered using HTML workarounds (like subscript or superscript) or by using UTF-8 characters. This markup is difficult to find and causes problems for screen readers (since they are not aware that the mode needs to be changed). If desired by the community, by using gathered statistics and edit history, we would be able to repair these damages.

The MediaWiki math extension currently allows limited semantic macros. For example, one can use `\Reals` to obtain the symbol \mathbb{R} . At the moment, semantic macros are infrequently used in the English Wikipedia (only 17 times). One could potentially benefit by increased use of semantic macros by taking advantage of the semantic information associated with these. In the future, one could use this semantic information to take advantage of additional services, such as MathSearch [202], a global search engine which takes advantage of semantic information. Also, the use of semantic macros would provide semantic information which would provide improved screen reader output.

¹³<http://www.formulasearchengine.com/node/189>

Chapter 4

Evaluation of Augmentation Strategies and Augmentation Applications

However, despite a very careful investigation of the data of the task result, a causal dependency between task performance and degree of fulfillment of the IN of a mathematically literate user of an MIR system was not obvious. Based on the experience from the Digital Repository of Mathematical Formulae (DRMF) project (cf. Section 3.1) and the NTCIR 10 MIR task, we developed a new MIR task in Section 4.1. This is done to address the needs of Research Task 4 ‘Develop measures and benchmarks that allow for objective evaluation of MIR systems performance’. This task was accepted as official subtask at NTCIR 11 as described in “NTCIR-11 Math-2 Task Overview” by Aizawa, Kohlhase, Ounis, and Schubotz [7] and also accepted as a short paper “Challenges of Mathematical Information Retrieval in the NTCIR-11 Math Wikipedia Task” by Schubotz, Youssef, et al. [208] at the SIGIR first 2015 conference. Moreover, it’s worthwhile noting that other researchers used our test collection to evaluate their systems based on that task. For instance a full paper presented at SIGIR first 2016 [260] used our evaluation method.

Moreover, in Section 4.2, we compare the performance of a human brain with the capabilities of the MIR systems at large. This work was published as “Exploring the One-Brain-Barrier: a Manual Contribution to the NTCIR -12 Math Task” by Schubotz, Meuschke, et al. in 2016 [210]. The outcome of this comparison is not only a manifestation of human skills with regard to MIR task measures, but also a gold standard suitable to train future MIR systems.

Finally, in Section 4.3, we evaluate the similarity-measure factors proposed by Zhang

and Youssef based on the NTCIR 11 gold standard. In contrast to Zhang and Youssef, we evaluate those factors individually. The evaluation indicates that four of five factors are relevant. The fifth factor alone is of lower relevance than the other four factors. However, we do not prove that the fifth factor is irrelevant. This chapter is an extended version of the work already presented in “Evaluation of Similarity-Measure Factors for Formulae” by Schubotz, Youssef, et al. [209] and was part of the NTCIR 11 math task [7].

That last section serves as fulfillment for Research task 5 ‘Identify key components in the evaluated MIR systems and evaluate the impact of each individual building block’ where the building blocks are the similarity factors.

4.1 Developing a Standard Test Suite for Mathematical Information Retrieval Systems

MIR concerns retrieving information related to a particular mathematical concept. The NTCIR 11 math task develops an evaluation test collection for document sections retrieval of scientific articles based on human generated topics. Those topics involve a combination of formula patterns and keywords. Another task in NTCIR 11 is the optional WMC which provides a test collection for retrieval of individual mathematical formula from Wikipedia based on search topics that contain exactly one formula pattern. We developed a framework for automatic query generation and immediate evaluation. This section discusses our dataset preparation, our topic generation and evaluation methods, and summarizes the results of the participants, with a special focus on the WMC.

In order to compare different approaches and measure their performance, test collections are needed. At the CICM 2012 conference in Bremen, Germany, the first ‘MIR happening’ took place with two participants, 10,000 arXiv documents and a dataset size of 293 MB. In 2013, the NTCIR 10 math pilot task [6] for MIR attracted 6 participants and used 100,000 arXiv documents with a dataset size of 63 GB. Based on the gathered experience, MIR qualified for a main task at NTCIR 11 [7] which took place in 2014 with 8 participants and 8 million document sections of 174 GB in total. Additionally, the newly introduced WMC was appreciated by the participants. We expect that the automated feedback and evaluation framework will lower the entrance level for participants and attract even more participants in the future.

In Section 4.1.1, we describe how the Wikipedia dataset was prepared and augmented. In Section 4.1.2 and 4.1.3, we describe the query design and evaluation process respectively. In Section 4.1.4, we present the participating teams and the performance their MIR systems. In Section 4.1.5 we give a future outlook for the WMC.

4.1.1 Dataset and Feedback to Wikipedia

The test collection used at NTCIR 10 in 2013 was based on a random selection of arXiv articles converted via LaTeXML to HTML5 [222]. The arXiv is a vast and expanding source of knowledge for researchers and experts in highly specialized domains. However, neither math search engine developers (participants) nor the assessors that evaluate the search engine results usually are domain experts in all the topics addressed in the arXiv publications. Some topics discussed in the research papers are so specialized that it becomes impossible for participants to get even a preliminary idea of the content and to decide on the relevance of a formula with respect to a topic (i.e., the underlying IN). This fact adds additional complexity to debugging and testing of the math search engines. In contrast to the arXiv dataset, the Wikipedia encyclopedia contains most of the mathematical world knowledge explained in simple terms. While this knowledge is not sufficient for new research, it is perfectly suitable as a test corpus for math search competitions. This knowledge simplifies debugging and testing of the math search engines and enables the participants to test their systems on a dataset that is easier to understand and contains all formulae they are already familiar with. The English Wikipedia contains about 30k encyclopedic articles with mathematical formula. Those are written using the \TeX -like input format `texvc`. Even though the syntax in `texvc` is restricted and does not allow to write Turing complete programs, as it is possible with \TeX , \TeX is neither the optimal way to represent Mathematics on the web nor to search for formula. In contrast, MathML was designed to serve the aforementioned purposes.

Schubotz and Wicke [207] (cf. Section A.3), compare different conversion methods and identify the LaTeXML converter as the most reasonable solution with Content MathML support for Wikipedia. A majority of the participating systems use content for the search task. Therefore, both tasks (arXiv and Wikipedia) use LaTeXML to convert the original user input to MathML with parallel content and presentation markup and the original input as annotation.

In order to generate stable and unique references to each individual formula used in Wikipedia, we created a unique index. Therefore, we concatenated the page revision identifier (`oldId`) with the consecutive number for each formula starting at 0 to generate the index key. For example, `math. 1485.0` links to the first formula on revision 1485 from the Wikipedia page about the Euler identity $e^{i\pi} + 1 = 0$. From those ids one can derive many interesting statistics about the usage of mathematics within Wikipedia. For example, Figure 4.1 shows the frequency distributions of the formulae. One use case that we published at formulasearchengine.com is an auto-completion list for \TeX commands based on the usage statistics that we gathered by tokenizing the frequency ordered formulae. People editing Wikipedia articles about math with mobile devices will benefit from this feature.

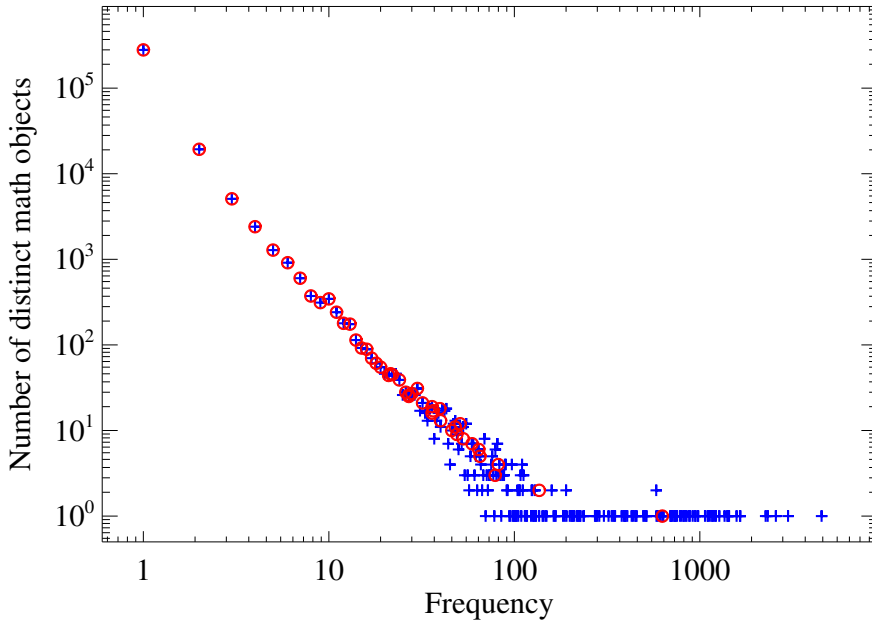


Figure 4.1: Distribution (blue ‘+’) and mean (red ‘o’) of the number of distinct formulae and their frequencies. For example, the expression n (lower right corner) occurs 2988 times and there are about 280 000 formulae that occur only once (upper left corner).

4.1.2 Topic Design

From our experience with the Math pilot task at NTCIR 10, we draw the following conclusions:

1. For each query there should be at least one relevant hit in the dataset;
2. The semantics of query variables should be well-defined;
3. only the information that would be exposed to a MIR system should be communicated to the participants;
4. The relevance criteria for each topic should be independent of the topic author and assessor. While the main task addressed these issues with human intuition, we chose a different approach for the WMC that does not involve humans.

We developed a program that generates queries in three simple steps. At first, the so called seed formula is chosen based on random selection from our mathindex. In a second step, we inject query variables. In contrast to the main task, where humans

Table 4.1: This table lists the best performing runs, with regard to (success upper number in %) and mrr (lower number in %) with page and formula-centric evaluation methods. Furthermore, we have displayed the individual results with respect to the query categories **easy**, **frequent**, **variable** and **hard**. The following acronyms were used: TUB Technische Universität Berlin (Germany), KWARC Jacobs University Bremen (Germany), RHMS Richard Montgomery High School (USA), RIT Rochester Institute of Technology (USA), MIAS Masaryk University (Czech republic), TUW Vienna University of Technology (Austria), National Institute of Informatics (NII) (Japan)

Participant	runs		page					formula				
	total	distinct	total	easy	frequent	variable	hard	total	easy	frequent	variable	hard
TUB	4	4	91	100	96	74	87	87	100	92	70	63
			73	94	30	90	46	68	87	25	86	30
KWARC	1	1	75	83	75	70	50	-	-	-	-	-
			82	95	44	97	67	-	-	-	-	-
RMHS	1	1	48	54	46	40	50	-	-	-	-	-
			02	01	00	03	01	-	-	-	-	-
RIT	17	4	88	98	79	89	63	78	95	50	81	63
			80	96	31	92	83	86	94	47	96	83
MIAS	19	4	65	97	92	15	-	63	95	83	15	13
			76	93	46	83	-	81	91	71	83	01
TUW	5	3	97	100	100	93	88	93	100	96	89	63
			82	97	50	96	54	88	96	72	94	71
NII	9	4	97	98	100	93	100	94	98	96	89	88
			76	99	49	82	67	77	89	92	78	48

chose a meaningful name for the query variable, we call our query variables x_0, x_1, \dots ¹. Finally, we generate the NTCIR 11 Extensible Markup Language (XML) Topics [110] using LaTeXML. Our relevance criterion is to find a formula similar to the seed. By our naming convention for the query variable and the absence of a topic title we ensure that no information is exposed to the participants that is not intended to be used according to the topic specification, i.e., MIR systems cannot use the name of query variables for relevance ranking. Our method generates two meta-information pieces $f(t)$ and $q(t)$ for each topic t . Here, $f(t)$ is the frequency, that indicates how many exact matches (based on exact matches on the original TeX input) for each seed are contained in the dataset, and $q(t)$ is the number of query variables used. This allows for an **a priori** classification of the search topics based on f and q . For simplicity, we partition the set of generated topics T (Figure 4.2) into the 4 following groups:

¹Our hope was that this notation, would be adapted to future tasks which was not the case (cf. 4.2).

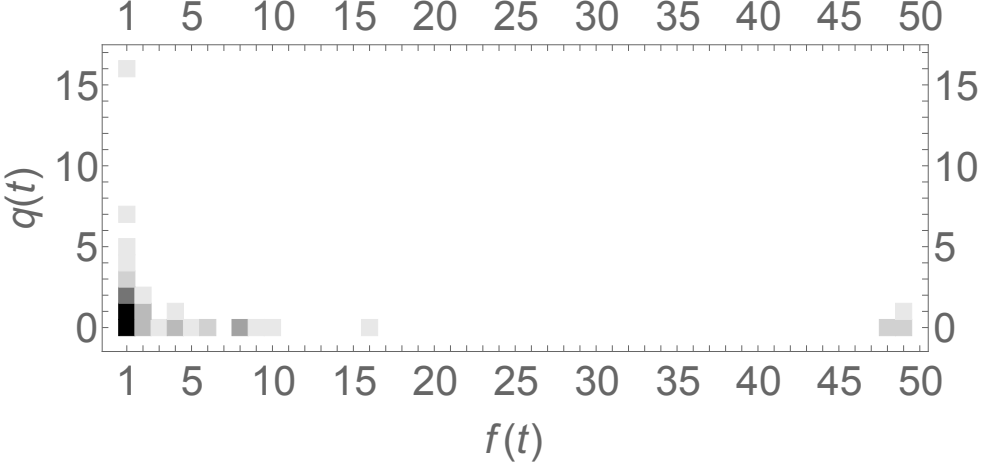


Figure 4.2: Density of the topics with regard to the frequency of seed formula $f(t)$ and the number of query variables in the topic $q(t)$. The blackest box corresponds to the highest count (up to 41 topics) and the lightest grey corresponds to only 1 topic.

Easy topics without query variables and exactly one precise match $E = \{t \in T : f(t) = q(t) - 1 = 1\}$;

Variable topics with query variables but only one exact match for the underlying seed $V = \{t \in T : f(t) = 1 \wedge q(t) > 0\}$;

Frequent topics without query variables but with non-unique seeds $F = \{t \in T : q(t) = 0 \wedge f(t) > 1\}$ and

Hard topics that contain query variables and non-unique seeds $H = \{t \in T : f(t) > 1 \wedge q(t) > 0\}$.

For the set of queries used in NTCIR 11 the following cardinalities were given: $|E| = 41$, $|V| = 27$, $|F| = 24$, $|H| = 8 \Rightarrow |T| = 100$.

4.1.3 Evaluation Process

For retrieval tasks with one known good result, a typical evaluation measure is the Mean Reciprocal Rank (MRR) [238]. In addition, our automatic evaluation software calculates glsmap in the first k hits for different levels of k , and counts the **number of found seeds** referred to as **success** in the rest of this section which corresponds to the recall for $k \rightarrow \infty$. The evaluation tool performs two types of evaluations, a **page-centric** evaluation that regards a hit as correct if the seeding page was found, and the **formula-centric** evaluation, which assumes that a hit is correct, if a formula with

exactly the same \TeX input was found. To avoid over-fitting, only aggregated results are displayed as feedback to the participants. Thus, the participants get feedback on how their systems performed on average for all topics, but they do not know how the systems performed on an individual topic or on a topic category. We observed that the intermediate feedback feature was highly appreciated by the participants, because it helped them to identify and fix bugs in their software. We observed that participants submitted 3 to 5 times until they were satisfied with the results. Some participants submitted subsequent runs under different names. This justifies the reports by the participants that the submission system helped to improve MIR systems. Some teams have improved their MRR by 50% or more.

4.1.4 Participants and Evaluation Results

We had seven participants from five countries and in total 56 runs with about 2 million hits. The results of the evaluation described in the former section are listed in Table 4.1. A detailed overview of the participants and their MIR approaches can be found in [7]. The best result with regard to success was submitted by TUW and NII. Both runs found 97% of the topics according to the page-centric evaluation. With $\text{mrr} = 82\%$ the ranking of TUW is slightly better compared to NII (74.5%). For the formula-centric evaluation NII has the best success with 94% and $\text{mrr} = 82\%$. The best TUW run achieves a mrr value of 88% at a success rate of 93%. Table 4.1 shows that there is a high correlation between the topic category and the system performance. Furthermore, all teams that submitted more than one run got very good results for the easy topics. The difference between page and formula exact evaluation is not significant.

As shown in Figure 4.3, the performance results from different teams vary more than different runs submitted by the same team. The MIaS team runs (circles) show the well-known behavior that MRR (or precision) decays with growing success (or recall respectively).

We observe that all topics except one were found by two or more participants, even if the formula exact evaluation method is used. The known good result for query 99 ($\frac{?x_0}{?x_1}$) that can be verbalized as ‘any fraction’, was found only by one team at rank 8983. More interesting is that 4 teams assigned a high rank to the result $\frac{\frac{x}{y+z}}{y}$ from the Harmonic progression page in contrast to the first mathematical expression $\frac{1}{2}$ from the Wikipedia article titled fraction that was not ranked very high.

4.1.5 Conclusion

We discussed the NTCIR 11 WMC that lowers the entrance barrier for new participants to MIR and broadens the scope of NTCIR math tasks to encyclopedic

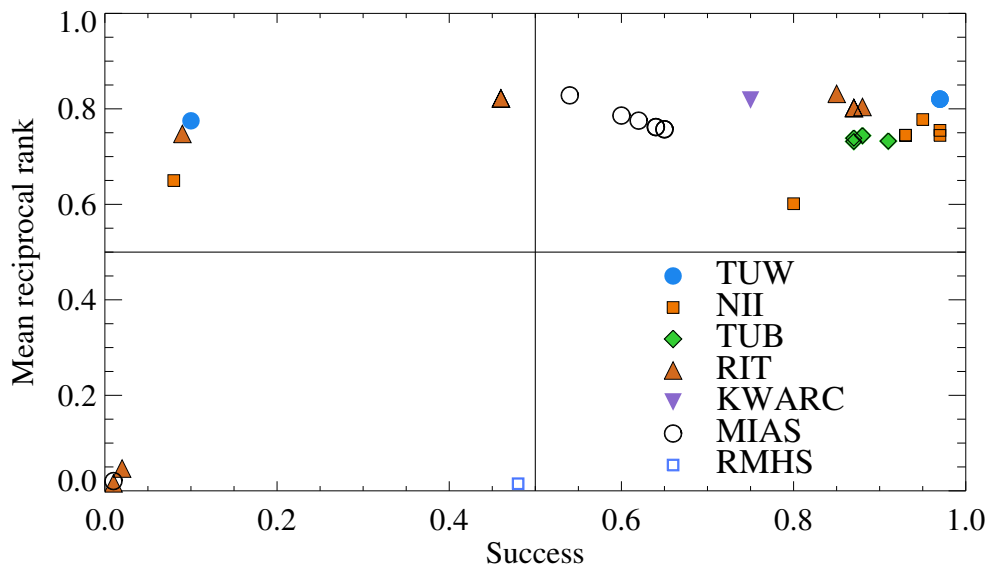


Figure 4.3: Page-centric evaluation: The Figure shows all runs with regard to MRR and success. For example, the runs of the MIAS team show a typical trade-off between MRR and success. Other teams reported that they used the automated feedback from the submission system to fix implementation problems. This increased MRR and success at the same time.

applications. We presented three main technology contributions, integrated in the MediaWiki MathSearch extension. First, we developed methods to convert Wikipedia dumps (in any language) to the main task data format including Content and Presentation MathML. Second, we developed a method for automated search pattern generation with example hits. Third, our extension provides a fully automated evaluation framework with real time feedback at submission time and a comparative evaluation for multiple submissions including hit pooling.

For the future, we plan to continue development and improvement of the platform with regard to the following aspects. We will allow for user feedback for results with regard to relevance for the entries submitted by the systems. While it is questionable that volunteers can be found to evaluate the results, participants can evaluate their own results which will be helpful for system tuning. Furthermore, we will display some basic similarity scores for each hit and will allow users to create their own search topics. The queries used for NTCIR 11 will stay available for training and testing. A query exact feedback will be displayed for new submissions for the old topics. We intend to attract more participants and to find a Math Search interest group made of mathematicians, scientists and people from the traditional Information Retrieval (IR) community. Due to the continuously available portal, participants will

be able to test new features whenever they are ready. We will publish new queries on demand, and synchronize participants with the NTCIR events.

4.2 Mathematical Information Retrieval Systems vs. a Human Brain

This section compares the search capabilities of a single human brain supported by the text search built into Wikipedia with state-of-the-art math search systems. To achieve this, we compare results of manual Wikipedia searches with the aggregated and assessed results of all systems participating in the WMC. For 26 of the 30 topics, the average relevance score of our manually retrieved results exceeded the average relevance score of other participants by more than one standard deviation. However, math search engines at large achieved better recall and retrieved highly relevant results that our ‘single-brain system’ missed for 12 topics. By categorizing the topics of NTCIR 12 into six types of queries, we observe a particular strength of math search engines to answer queries of the types ‘definition look-up’ and ‘application look-up’. However, we see the low precision of current math search engines as the main challenge that prevents their wide-spread adaption in STEM research. By combining our results with highly relevant results of all other participants, we compile a new gold standard dataset and a dataset of duplicate content items. We discuss how the two datasets can be used to improve the query formulation and content augmentation capabilities of match search engines in the future.

The flexiformalist manifesto [113] describes the vision to use machine support to break the ‘one-brain barrier’, i.e., to combine the strengths of humans and machines to advance mathematics research. The one-brain barrier is a metaphor describing that to solve a mathematical problem all relevant knowledge must be co-located in a single-brain. The past and current NTCIR math tasks [6, 7, 8] and the MIR happening at CICM 12 defined topics supposed to model mathematical INs of a human user. These events have encouraged for submissions generated using IR systems as well as submissions that domain experts compiled manually. However, there have not been any manual submissions to these events so far.

Having submitted machine-generated results in the past [202, 204, 209], we submitted a manual run for the WMC. Motivated by a user study that analyzed the requirements on math search interfaces [107], we use the manual run to derive insights for refining the query formulation of our math search engine Mathosphere and for additional future research on MIR. Metaphorically speaking, we put human brains in the place of the query interpreter to investigate how well the query language used to formulate the queries of the NTCIR 12 task can convey the respective INs. We want to see, how the human assessors judge the relevance of results that another human retrieved after interpreting the query compared to the results a machine retrieved.

4.2.1 Methods

4.2.1.1 Task Overview

The WMC (English) of NTCIR 12 is the continuation of the WMC of NTCIR 11 [208] (cf. Section 4.1). The participants received 30 search topics, which are ordered lists, whose elements can either be keywords or math elements. For example, topic 1 of NTCIR 12 consists of four elements:

1. (Text, **what**)
2. (Text, **symbol**)
3. (Text, **is**)
4. (Math, ζ).

To improve readability, we use a single space to separate the elements of topic descriptions hereafter, i.e., topic 1 reads: ‘what symbol is ζ ’. The math elements in topic descriptions can include so called **qvar** elements which are back-referencing placeholders for mathematical sub-expressions. For example, query nine includes a math expression with one **qvar** element: $*1*_{n+1} = r*1*_n(1 - *1*_n) *1*^2$. The **qvar** element can represent any identifier such as x , or a mathematical expression such as $\sqrt{x^2}$, as long as all the occurrences of $*1*$ are replaced with the same sub-expression.

Table B.7 lists all 30 topics of NTCIR 12. The participants received a subset of the English Wikipedia consisting of: (1) all pages that included a special tag used in MediaWiki to mark up mathematical formulae, hereafter referred as `<math/>`-tag; and (2) a random sample of Wikipedia pages without `<math/>`-tags.

The task was to submit ordered lists of hits for each topic. The results in all submissions (manual and automated) were pooled. We expect that the organizers used a ‘page-centric approach’ for result pooling, i.e., that hits with the same page title, but pointers to different positions in the page were combined. Due to the pooling process, tracing which engine and how many different engines retrieved a particular result is no longer possible. This aggregation is detrimental to answering our research question, since we cannot evaluate if the results returned by our ‘single-brain system’ were also found by the math search engines at large.

Two assessors independently evaluated the relevance of each result in the pool using a tripartite scale ranging from ‘not relevant = 0’ over ‘partially relevant = 1’ to ‘relevant = 2’. As in past NTCIR math tasks, the assessors at NTCIR 12 received relevance criteria as a guideline for their assessments. These relevance criteria have not been

²Note, that changes in the notation of query-variable from $?x$ in [6] to x_1 in [7] to $*1*$ in [8] were decided by the task organizer boards

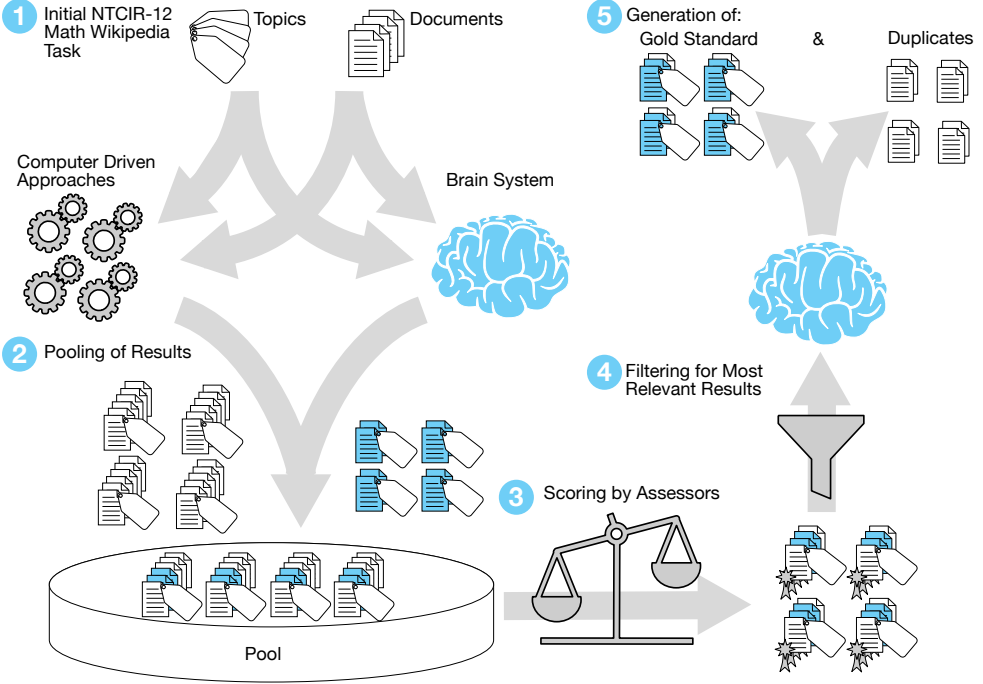


Figure 4.4: Overview of our experimental setup

made available to the participants prior to the manuscript submission deadline. The anonymized results of the assessment process were distributed to the participants. The task organizers aggregated the assessment scores by summing the two scores given to each result. For future NTCIR math tasks, we propose to provide the individual scores given by each assessor or to state the assessor agreement. The aggregated relevance scores do not allow to deduce assessor agreement with certainty, because of the ambiguous score combinations 2, 0 and 1, 1. Details on the topics, the data, the result formats, the dataset, the pooling process, and standard performance measures such as Mean Average Precision (MAP) are available in [8].

4.2.1.2 Pre-Submission

To generate our results, we followed a simple approach (see Figure 4.4). Our ‘single-brain system’ associated search terms, and entered them into the search interface at en.wikipedia.org. For some topics, the German Wikipedia was used instead and the inter- language links were used to retrieve the corresponding page in the English Wikipedia. Note that our ‘single- brain system’ was trained in physics and computer science which might have biased the results. In a second step, we identified the corresponding document in the test collection for the NTCIR 12 task which was

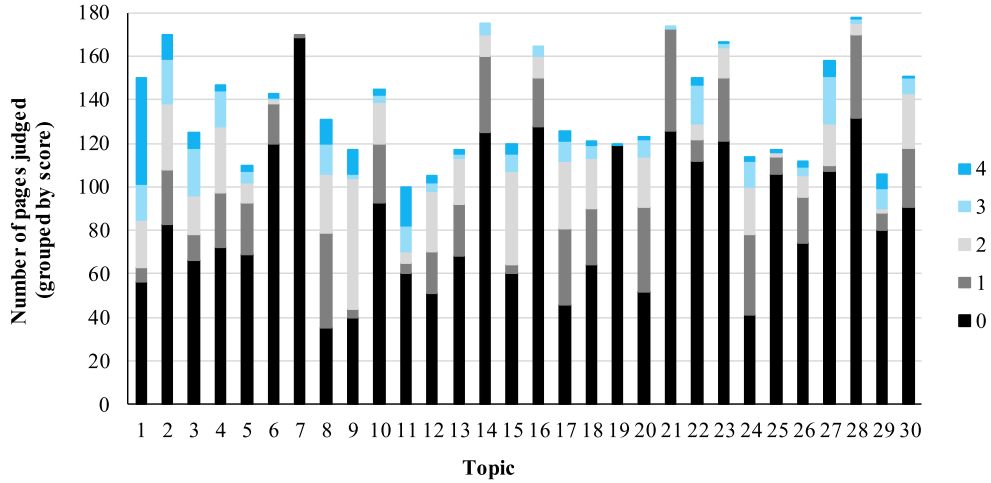


Figure 4.5: Overview of the 8,214 assessments by topic (4,107 hits, each rated by two reviewers).

not possible in four cases.

4.2.1.3 Post-Submission

After receiving the relevance feedback of the human assessors, we analyzed all pages that the assessors judged as highly relevant, but our search did not retrieve. We define results that received a score of 4 as highly relevant, i.e., both assessors classified the result as relevant. We used highly relevant results to refine our result list, with the goal of using it as a gold standard for our math search engine Mathosphere in the future. Additionally, we generated a list of duplicate results which we plan to use as training data to improve content augmentation for Mathosphere.

Figure 4.5 shows the distribution of relevance scores for each topic, i.e., how many result pages were assessed per topic and which scores were given. The number of assessed pages ranges from 100 for topic 11 to 178 for topic 28. More interestingly, the number of pages judged as highly relevant varied significantly from 0 (for topics 7, 14, 16, and 21) to 49 (for topic 1). It appears that some topics were more challenging for the participating search engines than others.

To create the gold standard, we used the following procedure:

- We added results to the result list that other participants retrieved, but we missed;
- We re-ranked our result list given the new suggestions of other participants;

- We removed results that we no longer consider relevant, because they are not helpful to gain further information as everything was already covered by items we ranked higher in our final result list;
- We excluded topics, for which we considered no result relevant to the query.

During this process, we also tracked duplicates, i.e., content that covers the same information for a topic. While we have not yet fully formalized our notion of duplicate content, we differentiate between two types of duplicates i) parent-child duplicates and ii) sister duplicates. We define ‘parent-child duplicates’ as a set of duplicate content elements with one distinct element (parent) to which all other elements (children) link. On the contrary, we define ‘sister duplicates’ as duplicate content elements that do not exhibit a distinctive link pattern between each other.

4.2.2 Results

This section presents the results of our study with regard to

- Our performance in comparison to other participants,
- The creation of a new gold standard, and
- The compilation of a dataset of duplicate content.

All results we report have been derived from analyzing the official evaluation results distributed to the participants.

Table 4.2: Assessment matrix for our results. Rows represent the rank we assigned to a result. Columns represent the relevance score a result received from the two assessors.

		relevance score					Σ
		0	1	2	3	4	
rank	1	1	1	4	3	19	28
	2	2	1	1	1	2	7
	3	0	0	0	1	0	1
	4	0	0	1	0	0	1
	5	0	0	1	0	0	1
Σ		3	2	7	5	21	38

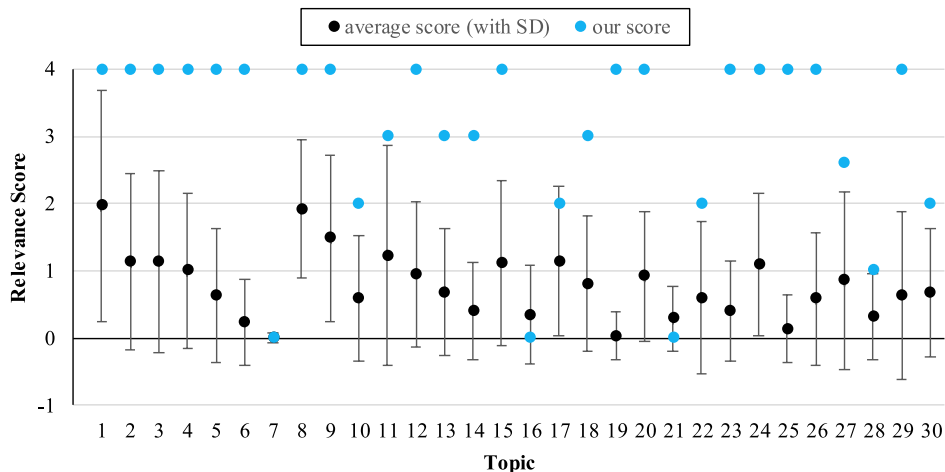


Figure 4.6: Comparison of our results to the average of the other systems

4.2.2.1 Performance

For the 30 topics, we retrieved 42 pages that we deemed relevant from en.wikipedia.org (see Table B.7). Four of our hits (the top hit for topic 7 and the lowest-ranked hits for topic 2, 3, and 13) were not part of the NTCIR corpus. Table 4.2 shows the relevance assessments of the 38 pages that were part of the corpus. Twenty-one of our results were judged as relevant by both assessors, additional five results were judged as relevant by one and as partially relevant by the other assessor. Of the 28 pages that we considered as top hits, 19 pages were judged as relevant by both assessors. Only three of our results were judged as irrelevant by both assessors. We could not deduce the reasons that caused the assessors to judge one of our top-ranked results as irrelevant and eight top-ranked results as partially relevant. One explanation could be that the assessors received relevance criteria favoring a perspective that was not apparent from the query. Table B.7 explains for each topic whether and why we agreed with the assessors' judgment.

Figure 4.6 shows for each topic the average relevance score our results received and the average relevance score of all participants. Except for three topics (7, 16 and 21) our scores are clearly above the average for all systems. For 26 of the 30 topics, our average relevance score for the topic exceeded the average score of other participants by more than one standard deviation. The NTCIR 12 organizers report the precision of participating systems at rank 5, 10, 15 and 20. Since we submitted only one result for 24 of the 30 topics and less than five results for 29 of the 30 topics, we achieved a nominally low average precision compared to the other participants. We are confident that this low precision is mainly caused by the small number of results we submitted and to a much lesser extent by the number of false positives. We assume that our precision at rank 1 is more competitive to the results of other systems and better

represents our true performance. However, the distributed evaluation results lack the necessary detail to quantitatively substantiate this assumption. Interesting additional analyses would be to compare for each topic our top- k results to the top- k results of the best performing system and to the highest rated results across all systems.

4.2.2.2 Gold Standard

Since we consider the set of topics as representative for typical queries an average Wikipedia user might have, we generated a gold standard dataset from the topics and results, to train our search engine Mathosphere. We exclusively included highly relevant results in our gold standard. Therefore, we excluded two topics (7 and 16) (see Table B.8), for which no participant retrieved relevant results. However, we decided to keep topic 21, although the assessors judged our results as irrelevant (see Table B.7, Column 21), because we consider our result a good hit for the use cases and INs our search engine addresses.

To compile the gold standard, we reviewed the 138 results judged as relevant by both assessors and obtained the following results: For 12 of the 28 topics, we found new relevant results (30 in total) that we would not have found without the help of the math search engines participating in NTCIR 12. The search engines may have returned more results that would be beneficial to us, but which we did not review, because they received a score of less than two from either of the assessors. Time constraints caused us to set this strict exclusion criterion.

Finding 30 new results in a set of 138 results pre-classified as relevant might seem like a low yield. However, one has to keep in mind that our goal differs from that of the assessors. While the assessors judged whether the search results are relevant to a specified IN unknown to us, our goal is to decide whether the results are relevant to the query.

For the topics 1 and 27, we discovered the largest numbers of additional relevant results (6 and 7 respectively). These topics reflect two typical types of INs, i.e., finding definitions of an identifier (topic 1) and finding instantiations of a formula (topic 27). We discuss these types of INs in Section 3.2 [211].

4.2.2.3 Duplicates

During the evaluation, we identified 60 instances of duplicate results. Most duplicates (57) were parent-child duplicates. For example, in the context of topic 8, the Wikipedia article on Wavelength (child) uses the formula $\nu = c/n$ and links to the refractive index (parent) in close proximity to the formula. The other children articles listed in Table 4.4 exhibit a similar pattern. On the contrary, the Wikipedia article on refractive index, uses the formula $n = c/\nu$ in the abstract and elaborates on this

Table 4.3: Sister duplicates.

Sister A	Sister B
Logical equivalence	Distributive property
R:K selection theory	Population dynamics
Tautology (rule of inference)	Exportation (logic)

relation throughout the page.

For the sister duplicate relation, we could not identify one article as the main, i.e., parent article. For example, in the context of topic 2, the pages ‘Tautology (rule of inference)’ and ‘Exportation (logic)’ contain the exactly same sentence to describe \Leftrightarrow : ‘Where ‘ \Leftrightarrow ’ is a metalogical symbol representing ‘can be replaced in a logical proof with’.’

4.2.3 Discussion

As part of the Math Task of NTCIR 12, we submitted a set of 38 manually retrieved results. For 17 of the 30 topics, our results achieved a perfect relevance score, for 26 topics our average relevance for the topic exceeded the average relevance score of other participants by more than one standard deviation (cf. Figure 4.6).

This outcome demonstrates the strength of our ‘single-brain system’ and the weaknesses of current math search engines. A human can easily distinguish different query types by analyzing the keywords given in the topic, e.g., retrieving the definition of an identifier unknown to the user opposed to retrieving a complete proof. State-of-the-art math search engines, such as our engine Mathosphere, do not yet adapt their search algorithms depending on the keywords given in the topic. Therefore, we see the development of focused MIR algorithms as a promising task for future research. From the current list of topics, we derive the following preliminary set of query types which correspond to focused MIR tasks and should be supported by a query language:

- Definition look-up (topics 1-3, 5, 6, 19, 24)
- Explanation look-up (topics 20-23)
- Proof look-up (topics 25, 26)
- Application look-up (topics 27-30)
- Computation assistance (topics 14-18)
- General formula search (topics 4, 7-13)

One approach to address focused MIR tasks is to associate mathematical formulae with corresponding metadata extracted from the mathematical documents and other sources. The DRMF Project [48, 49] (cf. Section 3.1) follows this approach. The DRMF offers formulae and their metadata isolated from the formulae’s context. Therefore, formulae can serve as standalone retrieval units, since all information necessary to interpret the formulae is given as part of a so called Formula Home Page (FHP). Extracting high quality metadata is the hardest challenge for this approach. Although research on formulae metadata extraction exists [251, 121, 211], the authors are not aware of search engines that associate metadata with formulae to improve the similarity assessment of formulae. The weakness of our ‘single-brain system’ is the lower recall compared to math search engines. We submitted only one result for 24 of the 30 topics and less than five results for 29 of the 30 topics. The task organizers report precision at ranks 5, 10, 15 and 20. Given the small number of results we submitted, our precision at rank 5 and above is low. We would have liked to compare for each topic our top- k results to the top- k results of the best performing system and to the highest rated results across all systems. However, these comparisons were unfeasible, since the official evaluation results exclusively stated aggregated performance measures for other participants.

We reviewed all results that both NTCIR assessors rated as relevant, to identify the weaknesses of our ‘single-brain system’, i.e., which relevant results we missed and why we missed them. Performing this analysis showed that math search engines participating in NTCIR 12 retrieved a number of relevant results, we were unable to find without the support of these engines. This indicates that the capabilities of today’s math search engines to identify relevant formulae exceed the capabilities of humans. However, these super-human capabilities mostly derive from higher recall, while the precision of current math search engines is still low - in our view too low to warrant wide-spread use of these systems by a broad audience.

As we present in section 4.2.2.2, we observed a particular strength of math search engines in answering the queries of topics 1 (definition look-up) and 27 (application look-up). Also for other topics with queries of the types definition look-up (topics 2-5, 24) and application look-up (topics 28-30) math search engines retrieved highly relevant information that that our ‘single-brain system’ missed.

To train our Mathosphere engine to reach a level at which the system could become a widely-used formula search engine for Wikipedia, we compiled a new gold standard dataset (cf. Table B.8). During this process, we additionally created a dataset of duplicate content items (cf. Tables 4.3 and 4.4). We envision to use the duplicate content dataset to enhance the content augmentation phase of the MIR process, i.e., during metadata extraction and index creation, rather than as training data for content querying.

4.2.4 Conclusion and Outlook

In conclusion, we regard the WMC of NTCIR 12 as a valuable preliminary step to develop a formula search engine for Wikipedia. We observed strengths of current math search engines for looking up definitions and applications of formulae. The weakness of current math search engines is their low precision. Improving math search engines to the point where they will become a similarly central fixture for STEM research as keyword-based search engines like Google have become for general purpose queries, requires substantial further research and development efforts.

We propose the following steps to reach that goal:

1. Using the gold standard dataset (Table B.8), we derived in this section to train and thereby improve the effectiveness of math search engine prototypes for Wikipedia. The dataset leverages the strengths of the human brain and balances its weaknesses with results of the participating math search engines.
2. Optimizing the efficiency of math search engines as outlined in section A.2 [204].
3. Generalizing the scope of the improved math search engines from Wikipedia to other collections and more advanced retrieval operations.

4.3 Analysis of Similarity Measure Factors

In our Formula Search Engine (FSE) team contribution for NII Testbeds and Community for Information access Research(NTCIR) 10 (cf. Section 4.1), we developed a concept which we claimed would enhance math search research significantly [204] (cf. Section A.2). The idea is to separate questions regarding **big data processing** from conceptual questions regarding math search. This leads to an accelerated development cycle because the processing in regard to math search is not distracted by data organization efforts. In this section, we take advantage of this accelerated development cycle to evaluate the similarity measure factors for formulae recently proposed in Zhang & Youssef (2014) [261] based on the NTCIR 11 data set.

This section makes two major contributions. First, using the large human-generated ground truth in NTCIR (i.e., 2500 manually evaluated document sections), this section performs a broader evaluation of Zhang and Youssef's similarity search than the one reported in [261]. That is the larger and standardized NTCIR test collection is being used, rather than a hand crafted subset of the Digital Library of Mathematical Formulae. Second, this section evaluates the contribution of each **individual** similarity factor out of the five factors identified by Zhang and Youssef. In [261], the similarity measure combines all five factors into a single metric which was evaluated collectively. In contrast, this section evaluates the impact of each factor separately,

thus providing a more fundamental insight into the contributions of each factor, and leading the way for a more targeted fine-tuning of similarity search parameters and thus to better optimization of math similarity search.

4.3.1 Formula Similarity Search

Exact formula search queries can be formalized using languages such as XQuery, XPath, or the MathWebSearch (MWS) α -equivalence language concept [118]. The result set depends on the data only, and if implemented correctly, is independent of the realization in the language used. This method works well for queries that lead to few results (on the order of 10).

For larger result sets, usual query refinement techniques (such as [46, 161]) known from traditional, general-purpose databases, can be applied in order to reduce the result set size. The authors are not aware of any math query refinement techniques specific to exact search.

In contrast, the concept of **similarity based** search (hereafter similarity search) is that a score is assigned to the Cartesian product of the formulas and search patterns. For each query, a partially score-ordered result set is returned. Since the score calculation might be computationally expensive, approximations to this exact scoring method are usually used. Note that in the worst case, the score for a pattern-formula tuple depends on the full collection of formulae in the data set. It is common that this score will depend on a set of aggregated values derived from the data set. An example of aggregated values are the frequencies of variable occurrence. Regardless of the technical aspects, there is no established way to define similarities between formulae. For example one system which uses similarity search is MIaS [218].

4.3.2 Similarity Measure Factors

The following math similarity measure factors are listed and explained in [261]:

1. Taxonomic Distance based on Content Dictionaries;
2. Data-Type Hierarchical Level;
3. Match Depth;
4. Query Coverage; and
5. Formula vs. Expression.

Factor 1 assumes a taxonomy of functions, and assigns more similarity between two functions if they belong to the same taxonomic class (e.g., if both are trigonometric

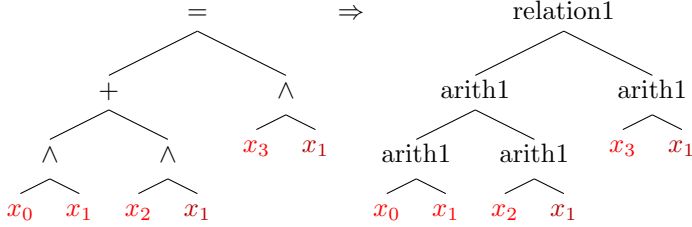


Figure 4.7: Projection to the Content Dictionary dimension for the search pattern $x_0^{x_1} + x_2^{x_1} = x_3^{x_1}$.

functions), and less similarity if the functions belong to two different classes, within a larger super-class (e.g., one trigonometric function and the logarithm, being in different classes but within the super-class of elementary functions). **Factor 2** assumes a hierarchy of math objects, such as constants (level 0), variables (level 1), functions and operations (level 2), functionals like integration and differentiation (level 3), and so on. The higher the levels of two math objects are, the more weight is assigned to their similarity/dissimilarity. **Factor 3** assigns a larger distance (less similarity) between a query expression/formula Q and a hit expression/formula E when Q is more deeply nested in E . For example, if Q is $x^2 + y^2$, E_1 is $x^2 + y^2 + 2xy$, and E_2 is $\exp\left(\frac{1}{x^2 + y^2 + 5}\right)$, then Q is assumed to be more similar to E_1 than to E_2 . **Factor 4** measures how much of a query Q is present in a potential hit E : the more terms and structure of Q there is in E , the more similarity is assigned between Q and E . Finally, **Factor 5** assigns more weight to hits that are formulas (involving a comparison operator) such as ' $\sin^2 x + \cos^2 x = 1$ ' than to non-formula expressions like ' $\sin x + \cos y$ '.

In our evaluation, we treat each factor as a separate measure and we qualify these factors in the following way. For Factor 1, we perform a reduction of the math objects and search pattern by projecting patterns and formulae to the content dictionary dimension. For example, we replace arithmetic operators, such as those in $\{+, -, *, /\}$, by their content dictionary 'arith1' (cf. Figure 4.7) to generalize to the taxonomic class.

For Factor 2, we do the same thing but with regard to the data-type dimension. If a reduced pattern matches a reduced formula, we call this a generalized hit. We count the number of generalized hits with regard to the assessor ranking v .

For Factor 3, we calculate the Match Depth penalty factor (or level-coefficient) [218]. Note that since only 8 of the 55 given search patterns contain exact matches at any depth, the sample size for this evaluation is significantly smaller. Average relevances are calculated for all Match-Depth penalty factors.

For Factor 4, queries and formulas are converted to bags of tokens. We compare each pattern with each formula and count the number of tokens from each pattern which

are also tokens of the formulae. We normalize this to the total number of tokens with regard to pattern. In a subsequent step, we group (i.e., quantize) the results into 11 buckets (0-5%, 15-25%, ... 85-95%, 95-100%) and calculate the average relevance ranking for each bucket.

For Factor 5, we apply a method similar to measure factors 1 and 2. For every math object, we determine if it is in the Formula category or in the Expression category. The math object is in the Formula category if it contains a relational operator at the root level, and otherwise is in the Expression category. The following set of relational operators were considered as indicators for the Formula category $\{=, \equiv, \neq, <, >, \leq, \geq\}$.

4.3.3 Evaluation

For the evaluation, we used the document sections originating from the arXiv. Those were selected in the pooling process of the NTCIR 11 task. We refer to this data set as the gold standard data set. For each of the 50 topics defined in the NTCIR 11 math task, human assessors assigned relevance rankings to 50 document sections. This leads to a collection of 2429 distinct arXiv document sections with 5 ranking categories from 0 (not relevant) to 4 (most relevant). For more details concerning the evaluation process, refer to the NTCIR 11 Math overview section [7] (cf. Section 4.1). Our evaluation deals with similarity measures for individual formulae. Most of the 2250 document sections that contain mathematical expressions have more than one math expression. This is unfortunate for the task at hand, since our similarity factors are formula-centric and not document section based. Furthermore, the consideration of keywords that are included in the topics in addition to the formula search patterns listed in Table B.6 add an additional source of error for our evaluation. However, due to the reasonable size of the data set, those effects might average out. Thus, we are still able to show the correlation between relevance ranking and the presence of similarity factors.

We mapped the relevance ranking for a document section to all mathematical objects contained in that section. For articles with more than one formula, this ensured that the formula that leads to classification of the article as relevant, is also marked as relevant. For example, if a document with two mathematical objects was considered as relevant to a particular topic by the assessors, both formulas are considered as relevant with regard to all math search patterns occurring in this topic. Forty-seven topics include only one search pattern, two topics include two search patterns, and topic 48 contains 4 search patterns (cf. Table B.6). The downside of this approach is that formulae that are in the same article and did not influence the ranking of the assessor in a positive way, were marked as relevant, even though they are not relevant.

For our implementation, we used Apache Flink with the Java API. We published our source code on github.com under the code name 'mathosphere2'. Since all the

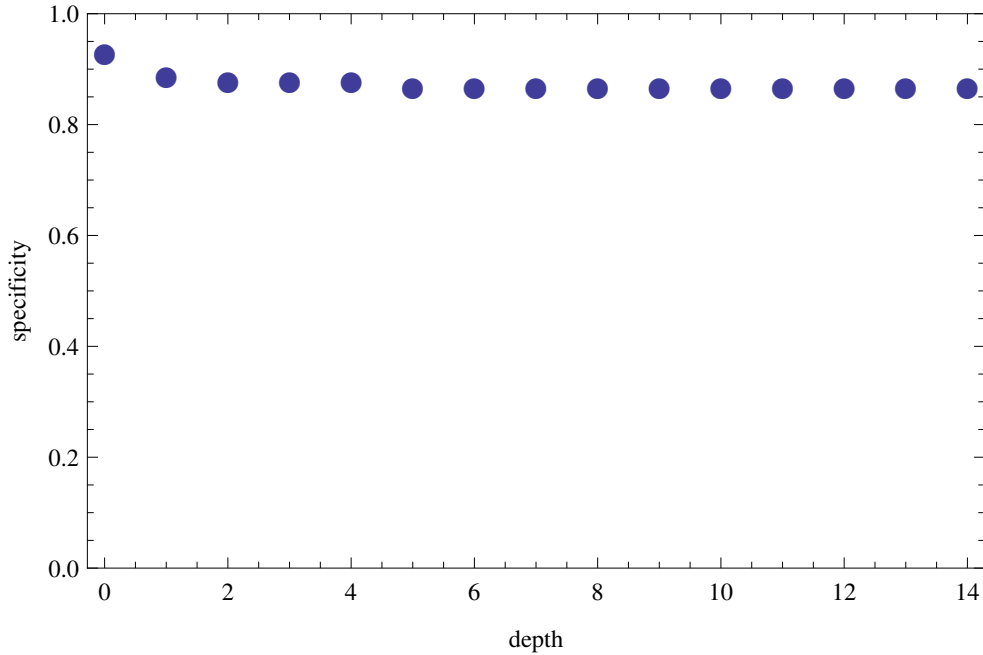


Figure 4.8: Specificity versus match depth.

algorithms described here are embarrassingly parallel, the required runtime for a fixed number of formulae scales almost linearly with the available computational resources. This demonstrates that the factors evaluated can be used in an interactive application. The remainder of this section presents the evaluation results for each individual factor.

4.3.3.1 Taxonomic Distance and Data Type

The 443 (449) matches for content dictionary (data type) abstraction have the following respective recall, precision and specificity metrics for both factors

$$r = 0.27, \quad p = 0.74, \quad s = 0.91. \quad (4.1)$$

The high specificity shows that both are relevant similarity factors.

4.3.3.2 Match Depth

Only 10 of 55 math patterns (namely 1f1.0, 1f1.1, 12f1.0, 13f1.0, 15f1.0, 18f1.0, 20f1.0, 38f1.0, 45f1.0, 47f1.0 in Table B.6) contain exact matches. For the 9 underlying topics 479 pages, were evaluated by the assessors. We considered documents that contain

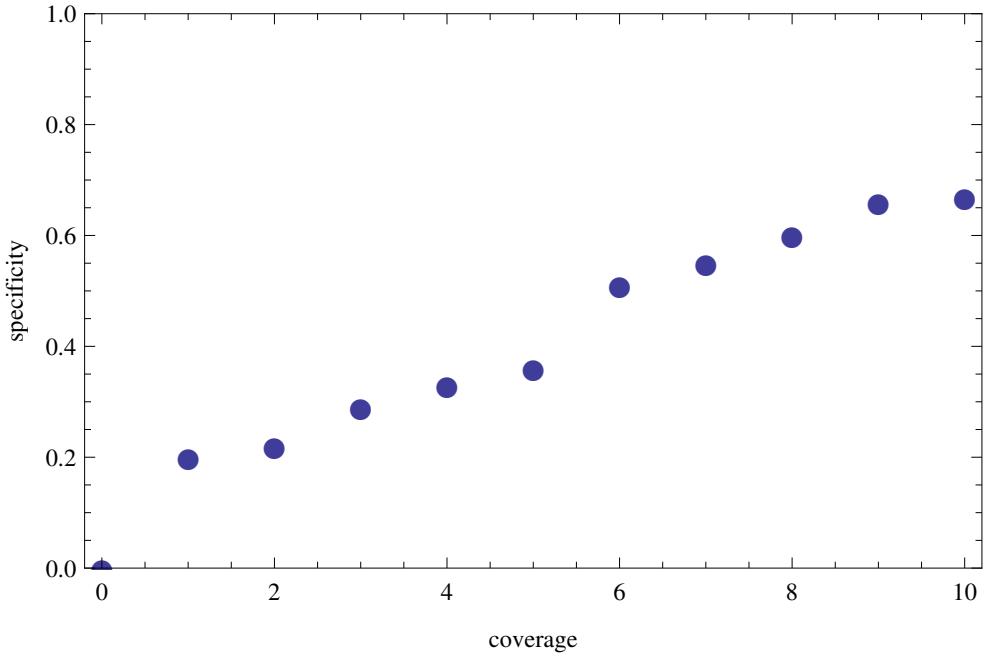


Figure 4.9: Specificity versus query coverage.

exact matches to the formula pattern as retrieved documents and calculated the match depths for them.

We measured the recall, precision and specificity for all 10 search patterns and for match depth from 0 to 14. The results are averaged over all the 10 search patterns for each depth, and are presented in Table 4.5. As evident in Table 4.5 and Figure 4.8, the specificity is very high, and it is higher for smaller depths. This indicates that the match depth factor is a relevant similarity factor, and confirms that the smaller the depth, i.e., the less deeply nested the match, the higher the relevance.

4.3.3.3 Coverage

To calculate the Coverage factor, we took the maximum value of the coverages over all (search pattern, formula) pairs within a document section. For each coverage level (from 0 to 10), we compute average recall, precision and specificity over all 55 search patterns. The results are presented in Table 4.6.

As Table 4.6 and Figure 4.9 show, the specificity is higher for higher levels of coverage, thus showing that coverage is a relevant similarity measure. Notice that the specificity of the coverage factor is not as high as the specificity of the earlier factors. This

could be attributed to the coverage being insensitive to the mathematical structure of expressions. This insensitivity makes coverage a less important factor.

4.3.3.4 Formula vs. Expression

In order to test the hypothesis that mathematical objects classified as formulae are more relevant compared to non-formula expressions, the search returns all and only articles containing at least one formula.

For this factor we found that the average recall, precision and specificity over all 55 search patterns are:

$$r = 0.28, \quad p = 0.49, \quad s = 0.26. \quad (4.2)$$

The low specificity shows that many sections that contain expressions but not formulae have been considered as relevant by the assessors. This seems to indicate that this factor is of lower relevance to search ranking than the other 4 factors considered.

4.3.4 Conclusion and Outlook

The NTCIR 11 data set provides a good basis for our evaluation. We have found good evidence that four out of five factors are relevant. For the nominal factors 3 (match depth) and 4 (coverage), we demonstrated (anti)-correlation to the specificity. This indicates that these factors can be used for result ranking in Mathematical Information Retrieval (MIR) systems. It is not very surprising that the measured categorical values for content and data-type abstraction are almost identical. While these abstractions differ in their conceptual background, their actual implementations are similar. This justifies the approach of Zhang and Youssef to combine both factors and use the taxonomic distance to compare nodes of data type function only. However, with special regard to query refinement and content summarization (which are not part of the task), further research in this direction is needed. At this point, we also note that the aforementioned factors heavily rely on high quality Content Mathematical Markup Language. Even though the Content MathML automatically generated by LaTeXXML is a decent starting pointing, we still see improvement potential here.

We have observed two points which will help improve the analysis of our similarity factors and formula-centric MIR systems which incorporate a combination of these factors. The retrieval units in the math task are document sections, not individual equations. In our evaluation, we used the best matching result if there was more than one formula. This assumption could easily be dropped if the retrieval unit was more fine-grained. In the NTCIR 11 math task, the influence of keywords was considered. Since the analysis presented in this section does not take keywords into account, the influence of the keywords to the relevance ranking adds random noise from the similarity factor viewpoint.

In NTCIR 11, there was a new experimental Wikipedia Subtask (cf. Section 4.1). Both of these weak points are not available in the Wikipedia Subtask and therefore we are looking forward to future NTCIR conferences which might incorporate the Math Wikipedia Task (WMC) as a second main task, and provide human evaluation for the Wikipedia query results.

Table 4.4: Parent-child duplicates.

Parent	Children
Binomial coefficient	Binomial theorem, Combination, Lottery mathematics, Pascal's triangle
Damping ratio	RLC circuit
Determinant	Complex number
Direct sum of modules	Exterior algebra, Linear complex structure, Unbounded operator
Faraday's law	Electromagnetic field, Maxwell's equations
Hypergeometric function	Barnes integral, Lauricella hypergeometric series
If and only if	Truth table
Legendre symbol	Jacobi symbol, Quadratic residuosity problem
Logistic function	Feedforward neural network, Sigmoid function, Maximum sustainable yield, Population model, Theoretical ecology
Logistic map	Attractor, Chaos computing, Coupled map lattice, Discrete time and continuous time, File:Cml2e.gif, Parameter space
Mild-slope equation	Sea state
Newton's laws of motion	Braking distance, Cauchy momentum equation, Dynamical simulation, Inertia, Mass in special relativity, Mechanics, Moment of inertia, Newton (unit), Perturbation theory, Pulse wave velocity, Rotation around a fixed axis
Propellant mass fraction	Single-stage-to-orbit
Pythagorean theorem	Pythagoras, Crossed ladders problem, Isosceles triangle, Law of cosines, Slant height, Special right triangles, Triangle
Refractive index	Aether drag hypothesis, Cherenkov radiation, Coherence length, Fizeau experiment, Multiangle light scattering, Optics, Total internal reflection, Wavelength

Table 4.5: This table lists the match depth d , average recall r , average precision p , and average specificity s , all averaged over 10 search patterns.

d	r	p	s
0	0.15	0.72	0.93
1	0.24	0.67	0.89
2	0.32	0.78	0.88
3	0.38	0.82	0.88
4	0.40	0.82	0.88
5	0.40	0.74	0.87
6	0.40	0.74	0.87
10	0.40	0.74	0.87
13	0.40	0.74	0.87
14	0.40	0.74	0.87

Table 4.6: This table lists precision p , recall r , and specificity s depending on the coverage category c .

c	r	p	s
0	1.00	0.48	0.00
1	0.79	0.48	0.20
2	0.77	0.48	0.22
3	0.69	0.47	0.29
4	0.65	0.47	0.33
5	0.62	0.47	0.36
6	0.50	0.48	0.51
7	0.47	0.49	0.55
8	0.37	0.46	0.60
9	0.34	0.48	0.66
10	0.31	0.46	0.67

Chapter 5

Conclusion and Greater Impact

In this last chapter, we summarize our results in Section 5.1 and their impact to the international research community as well as the benefit for possible applications and future research opportunities in Section 5.2.

5.1 Summary

Information Retrieval (IR) systems, such as search engines and recommendation services, have become part of our daily lives. Given their tremendous success on textual data, their scope has been extended to all kind of data types such as images or video streams. For their key strength, textual data, they support all kind of languages; even those who are not widely spoken, or require special characters and presentation rules (like printing text from right to left). However, **a weakness of current IR systems is the lack of support for mathematical expressions** which serve - like words - as constituents of sentences (cf. Figure 5.1). For example, the worlds largest encyclopedia, Wikipedia, did not treat mathematical expression as ‘first class citizens’ and used images instead of mathematical expressions. This did not only cause problems for research purposes such as Natural Language Processing (NLP) or data mining, but also effected people with limited vision, which are dependent on screen readers.

From the equality it follows that From the equality $\sqrt{2} = \frac{a}{b}$ it follows that $2 = \frac{a^2}{b^2}$.

Figure 5.1: An English sentence from a proof, without (left) and with (right) mathematical expressions.

To improve the math awareness of current Mathematical Information Retrieval (MIR) systems, this thesis augmented mathematical formulae to make formula related queries more effective. Moreover, the thesis showed how web content providers such as Wikipedia can present mathematical expressions more efficient with regard to speed, robustness, accessibility and scalability. Therefore, this doctoral thesis proposed a paradigm change for MIR research. Instead of dealing with the complex and interdisciplinary problems of MIR, informatics research should **focus on the problem of Mathematical Formula Data Management (FDM)**. In contrast to MIR, FDM neither addresses the analysis of humans and their mind setting, nor solves mathematical problems such as theorem proving or computation.

To address the FDM research challenges, we proposed a threefold classification of mutually orthogonal problem areas. The FDM research areas are **Content Querying, Content Augmentation and Efficient Execution**. In a first step, we analyzed MIR research papers with regard to our threefold schema. The results show, that most papers contribute to different categories in the classification schema. Consequently, each research group had to cope with a large variety of challenges at the same time. Prior to the raise of the NII Testbeds and Community for Information access Research(NTCIR) MIR tasks, individual papers were hard to compare, since they were using different datasets and different evaluation methods and metrics.

To address the evaluation problems, we contributed to the development an internationally accepted **standard for evaluating MIR systems**. In the thesis, we described our contributions from the very first cornerstone at the MIR happening at CICM in 2012 until our latest contribution to the second official full IR task¹ on mathematics retrieval at NTCIR 12 in 2016. In 2012, we were one in two participants of the ‘MIR happening’. We developed the MediaWiki extension MathSearch which combined the existing formula-only search engine MathWebSearch (MWS) with the built-in MediaWiki text search engine. To make the formulae readable for the math search engine MWS, we also had to replace MediaWiki math rendering engine. Furthermore, we had to convert the test data, publications from arXiv, to the MediaWiki markup language Wikitext. There was no formal evaluation. Instead, the topics, which were a combination of formula patterns and keywords, were given to the participants at the live demo. However, the live demo of both participating search engines showed some results which impressed the audience. Consequently, a working-group was founded which lead to NTCIR 10 pilot task that was launched in 2013. In our contribution to that task, we proved the effectiveness of the separation of Efficient Execution from the other FDM challenges. To achieve that, we combined the general purpose big data processing framework Apache Flink² with naïve MIR approaches in a framework we coined **Mathosphere**. To retrieve documents that are relevant to the **topics**, a direct comparison, using standard IR metrics provided by the NTCIR 10 math pilot task, showed that our approach performed well (position

¹Worldwide, there are three major conferences that accept IR tasks that is TREC from the U.S.A., CLEF from Europe and NTCIR from Asia. The conference organizers aim for distinct topics.

²Apache Flink was called Stratosphere at that time.

2 of 13 according to the applied metric). A weakness of our approach was the long processing time of about 10 minutes (in comparison to other systems answering each individual query within seconds after days of indexing). However, the overall low precision and recall scores of all system, did not justify to invest in performance improvements for research prototypes. Nonetheless, despite a very careful investigation of the data of the task result, a causal dependency between task performance and degree of fulfillment of the Information Need (IN) of a mathematically literate user of an MIR system, was not obvious for us. Based on this experience, we developed a new MIR task which was based on the Wikipedia corpus and did not require any human involvement. The whole pipeline, corpus generation, query generation, result submission and evaluation was fully automated. Moreover, the IN was described using a single pattern, described in query language which can be translates to xQuery. Seven teams from five countries participated and submitted runs, whose scores varied heavily. In contrast to previous tasks, that ranked the results on a tripartite scale (not relevant, partially relevant, relevant), our evaluation was based on a bipartite scale (known relevant, unknown). Thus, the evaluation was based on the position of the first known relevant result in the result list. Moreover, we classified our queries in four classes of difficulty which provided a more detailed investigation of the system strengths. Finally, in our latest contribution to NTCIR 12 in 2016, we compared the performance of a human brain with the capabilities of the MIR systems at the continuation of large. Based on this study, we created a gold standard dataset to test math search engines for their suitability for the application in Wikipedia.

To develop a better understanding of the mathematical IN, we started the Digital Repository of Mathematical Formulae (DRMF) project, together with mathematicians at National Institute of Standards and Technology (NIST). We customized the MediaWiki platform to create a digital compendium of mathematical formulae. Whereas Wikipedia and other web authoring tools manifest notions or descriptions as first class objects, the DRMF does that with mathematical formulae. Thus, in the context of this thesis, the DRMF project defined the desired format for mathematical formulae. The design of the DRMF platform aims to represent **individual formulae in a context-free form** with well-defined semantics and without ambiguity. The most desired DRMF search capability, is to filter the repository using semantic features. These semantic features predominantly consider the leave nodes of the formula content tree ³. Thus, the main effort was put into the semantification, were we got tremendous support by many student volunteers from high schools close to NIST.

Following the desire for automated semantic augmentation, we developed the concept of Mathematical Language Processing (MLP), which adapts methods from NLP to texts containing mathematical expressions. Started as a student project in October 2012, we developed a mechanism that could automatically **extract the semantics of identifiers** in formulae from the text surrounding mathematical expressions. Later, we extended the approach to infer identifier semantics from related articles as well. In addition, we discovered that the implicit naming conventions for identifier names

³The formula content tree is comparable to the abstract syntax tree in computer science.

can be described using the notation of namespaces.

Despite a very careful investigation of the data from the NTCIR 10 task results, we were still unable to determine the essential factors, that caused a good performance at the task. To get deeper insights into the effect of similarity search for MIR, we investigated the **individual factors of similarity**. While we established a general method to investigate the performance of similarity factors, based on human generated relevance judgments, the absence of semantics remains the essential hindering factor to build more effective MIR tools. However, we continued the investigation of non-textual similarity features in a student project. In that project, we adapted a method from citation analysis to link analysis. As earlier in the Mathosphere project, we employed Apache Flink to keep the implementation effort to manage Wikipedia sized datasets. Unlike mathematical formulae, the unambiguous links created by humans, were directly accessible by our system. Thus, an advantage of our approach, in contrast to ordinary text-based methods, was that our input data was better machine-readable. Our methods did not depend on a particular language, and had used less computing power, even though our novel approach was not particularly well-trained for a particular IR task. The evaluation indicates that our method returns related articles, whereas the text based methods return similar articles.

To address the issues with poor presentation of mathematical content in the MediaWiki platform, which we uncovered at the MIR happening in 2012, we evaluated several approaches to render mathematical formulae in the MediaWiki platform. In the end, none of the tools available at that time performed reasonably well with regard to the evaluation criteria **Speed, Robustness, Scalability and Accessibility**. Consequently, we developed a novel MathJax based rendering service which we coined **Mathoid**. After the presentation of our approach in 2014, it took another two years, until Wikimedia decided to use the new method as default for formula rendering. Note, that the Mathoid follows - once again - the separation approach. In contrast to the old rendering method, which is deeply integrated into the software and requires the installation of a LaTeX on the web server, Mathoid uses a service-oriented approach. In particular Mathoid is an isolated service with the sole purpose to render mathematics. While Mathoid was the first specialized service used in the Wikimedia environment, the service oriented approach was thereafter adapted by other services such as graphoid (the rendering service for graphs) and citoid (the rendering service for citations). Motivated by our experience with a lot of corner cases, where our new math rendering method deviated from the old rendering, we developed automatic visual regression testing for mathematical formulae rendering. This method is capable to automatically identify significant differences in the math rendering output.

5.2 Impact and Prospectives

This thesis made three main contributions:

1. It **analyzed** the strength and weaknesses of current MIR systems and established the Math Wikipedia Task (WMC) task which has been continued by the NTCIR conference as one of two international standard MIR tasks;
2. Based on the analysis, it **augmented** mathematical notation as foundation for future MIR systems to better fit the IN from the Science, Technology, Engineering and Mathematics (STEM) domain; and
3. It presented a solution, how large web publishers, can efficiently **present** mathematics to satisfy the IN needs of each individual visitor which has been adapted by the Wikimedia foundation to serve mathematical content for 15 billion page views per month.

The MIR research presented in this thesis, resulted in:

- 14 peer reviewed papers [202, 204, 126, 48, 207, 180, 7, 208, 49, 211, 210, 205, 206, 213] at international conferences;
- Which have been cited 71 times⁴;
- Two invited talks;
- 6 contributions to workshops and doctoral programs; and
- 4106 commits to various open source projects⁵.

With regard to the research tasks defined in Section 1.2, the following contributions have been made:

Task 1 Research about existing MIR technology and classify their contributions regarding the three FDM research challenges.

We were the first, who classified exiting MIR technology according to the FDM research challenges. In accordance with another MIR survey [84], we conclude that there is still significant room for improvement. Moreover, the need for MIR research has been motivated sufficiently well. The available tools and technologies are a meaningful justification for future MIR research. However, our analysis suggests that future research should focus on one of the particular aspects. For computer scientists, we suggest to choose from one of the large variety of open research questions in the area of Content Augmentation or Content Querying. Moreover, we would greatly appreciate user studies to better model the IN of the mathematically literate audience. In conclusion, our FDM approach was a useful guide though this dissertation project and helped to focus on the essential informatics research questions. Moreover, it will serve as compass and theodolite to design novel MIR research projects.

⁴According to Google Scholar on 15th of September 2016

⁵According to github.com on 15th of September 2016

Task 2 Develop a representation for semantic formulae that might serve as augmentation target.

We created the first digital compendium of mathematical formulae. In contrast to Wikipedia and other web authoring tools, which manifest notions or descriptions as first class objects, the DRMF does that with mathematical formulae. On a technical perspective that is manifested using Formula Home Page (FHP). A FHP represents the full semantic information, w.r.t. an individual formulae. Definitions of symbols occurring in the formula, proof and constraint data are only examples of the requirement list to FHP. Since this requirement list was tailored to fulfill the need of working mathematicians, it is perfectly suitable as an augmentation target. In particular, we learned that mathematical information extraction for formula augmentation needs to focus on a very specific requirement.

While currently the amount of manual labor to import data from various sources to the repository is still high, and the interaction of mathematicians with the platform is low, the curation of the content itself and the advertisement of the platform in the mathematical community is neither in the scope of this thesis, nor part of FDM research. From an informatics perspective, the DRMF project will serve as prototype and gold standard for Content Augmentation tasks in the future. More specifically, we are currently developing quality metrics for mathematical notation that evaluates the mathematical data quality with regard to four quality dimensions: (1) Typographical- and Lexical-; (2) Syntactical- and Structural-; (3) Semantical-; and (4) Meta- data qQuality.

For research task 2 the following publications were significant:

- Our vision paper “Digital Repository of Mathematical Formulae” by Cohl, McClain, Saunders, Schubotz, and Williams which was accepted as a short paper at Conference on Intelligent Computer Mathematics (CICM) 2014; and
- A full paper at CICM 2015 titled “Growing the DRMF with Generic LaTeX Sources” by Cohl, Schubotz, McClain, Saunders, Zou, Mohammed, and Danoff.

Task 3 Augment mathematical formulae with regard to the identified extraction targets.

Following our desire for semantic augmentation, and the insights gained from the DRMF project, we elaborated our idea first published in [204] and extracted identifier semantics from the surrounding text. While the precision and recall values are still subject to improvement our solid gold standard and the well-defined IN provide a solid basis for future improvements. Moreover, our method can and should be expanded to extract all the semantic features identified in the DRMF project. In addition to that, the extracted semantics can be used for a wide range of applications. Two ongoing student projects are researching on a math-aware question answering system

for Wikidata. Another student project dealing about unit inference in Wikidata are only two possible application. Moreover, we submitted a grant proposal application that is currently under review which elaborates on further research directions founded on our MLP concept.

Without an efficient mechanism to eventually present mathematics all the research has no effect on our society. Our mechanism to present mathematical formulae in a fast, accessible, robust and scalable way outperformed all approaches available until know, and is now finally used in production by the Wikimedia foundation. Beside all the obvious advantages that are enumerated in the corresponding sections, we would like to emphasize that this has special impact on people in countries with poor Internet connections. Making math integral part of the HTML5 content, saves data, and - even equally important - reduces the number of requests required to load a page.

We envision that our automated visual regressing testing developments will shorten development circles for future improvements in math rendering such as supporting more mathematical characters, more semantics, formula interactivity, and author feedback. Moreover, we encourage other web publishers to consider using Mathoid for their math rendering.

Our research for research task 3 resulted in five publications:

- A work in progress paper “Mathematical Language Processing Project” by Pagel and Schubotz which was published at CICM in [180];
- A full paper at SIGIR first 2016 titled “Semantification of Identifiers in Mathematics for MIR” [211] by Schubotz, Grigorev, Leich, Cohl, Meuschke, Gipp, Youssef, and Markl; and
- Another work in progress paper at CICM 2016 “Getting the units right” by Schubotz, Veenhuis, and Cohl [206].
- Schubotz and Wicke presented “Mathoid: Robust, Scalable, Fast and Accessible Math Rendering for Wikipedia” [207]. That paper was accepted as full paper at CICM 2014.
- Moreover, in “A Smooth Transition to Modern mathoid-based Math Rendering in Wikipedia with Automatic Visual Regression Testing” [205] by Schubotz and Sexton published in 2016, we describe how automatic visual regression testing will shorten development circles for production level math rendering engines.

Task 4 Develop measures and benchmarks that allow for objective evaluation of MIR systems performance.

We were involved in setting the cornerstones for objective evaluations for MIR systems; not only as participants, but also as founders of the WMC task. After several

attempts and continuous improvement of datasets and evaluation metrics, we were able to create a gold standard, suitable to test search engine candidates for mathematical world knowledge retrieval from encyclopedias.

Even though more research is required to benchmark advanced MIR systems, we propose to use our gold standard to train MIR systems for world knowledge. Thus, making mathematical world knowledge available to MIR systems and eventually to a general audience is the next goal. Consequently, we propose to discontinue the NTCIR MIR tasks until more reliable real world usage data from productive MIR systems is available. In a consecutive step, the gained experience should be used to extend the scope and address the IN of researchers and domain experts which will also require to process additional data sources.

The following scientific publications emphasize our impact with regard to the research task at hand:

- Schubotz, Youssef, Markl, and Cohl described the aforementioned WMC task “Challenges of Mathematical Information Retrieval in the NTCIR-11 Math Wikipedia Task” [208]. This task was accepted as official subtask at NTCIR 11 task as described in “NTCIR-11 Math-2 Task Overview” [7] by Aizawa, Kohlhase, Ounis, and Schubotz and also accepted as a short paper at the SIGIR first 2015 conference. Note, that this benchmark was also used by others to evaluate the performance of their systems. For instance a full paper presented at SIGIR first 2016 [260] used our evaluation method.
- In Schubotz, Leich, and Markl we showed in our NTCIR 10 math pilot task contribution “Querying large Collections of Mathematical Publications” [204] how to separate the challenge of big data processing from fundamental research questions in MIR.
- In “Exploring the One-Brain-Barrier: a Manual Contribution to the NTCIR -12 Math Task” by Schubotz, Meuschke, Leich, and Gipp published in 2016 [210], we compared the performance of a human brain with the capabilities of the MIR systems at large.
- “Making Math Searchable in Wikipedia” by Schubotz [202] describes our contribution to the MIR happening at the international CICM in 2012.

Task 5 Identify key components in the evaluated MIR systems and evaluate the impact of each individual building block.

Many approaches (for instance [3, 71, 184, 91, 101, 247, 122, 128, 138, 136, 156, 158, 160, 162, 181, 166, 219, 254, 252, 255]) extract features from documents and calculate a similarity score based on the features. However, Zhang and Youssef [261] were the first, who systematically addressed the problem of similarity measurement for mathematical formulae and identified factors of similarity. Building upon on

that preliminary work, we establish a general method, to measure the performance of individual similarity factor based on human generated relevance judgments (such as those generated in the NTCIR MIR tasks). While we were able to show that four of the five proposed factors are relevant, the absence of semantics remains the essential hindering factor to build more effective MIR tools. However, we continued the investigation of non-textual similarity features in a student project which analyzed the proximity of Wikipedia links. While the employment of the big data processing Apache Flink helped to keep the implementation effort to manage Wikipedia sized datasets, we also gained additional justification for our claim that more semantics is needed. The superiority of linked-based approaches over text-based approaches might be explained by the fact that the unambiguous links created by humans were directly accessible by our system. In contrast, text-based methods need to investigate more effort is needed to extract semantics from textual data. This is a further indicator that extracting semantics from mathematical expression is an important task.

For the future, we are planning to use formulae (if available) in addition to links to improve the recommendation quality of related articles. A first prototype⁶ that used features extracted from mathematical formulae alone shows promising results. Moreover, our formula similarity factor assessment approach and the corresponding is useful for future research on formula similarity. Using this tool researchers can estimate the effectiveness of new formulae similarity approaches without building a math search system and perform an involved evaluation.

Our publications related to research task 5 were:

- Our NTCIR 11 contribution “Evaluation of Similarity-Measure Factors for Formulae” by Schubotz, Youssef, Markl, Cohl, and Li [209], and
- “Evaluating Link-based Recommendations for Wikipedia” by Schwarzer, Schubotz, Meuschke, Breitingner, Markl, and Gipp which was accepted as a full paper in 2016 at Joint Conference on Digital Libraries (JCDL)⁷ [213].

⁶ The prototype is in German and is part of the MediaWiki extension MathSearch extension as of April 20th 2014. See for example <http://1.formulasearchengine.com/fi> in the section ‘similar pages’.

⁷JCDL was ranked as A* in the CORE conference ranking.

Appendix A

Applications and Case Studies

In this chapter, we describe applications and case studies that have been performed in the course of this doctoral thesis.

Section A.1 describes our participation at the first MIR happening at the international CICM in 2012 and is based on the publication “Making Math Searchable in Wikipedia” by Schubotz [202]. As participant on this first MIR happening, we presented our approach to combine the α -equivalence based (cf. Definition 2.3.1) formula search engine MWS [118] with the standard text search engine Apache Lucene. The other participant used the webmias search engine [218]. Webmias uses a unification-based approach and is implemented as Apache Lucene extension. While there was no formal evaluation of this happening, it turned out that our approach returned fewer, but more relevant results. In addition to that, and even more important for our following research, this happening uncovered several research gaps that we addressed in subsequent publications.

One of the insights gained in this happening [202] was that a lot of effort has already been spent, to make the execution engine of MWS efficient. However, our overall subjective satisfaction with the results from the MIR happening was not very high. Thus, our goal was to develop a framework that was optimized for fast prototyping rather than efficient execution. For instance, we demonstrated in our open source project Mathematical Markup Language Query Generator that all MWS queries could be translated to xQuery expressions with only little effort. To address this issue we describe in Section A.2, how to separate the challenge of big data processing from fundamental research questions in MIR. To achieve that, we combined the general purpose big data processing framework Apache Flink (which was called Stratosphere at that time) with naïve MIR approaches in a framework we coined **Mathosphere**. To retrieve relevant documents to the **topics**, a direct comparison, using standard IR metrics provided by the NTCIR 10 math pilot task, showed that our approach

performs reasonably well. Parts of that section were published by Schubotz, Leich, and Markl in their NTCIR 10 math pilot task contribution “Querying large Collections of Mathematical Publications” [204]. We describe the Mathosphere system in Section A.2.

Finally, in Section A.3, we rethink the Mathoid math rendering project presented in Section 3.3. While this research contribution already described the issues with the math rendering service at that time and demonstrated the improvements in a research prototype, the infrastructure was not quite ready to use the new rendering method in the production environment. This version has been forked by benetech to provide accessibility support for people with limited vision. Thus, it took another two years, until the Wikimedia decided to use the new method as default for formula rendering. Note, that the service oriented approach that was first developed for the math rendering service Mathoid was also extended by graphoid (the rendering service for graphs) and citoid (the rendering service for citations). Moreover, in “A Smooth Transition to Modern mathoid-based Math Rendering in Wikipedia with Automatic Visual Regression Testing” by Schubotz and Sexton published in 2016 [205], we describe how automatic visual regression testing will shorten development circles for production level math rendering engines. Moreover, this project also serves as preliminary work for stage-3 of the DRMF project which aims to include mathematical formulae from not LaTeX sources.

A.1 Combining Text and Formula Search

Wikipedia, the world largest encyclopedia, contains a lot of knowledge that is expressed as formulae exclusively. Unfortunately, this knowledge is currently not fully accessible by intelligent IR systems. This immense body of knowledge is hidden from value-added services such as search. In this section, we present our MathSearch implementation for Wikipedia that enables users to perform a combined text and math search and fully unlock the potential benefits.

In this section, we demonstrate the power of a combined text and formula search suitable for Wikipedia. Next, we treat the conversion and indexing of formula section and then present our combined text and formula search interface in the subsequent section.

A.1.1 Excursus: Math Parsing in MediaWiki

As of 2012, the Wikimedia Foundation, as well as the majority of content providers that provide mathematical formulae, used images to display their mathematical content. On the one hand this is a very robust solution, but on the other hand it is has serious disadvantages [144]. IR is challenging using images alone. To overcome

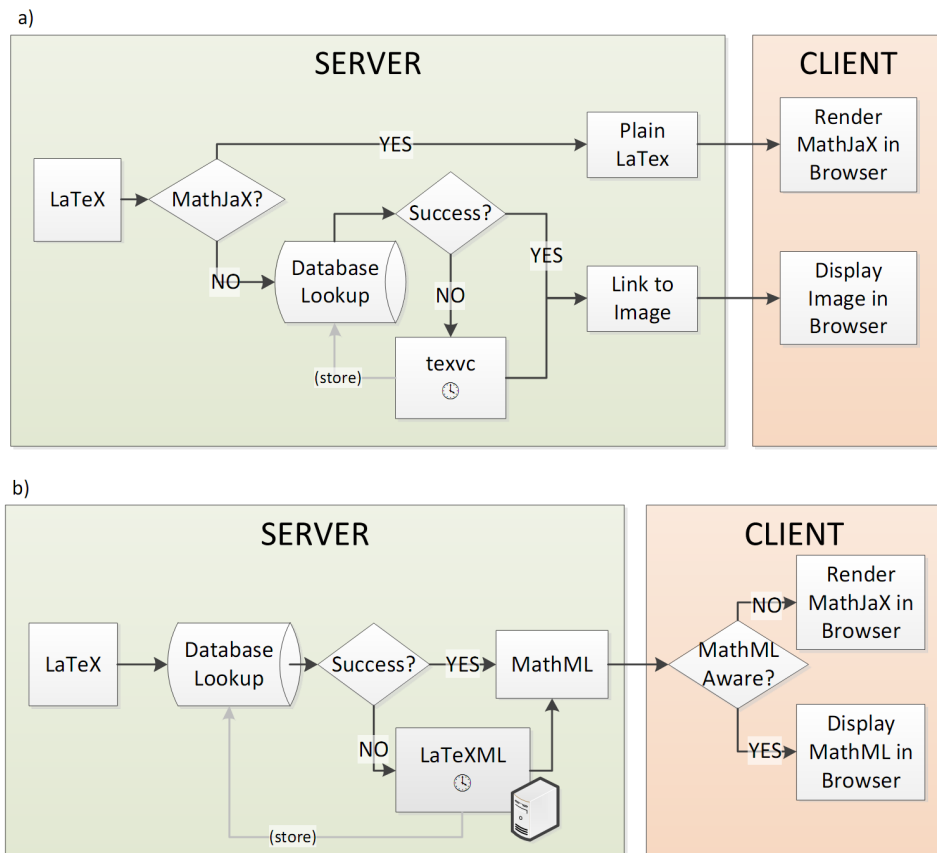


Figure A.1: An overview of the LaTeX rendering process in the current version of MediaWiki (top) and the proposed new method (bottom). The clock symbol denotes that the rendering process takes about a second per formula on a typical desktop computer. The computer icon denotes that the LaTeXML rendering process may be processed on another server.

these shortcomings MathML was developed to create mathematical notations suitable for websites [159, 250]. However, neither content providers, nor web browser developers incorporated this standard in an adequate manner. In fact, according to www.caniuse.com/mathml only 26.17% of today's browsers support MathML. As a consequence, the American Mathematical Society, the American Institute of Physics, Elsevier, IEEE and many other important players in this community support the MathJax project [45] initiated in 2009. The MathJax JavaScript library converts both, the plain LaTeX source and MathML, to a format that may be displayed on almost any device and browser. This is a major advantage, since it enables content providers to refrain from using the deprecated images.

From an IR and math search point-of-view MathML is better than LaTeXsource, since it can be processed without compilation. On the other hand MathML was not designed to be written by humans. Thus, a content provider who decides to use MathML will need to convert the standard LaTeXinput to MathML. In particular, for mathematical search it is highly recommended to use a converter that produces meaningful Content MathML¹ as well as correct Presentation MathML².

As of 2012, Wikipedia uses the texvc [244] renderer to convert all the formulae that are encoded in a variant of LaTeX in the wiki source code of the pages, to images³. A current trend among the MediaWiki [68] software development community is to bypass LaTeX rendering on the server side and to use MathJax to convert the LaTeX source instead (cf. Figure A.1-a). Unfortunately, this leads to a slow page loads since all the LaTeX formulae have to be compiled by the client side and information concerning mathematical search or information extraction is unavailable on the server side.

Thus, we have changed the way mathematical formulae are processed by MediaWiki as follows (cf. Figure A.1-b). The texvc renderer that was installed on the local MediaWiki server has been replaced by an http-interface to one or more⁴ LaTeXML daemons [74, 155] that might be installed on remote servers. These LaTeXML daemons convert the LaTeX formula to MathML that is stored in the central database and delivered to the client. On the client side MathJax might be used, if the client browser is not capable of displaying MathML out of the box. The source code of our implementation is publicly available in the LaTeXML branch of the MediaWiki source repository. Detailed instructions on the installation may be found at <http://www.formulasearchengine.com>.

In contrast to the other LaTeX to MathML converters, that provide correct Presentation MathML, LaTeXML provides Content MathML. This content information becomes valuable if one wants to search for formulae[221]. See Section A.3 for further developments in the area of ‘Math parsing in MediaWiki’ which are not directly required to understand the following sections.

A.1.2 Combining Math Expressions and Text

In order to benefit from the knowledge conveyed in the formulae, a specialized formula index is needed. The logical first step would be to apply well-known techniques from text search. This attempt was chosen for example by [164, 162]. Due to the fact that there are many degrees of freedom in the notation of the formula (e.g., choice of the variables, order of terms, ...) there are a couple of associated problems in practice. Mišutka and Galamboš [162] treat these problems by applying seven unification rules.

¹Content MathML is a semantic representation of the operation contained in the formula.

²Presentation MathML is the visual representation of the formula that is displayed by the browser.

³There were some approaches that use texvc to produce MathML as well, but due to the missing capabilities in browsers to display MathML, this attempt was not followed up later on.

⁴A basic load balancer has been integrated.

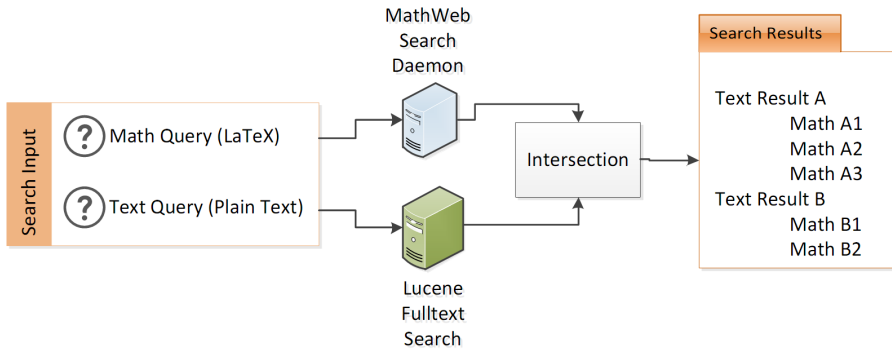


Figure A.2: Illustration of the system architecture corresponding to the MathSearch plug-in for MediaWiki platform. Users specify a search query that consists of a text query (e.g. Gröbner) and a math query (e.g. $a?x^2 + b?y^2 + ?z$) in a LaTeXfashion that might involve placeholder (e.g. $?x, ?z, ?z$). After the evaluation of the queries by the MWS Daemon and the Lucene fulltext search respectively, an intersection of the results is performed. All pages that match the text (An article that is about ‘Stable Normal Forms for Polynomial System Solving’) are listed, conditioned on whether at least one formula matched the math query is result. (in this case equation 16: $p_1 = ax_1^2 + bx_2^2 + \epsilon_1 x_1 y_1$). The matching formula is then displayed as subitem of the text result preview.

However, this does not treat the root cause of the problem.

More promising is the approach by Michael Kohlhase’s group. They have come up with a scalable [115] search index for mathematical formula called MWS [118]. MWS indexes the content representation of the target formulae in a unified format, so that the degrees of freedom in the notation have been eliminated. Although MWS meets these major user requirements in mathematical search, it leads to a large number of results that are not relevant to user queries. Therefore, we have come up with a relatively simple but effective idea. We combine the MWS unification based mathematical search engine with a traditional full text search engine, that provides good precision and reduces the number of irrelevant results dramatically.

Making use of the LaTeXML version (cf. Section A.3) of MediaWiki that contains formulae in Content MathML format, we build the open source MediaWiki site search expansion MathSearch [199] that combines text and formula search (cf. Figure A.2) in the following way. The existing MWS engine (that has no front-end) is combined with the text search engine Lucene [74]. We designed a simple frontend with two input fields. In one field the user can specify a query for the mathematical content and the other optional field is reserved for the textual query. Both queries are then processed by the corresponding search engine. After that, the search results are grouped by text search results and the corresponding mathematical search results are classified as subitems of the text results.

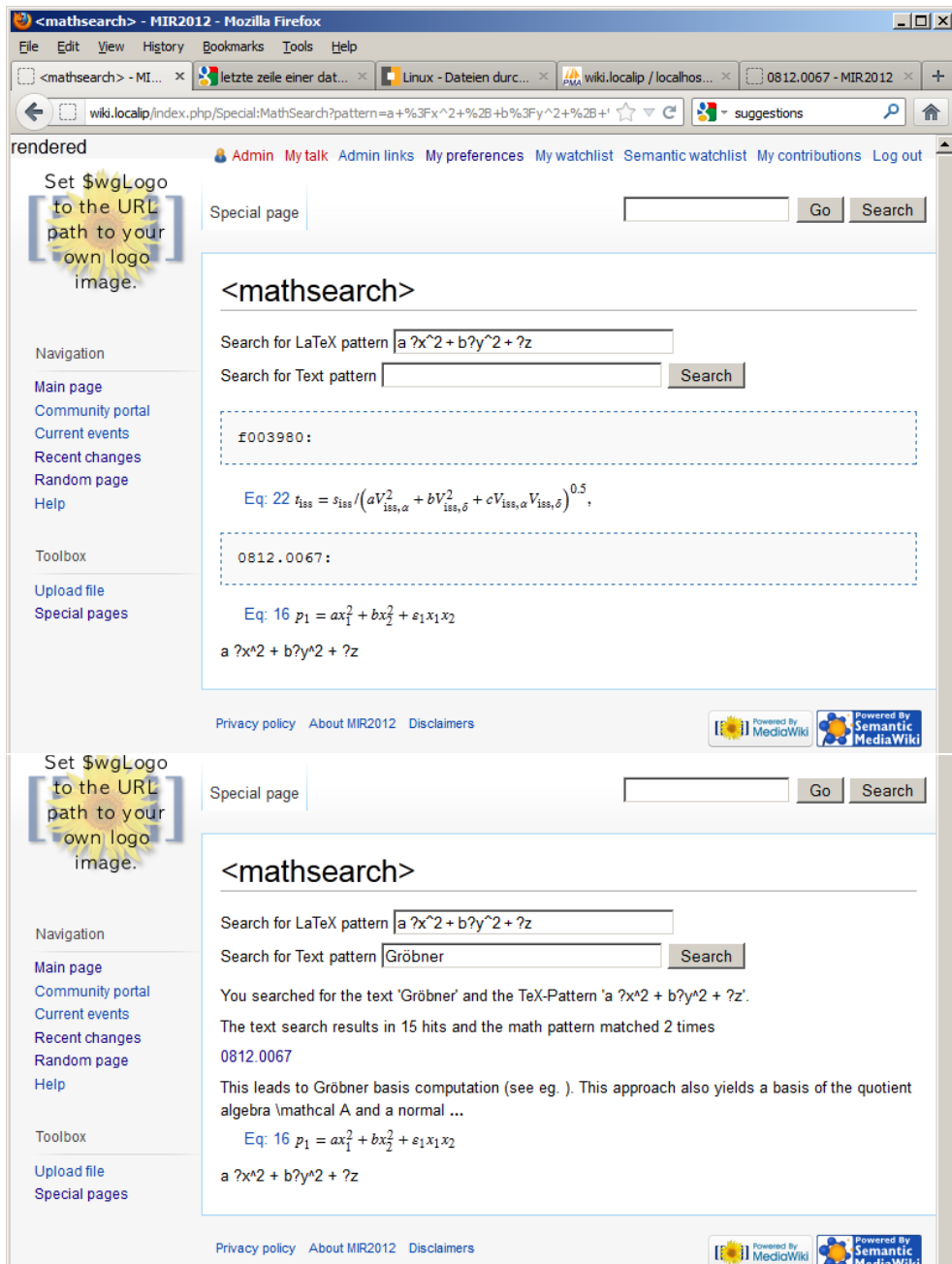


Figure A.3: First version of the MediaWiki MathSearch extension. Showing formula only search on the top in comparison to combined text and formula search (bottom).

A.1.3 Experimental Evaluation

Our combined MathSearch solution was demonstrated at the MIR happening at CICM 2012. At this event our system competed with WebMiaS [218] that is based on canonicalized Presentation MathML, math trees and similarity search. The implementation of WebMiaS is based on Lucene. For math indexing a specialized weighting function is used. At the MIR happening real human mathematicians post questions to the competitors in real time that have to submit the questions to their systems manually.

As a data basis, both systems got 10'000 documents extracted from the ArXiv corpus two days in advance. Whereas WebMiaS hosted the documents on a remote server, we converted the documents to the Wiki markup and demonstrated our MathSearch solution on a virtual machine hosted on a standard laptop.

One of the tasks was to find

$$B_{p+n} = B_n + B_{n+1} \bmod p \text{ for all } n = 0, 1, 2, \dots$$

for the query B_{p+n} . Both search engines came up with the correct result ranked at first position. WebMiaS found 455 additional results whereas our system just found one, since the search query did not specify to use α -conversions (cf. Definition 2.3.2).

For the combined text and math query Gröbner, $a?x^2 + b?y^2 + ?z$ (cf. Figure A.2) our system came up with the expected solution. A detailed report from the MIR can be found in [56]. Moreover, all the 16 MIR topics are published as [112].

A.1.4 Conclusion

In this section, we demonstrated how one can make use of the knowledge contained in the mathematical formulae by using a combined text and formula search interface. Furthermore, we showcased how content providers can publish their mathematical content in a modern and efficient way. In Section 3.2, we apply advanced text mining methods to retrieve additional information about the symbols used in the formula and to resolve ambiguities.

A.2 Mathosphere: No Index Mathematical Information Retrieval

In this section, we present our approach for searching mathematical formulae using a distributed big data analytics platform. We focus on a batch query approach that does not rely on **specialized indexes** which are usually domain-dependent and restrict the expressiveness of the query language. Instead, we use Stratosphere, a distributed

data processing platform for Big Data Analytics that accesses data in a non-indexed format. This system is very effective for answering batches of queries that a researcher may wish to evaluate in bulk on large data sets.

We demonstrate our approach using the NTCIR10 math task which provides a set of formula patterns and a test data corpus. We showcase a simple data analysis program for answering the given queries. We interpret the formula patterns as **regular expressions** and assume that matches to these expressions are also relevant search results to the end-user. Based on the evaluation of our results by mathematicians from Zentralblatt Mathematik and mathematics students from Jacobs University, we conclude that our assumption holds principally with regard to precision and recall.

Our work is just a first step towards a well-defined query language and processing system for scientific publications that allows researchers to specify their IN in terms of mathematical formulae and their contexts. We envision that our system can be utilized to realize such a vision.

We propose to use a query language with fixed and well-defined semantics that describes IN in terms of formulae and text. The main advantage of such a query language in contrast to natural language is that the query results are well-defined. This separates the hard research problem of transforming researchers questions to formal queries from the technical challenge of query execution.

The NTCIR10 math pilot task [6] provides a set formula/text patterns without fixed semantics and a reference corpus that consists of 100 000 documents from the Cornell ePrint arXiv⁵. These patterns are split into two subtasks. For the formula search subtask (FS) the patterns are a list of formulae with (back-referencing) wild-cards (cf. Table B.1). The full text subtask (FT) provides words in addition to such patterns (cf. Table B.4). Since there is no meta information on how to process these queries, we interpret the formula queries very strictly, i.e., we only return results that match the pattern without any implicit semantics. Furthermore, we treat the given words as a space delimited list of keywords.

Recent approaches to formulae search, e.g., [118], focus on matching the tree structures of formulae and patterns. A lot of development effort has been put into the efficient and distributed execution of such substitution tree-based queries [115]. These approaches [118, 138, 115] consist of two phases. In a first step, the content is indexed and in a second step queries are answered based on the information contained in the index. Obviously, the advantage of this approach is the short response time, if the query can be answered using the data stored in the index. One drawback of the current implementation of MWS is that the corpus must not change after indexing. Even though, if update operations will be supported in the future, they will be connected with computational overhead. Systems based on Lucene [77] for example, allow for temporary updates that require index re-builds in the long term.

⁵<http://www.arxiv.org>

In our approach, we focus on the quality of the result, rather than runtime. We choose to improve the quality of our algorithm iteratively, until the result provides an added value for researchers. Following the concept of rapid prototyping [152], our approach speeds up the development effort dramatically, and allows to focus on the core functionality i.e., to decide if a formula matches a query. The management of the data volume is done by the Stratosphere platform in the background. In the context of the NTCIR10 math task, we showcase that our system which was developed in less than two person months, is able to filter and rank formula that match the query.

Currently, we do not address strategies for optimizing the runtime and the ease of use. For these optimizations, one can learn from elaborated concepts for query optimization originating from the database community which is subject to future work.

A.2.1 System Description

We use Stratosphere [30] as a platform for executing the queries specified in the NTCIR 10 math task. Stratosphere consists of three layers:

- Meteor [94] the high level scripting language;
- The PACT [30] programming model for implementing operators; and
- The Nephele [30] execution engine for parallel computation.

To solve the NTCIR 10 tasks, we identified following processing steps:

1. Load the data
2. Parse queries and data
3. Filter data based on queries
4. Calculate individual ranking function
5. Evaluate ranking functions based on whole-corpus
6. Return top-ranked results
7. Export the results

In this subsection, we describe how we execute these steps. Thereby, we outline which parts are evaluated by Stratosphere's native code, and which parts are performed by our user defined code. Due to the easy extensibility of the Stratosphere platform that allows to integrate user code as well on the Meteor as on the PACT level, very little overhead is required to extend the core functionality.

A.2.1.1 Data Preparation

After downloading and extracting the NTCIR 10 test collection, we concatenated the included files to a single one that was transferred to our distributed file system (HDFS). A single file is more efficient, since the system can read the data sequentially from disk. The content of this file has the following structure `<ARXIVFILE
Filename=$filename> $filecontent</ARXIVFILE>`.

The resulting file was stored in a distributed file system so that all computers that have been used to process the data could read the required data block wise from local disk.

A.2.1.2 High Level Program Design

After having imported the data to our environment, we designed the principal data flow of the system. Therefore, we had to specify the inputs and outputs and the type of each operator. Stratosphere extends the MapReduce [57] concept and introduces new operator types like *Cross*, *Match* and *CoGroup* [30]. However, for this task the operator types *Map*, *Reduce*, *Cross* and *Match* are sufficient. These operators can be regarded as second order functions that execute arbitrary user code based on the following number of inputs and outputs record

$$Map : \mathbf{R} \rightarrow \mathbf{R}^* \quad (\text{A.1})$$

$$Reduce_k : \mathbf{R}^* \rightarrow \mathbf{R}^* \quad (\text{A.2})$$

$$Cross : \mathbf{R}^2 \rightarrow \mathbf{R}^* \quad (\text{A.3})$$

$$Match_{k,l} : \mathbf{R}^2 \rightarrow \mathbf{R}^* \quad (\text{A.4})$$

Here \mathbf{R} denotes the space of all possible records and \mathbf{R}^* the Kleene Closure of \mathbf{R} . Each record $r \in \mathbf{R}$ consists of a finite number of fields r_i . The *Map* operator gets one record at a time as input and can output zero or more records for each input record. In contrast to the original *Reduce* operator [57], which gets all records with the same key, the Stratosphere *Reduce* operator ($Reduce_k$) has an additional parameter k that defines the index of the record field that is used as key. As a consequence the input record set $\mathbf{I} \subset \mathbf{R}^l$ of length l is portioned into $n = |\text{dom } \mathbf{I}_k| \leq l$ lists of records, each having the same value for field k . Here n is the number of distinct values for \mathbf{I} . Thus, exactly one list will contain a record that has a particular key-value for field k . For the special case, $n = l$, *Map* and *Reduce* are equivalent. Furthermore, the *Reduce* contract has the option to sort the input list with regard to a secondary field that we use to get a sorted list of the query result. *Cross*, receives the Cartesian product of the input record sets and *Match* is a *Cross* followed by a filter that only emits record pairs with matching key attributes $r_k = r_l$.

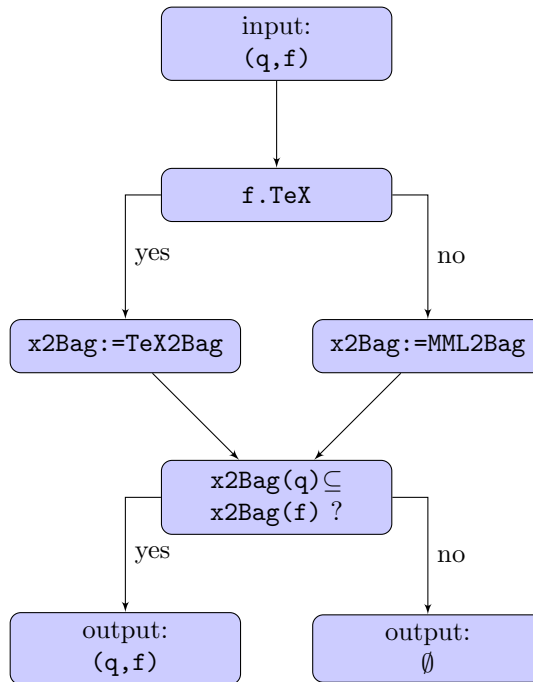


Figure A.4: Logic of `formulafilter` operator: Filters formula that contain all tokens specified in the query.

A.2.1.3 Implementation of the Operators

This section describes the user defined operators that we developed extending the core functionality of the Stratosphere system in detail. The principal data flow is the following:

- First, the queries and the data is loaded. In a subsequent step, token-based query filters identify the equations that are match candidates.
- Independent, overall word- and variable- frequencies are obtained.
- In the following scoring phase, we compare the equation structure of query and match candidate and calculate a score that takes into account the overall frequencies. Finally, the results are sorted and the top-scored hits are returned.

Query Compilation The query compilation is implemented via a *Map* operator (cf. Listing A.1 [line 5-12]) that gets one record of a single text field with the NTCIR topic Extensible Markup Language (XML) subtree which includes formula pattern and



Obviously, the \TeX token filter contains more information compared to the MathML filter. The latter extracts variables, numbers and operators only, rather than the full feature set of MathML elements which would require an unified MathML representation. Unification of MathML is not yet solved, or at least not available as a program library⁶. Without unification, we face a lot of problems due to degrees of freedom in the MathML standard. For example, the underscore ($_$) in the \TeX code could be represented via the `msub`, `msubsup` or `mmultiscripts` Presentation MathML⁷ element. Thus, the we use the MathML filter only, if no \TeX code is available (cf. Figure A.4).

⁷<http://www.w3.org/TR/MathML3/chapter3.html>

compression, e.g.,

$$\begin{array}{l}
 \text{} \\
 \text{ <qvar name="x">} \\
 \text{ <d>} \\
 \text{ f} \qquad \qquad \qquad \rightarrow a[(. * ?); d[f]; \backslash 1] \\
 \text{ </d>} \\
 \text{ <qvar name="x">} \\
 \text{}
 \end{array} \tag{A.5}$$

is inspired by Mathematica's full form style ⁸ with square brackets. We ignore XML attributes and replace the `qvar` elements (*?a* and *?p* in the running example) by regular expression. Multiple occurrences of the same name attribute of the `qvar` element are treated with back-reference as in (A.5). That said, the fourth and fifth field read *mrow[mo[Σ]; mrow[msub[(. *); mi[n]]; msub[(. *); mi[n]]]]* and *apply[sum; apply[times; apply[csymbol[subscript]; (. *); ci[n]]; apply[csymbol[subscript]; (. *); ci[n]]]]*. These regular expressions take into account the structure of the query, rather than just counting the features as done in the MathML filter.

The last field is a list with keywords (convergence in the sample query). One attempt is to include the porter stemming algorithm [245]. Even though this algorithm was quite successful in standard NLP tasks, it has some drawbacks for math search. For example, the words **derivative** and **derivation** are both transformed to the stemmed term **deriv**. Thus, we just use the original lower-case string for the text search.

Extraction Operators This section describes the extraction of variables and terms from the test corpus data. Our four extraction operators are implemented as *Map* contracts.

- The first *Map* (cf. Listing A.1 [14-17]) gets a publication formatted as HTML document and shipped in a PACT record with only a single string field as input. It attaches another field that contains the file identifier for the later use as key.
- The consecutive formula filter operator (cf. Listing A.1 [19-23]) uses a regular expression to find all math tags. It emits one record per formula containing the MathML element and the document id. Another regular expression captures T_EX code of the formulae, if available and attaches an additional field to the record.
- The following *Map* task (cf. Listing A.1 [25]) tokenizes the T_EX and MathML code respectively, the same way as done with the query T_EX/MathML. Even though these tasks are different operators, they share code for the tokenization.
- Another *Map* task (cf. Listing A.1 [26]) tokenizes the words of the article in the same style.

⁸<http://reference.wolfram.com/language/ref/FullForm.html>

```

1  using math, xml; //load the custom package
2  $d = read from "hdfs://localhost/NTCIR.xml" split at "ARXIVFILE";
3  $q = read "hdfs://localhost/queries.xml" split at "topic";
4  //map-task for query compilation
5  $q = transform $q into{
6    num: xGet($q, "./num"),
7    texFilter: TeX2Bag(xGet($q, "./TeXquery")),
8    mmlFilter: mml2Bag(xGet($q, "./pquery")),
9    pmml: compress xGet($q, "./query/pquery"),
10   cmml: compress xGet($q, "./query/cquery"),
11   words: split xGet($q, "./query/words"),
12 };
13 //map-task for file ID extraction
14 $d= transform $d into {
15   fileId: xGet($data, "/.@Filename", "f(\d{6})\.xhtml", "$1"),
16   HTML: $d
17 }
18 //map-task for formulae extraction
19 $f = xtransform "//math" as $m in $d.HTML into{
20   $fileID: $d.fileID,
21   $mml = $m,
22   $tex = xGet($m, "/.@altText")
23 }
24 //map-task formulae tokenization
25 $f = addFields $f { vars = $f.TeX ? TeX2Bag($f.TeX):mml2Bag($f.
    math) };
26 $d = addFields $d { words = HTML2Bag($d.HTML) };
27 //reducer for counting total number variables
28 $vars= bag_union $f[*].vars;
29 $words = bag_union $d[*].words;
30 //cross-task for formulae filtering
31 $fq = formulafilter $q in $f;
32 $dq = textfilter $q in $d;
33 //cross-task that assigns scores with a TFIDF-like method
34 $fq = score $fq use $vars;
35 $dq = score $dq use $words;
36 //join-task: increase formula by document score
37 $fq = raisescore $fq use $dq where $fq.num=$dq.num and $fq.fileid=
    $dq.fileid;
38 //reduce-task concatenates results for each query
39 $results = formulagroup $fq by $fq.num sort desc by $formulae.
    score limit 30;
40 //reduce-task formats results
41 $result =format results $results parallel 1;
42 write $result to "hdfs://localhost/results.xml"

```

Listing A.1: Meteor pseudocode: Script for answering the fulltext search task. For a detailed description of the operators cf. Section A.2.1.3

Aggregation Operators As a next step, we calculate the total number of words and variables in each document via bag union (cg. Listing A.1 [25-26]). The bag union operator is implemented as a combinable *Reduce* contract. This means, it merges the multi sets containing the words or variables of a document or formula in two steps. First, all nodes compute the union of their local multi sets. Then, these intermediate multi sets of all nodes are united. We use the multi set implementation of the guava libraries⁹ which turns out to be quite efficient with regard to memory usage and runtime. As a result there are two records, one for the variables and one for the words. Each contains one field with a multi set of the counts.

Filter Operators The crucial part of the execution is the formula filtering that identifies the match candidates for the queries based on the formula tokenization. Logically this works is shown in Figure A.4. Technically, the formula filter *Cross* task gets a tuple of records as input that originates from the Cartesian product of query and formula records. If the multi set of the tokens of the formula is a superset of the tokens in the query, the formula is regarded as hit candidate. In that case the fields of the query record and the fields of the formulae record are concatenated and emitted. Otherwise, other nothing is emitted.

We use a similar *Cross* task for the filtering of the documents. Only documents that include the keywords, specified in the query, pass the filter. For those, we emit a record that contains the concatenation of the query and document fields.

Scoring Operators After the filtering step the result set is still too large and has to be ordered. Therefore, the scoring task is used. Since we want to use Term Frequency Inverse Document Frequency (TFIDF) like methods to score the tokens, we need to know the total number of occurrences of the tokens.

Foreclosing the evaluation, the following statement can be made: We developed a scoring mechanism that calculates a score based on the absolute and relative number of tokens, Presentation and Content MathML, as well as the occurrence of the keywords. However, the ranking results are poor, because free parameters that are used to calculate a reasonable norm for this score vector, were specified with an ad-hoc-method.

A future research task, is to adjust the free parameters in a way that the ranking results correlate better to the experts relevance ratings. Certainly, during that task over-fitting must be avoided.

For the scoring of the MathML part, the MathML tags were compressed in same way as done for the queries. After that, the regular expressions are applied to the compressed MathML. Thus, even the instances of the placeholders were specified which is valuable for displaying the results in a future use case.

⁹<http://code.google.com/p/guava-libraries/>

We demonstrate the scoring function using the sample query FT-14 and the relevant formula $\sum a_n b_n$ 190/f075790 .xhtml#id73874 that is ranked best. The final score of 10155 is calculated in the following way.

Since all tokens are found in the TeX code a score of 100 is assigned. In addition, the score for the Content MathML match (with $\$1=\text{ci}[\text{a}]$ and $\$1=\text{ci}[\text{b}]$) was scored with 5000. Additional 5000 points are assigned, because there are no other expressions in the Content MathML representation. Presentation MathML did not match by accident, because there is an additional invisible times between a_n and b_n that is not removed while compressing the Presentation MathML. The seven occurrences of the word **convergence** are rated inspired from the geometric series with $2^{-7}(2^{7+1} - 1)$ wordscore(convergence). Here, the word score is obtained from corpus wide frequency of **convergence** in relation to the sum all relevant word frequencies. In the same way token scores for $n, \sum, \backslash, -$ are calculated.

Result Operators There are two result operators. The first one groups formula that have the same query number and the second one summarizes the result of all queries.

The input for the first *Reduce* operator is sorted according to the score specified in descending order. We modified the normal grouping in the way that the operator only collects the first 30 input records and normalizes them according to the NTCIR submission guidelines.

In former steps, the results of all tasks are concatenated to the XMLresult file that was downloaded from the HDFS and was sent directly to the NTCIR office without manual modifications.

A.2.1.4 Limitations

We designed our system in a way that it produces deterministic and traceable results for the given queries. Our conservative approach implies that the result set must be explainable and reproducible. This allows for easy debugging and testing of the code. However, verbose justification statements lead to a reasonable amount of development work, and influence the performance of the system in a negative way. Especially early selection of the probably best results and random sampling gets impossible with our approach. Furthermore, no query expansion is performed. For example, if the query is x^2 , but no equation that contains x^2 exists, the system would answer with an empty result set, rather than displaying results for x or y^2 . For that reason, some of the results that were considered as partial match by the experts could not be found by our system.

Furthermore, we were not able to discover equations that involved a linebreak in the MathML source code which is a corner case. Additionally, only equations that con-

tained an `alttext` attribute were considered. After fixing these bugs the performance of our system has slightly improved.

For the fulltext search, we interpreted the text input as a list of keywords. One query contained the word `not` (`parseval`) which probably means that the word `parseval` should be excluded. However, in this first version of our system, we did not treat this special case.

Furthermore, we ignored the task FT-3 that deals with searching for LaTeX-pseudo-code.

A.2.1.5 Optimized Execution

Instead of using the Meteor to compile the source from Listing A.1 automatically to the Pact layer, we hard code the PACT plan (cf. Figure A.5). In this context, we perform some manual performance optimizations and customizations. For example, we just count the keywords and tokens that occurred in one of the search queries rather than to compute TFIDF for the whole document collection. Furthermore, we summarize concurrent *Map* tasks, like for example extraction of the `TeX` code and the tokenization of it. In addition, we specified compiler hints for output cardinality estimation. Especially, for choosing *Cross* and *Match* strategies these estimates make a difference. The query list has about 30 entries and is much smaller than the number of equations with more than 130 million records. Thus, a broadcasting of the queries and keeping them in memory for the whole time the task runs makes sense.

A.2.2 Results

We, the **formulasearchengine** team (FSE) submitted 373 results in the category formula search (FS) and 244 results in the category fulltext search (FT).

In the category FS 290/373 results were judged. 106 of the evaluated results were regarded as (partially) relevant. Thus, the precision (relevant/submitted) results evaluates to 28.4% for formula search which is rank 2 of 13 submitted result sets.

For the fulltext search task all 244 submitted results were judged. 54 of them were regarded as (partially) relevant and the precision evaluates to 22.1%. In this category only two teams participated. The other team achieved 31% precision.

Even though we do not perform a detailed performance analysis, we present some indicators of the system runtime. For the actual run, our system consisting of two desktop computers with 8 CPU cores 1 HDD and 16GB main memory per machine, reported an overall runtime of about 9 minutes (512 207ms). As an indicator for the shortest possible runtime on our system, we read all the data and write the first 1000 bytes of each file back to disk. This took less than 4 minutes (226918-228194ms) and

meets our expectations, since processing the 42 GB dataset with a single disk, and an average effective data transfer rate of 100MB/s leads to 7 minutes reading time in theory.

For a detailed analysis of the results, we performed another run that ignores all results that were not rated by the experts. With this configuration, we generate detailed output, even if the formula would have been suppressed by the filter. We published this output at <http://www.formulasearchengine.com> (cf. Figure A.6) in full length and analyze the crucial aspects in this section.

Our score s that was used for ranking is partitioned in three categories and relates to the experts ratings relevant ($++$), partially relevant ($+$) and not relevant (o) via

$$\begin{aligned} s < 50 &\rightarrow \text{no match} && \leftrightarrow o \\ 50 \leq s < 2000 &\rightarrow \text{token only} && \leftrightarrow + \\ s \geq 2000 &\rightarrow \text{token} + \text{filter} && \leftrightarrow ++. \end{aligned}$$

In Table B.1 to B.3 and Table B.4 to B.5, 3×3 matrices (M) for all results show the relationship between expert rating and the classification of the system score. The diagonal entries denote that our system and the reviewers agree, entries in the upper triangle of the matrix mean that our system rates the result as more relevant than the reviewers did, and entries in the lower triangle denote the opposite. They relate to the classical 2×2 binary confusion matrix (B) by summarizing the non corner entries via

$$\begin{aligned} B_{i,j} = M_{2i,2j} + \alpha M_{2i,2j+1} + (1 - \alpha) M_{2i,2j-1} \\ + \beta M_{2i+1,2j} + (1 - \beta) M_{2i-1,2j}. \end{aligned} \quad (\text{A.6})$$

For example, summarizing as well relevant ($++$) as partially relevant ($+$) to relevant ($\alpha = 1$) and regarding all entries beginning from token only (score > 50) as retrieved entries ($\beta = 1$), leads to $B_{0,1} = M_{0,2} + M_{1,2}$ false positives. The β parameter is not relevant for the further discussion since, the calculated score is more fine grained than shown in the table. Thus, we calculate the average precision $\langle P \rangle$ defined as $\langle P_\alpha \rangle \equiv \int p_\alpha(r) dr$. For $r \in R_\alpha$, the set recalls of the ranked result list $p_\alpha(r) \equiv \max\{p'_\alpha(k) : r = r_k\}$ is the maximal precision for $p'_\alpha(k)$ the precision for rank k with recall r_k . Since, this are only discrete values, $\langle P_\alpha \rangle$ is approximated in upper Riemann sum style by using $p_\alpha(r) \equiv p_\alpha(\min\{r' \in R_\alpha : r' > r\})$ for $r \notin R_\alpha \wedge r < \max R_\alpha$ and $p_\alpha(r) \equiv 0$ for $r > \max R_\alpha$.

In the tables, we use the more intuitive notation $\langle P_{++} \rangle = \langle P_0 \rangle$ and $\langle P_+ \rangle = \langle P_1 \rangle$. Furthermore, the Mean Average Precision (MAP) $\langle \langle P \rangle \rangle$ was calculated by averaging over all $\langle P \rangle$ for each subtask. Comparing the result of $\sim 30\%$ for FS and $\sim 15\%$ for FT shows that especially the combined search needs to be improved and indicates that keyword search is not sufficient.

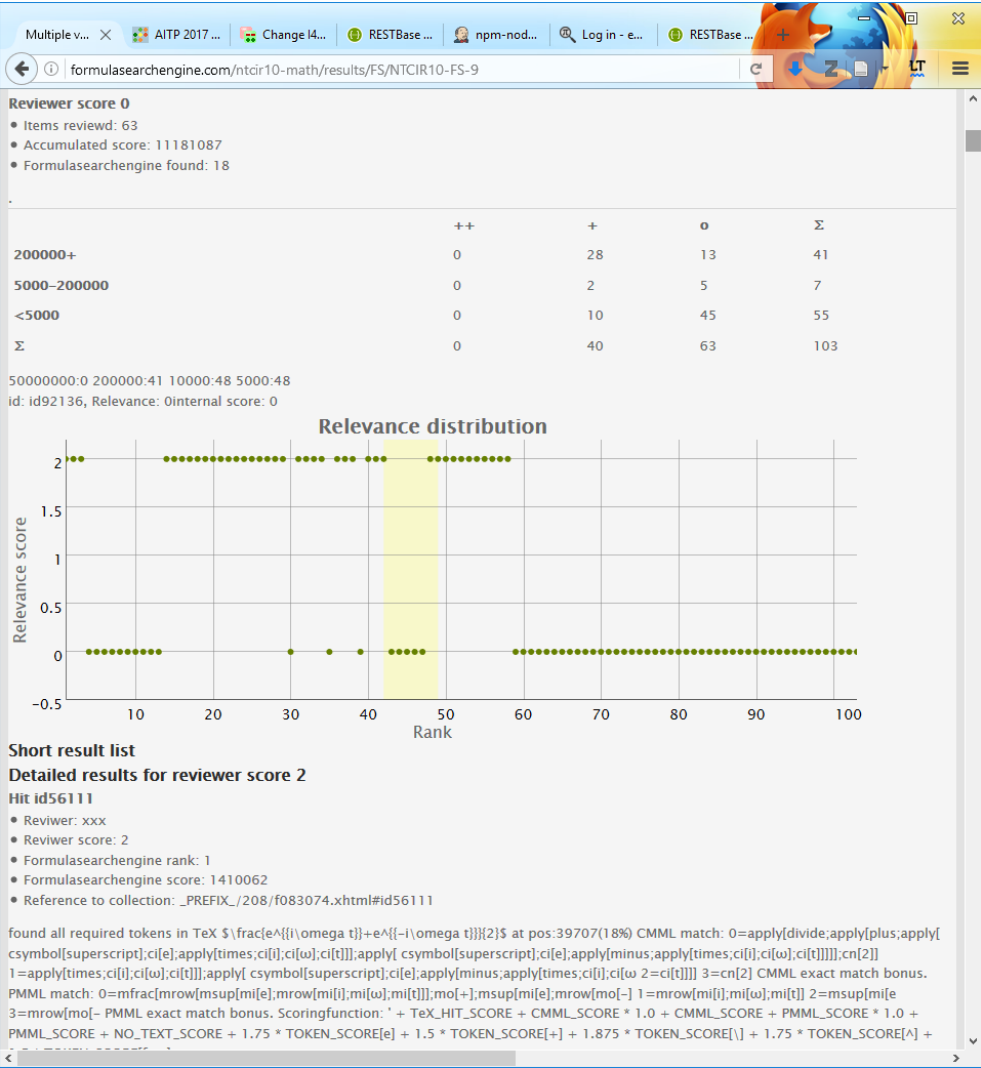


Figure A.6: Full length of the output of the analysis of the topic results. The visualization in the middle shows the assessment with regard to the ranking.

A.2.2.1 Qualitative Evaluation

In the following subsection, we explain the results based on some examples. Therefore, we group our observations into false negatives and false positives.

False Negatives Regarding our running example query (FT-14), the formula 132/f052473#idp561824: $\sum_{\alpha \in \mathbb{F}_n^+} a_\alpha Z_\alpha$ did not pass the token filter (cf. Figure A.4), since the query contains two n , whereas the formula has only one. However, it is one of the seven relevant formulae. This is an argument for lowering the filter barrier further in the future. One option is to use a set rather than a multiset. On the other hand 76 of the 100 evaluated formulae passed the filter. Thus, adjusting the filter granularity on a meta level might not be the option of choice. Looking at more examples indicates that it might be reasonable to improve the filter on a token level. For example, the token `/` could be regarded as a match for the token `frac` as well. According to B.514, five additional entries were ranked in the intermediate category. Three of them originate from document 15/f005755 that does not involve the specified keyword *convergence* and were down ranked even though they are a perfect structure match. The other two use a specified sum ($\sum_{n=1}^\infty$) rather than the unspecified \sum which prevents them from a MathML structure match, since this would require query expansion.

False Positives The two best-ranked false positives ($\sum \lambda_n Q_n, \sum \epsilon_n x_n$) for the running example originate from documents that contain all required keywords. However, the hits were considered as not relevant by the experts. The reason for that can not be determined without considering the context of the equation. Therefore, it will be important to consider the context of the hits in the future. The same argument holds for the second best ranked hit $\sum a_n b_n$ that was classified as partially relevant. Due to the fact that the TeX filter contains only two symbols (cf. Section A.2.1.3) the relative large number of 63 false positives for the filter is obvious.

A.2.2.2 Ranking

Even though we do not focus on ranking, the results are reasonable, especially for cases where more than 10 relevant results were found by the experts. In Table A.1, we list those queries and print out the experts ratings compared to the ranking position. According to this measure, most results are quite relevant. However, since only 6 of 22 (for FS) queries lead to ten or more relevant results, it has to be remarked that this good ranking holds for the ‘easy’ queries. In a further step we investigated the impact of the fine ranking. For the detailed evaluation in tables (cf. Table B.1 to B.3 and Table B.4 to B.5), we calculated the MAP based on a numeric score with two internal decimal places. Rounding to natural numbers has a large effect on the overall result. Our evaluation shows that just improving the inner decimal digit score can lead to an increase in MAP for almost 50%, ($\langle P_{++} \rangle$) increased from 31.5% to 46.6%

Table A.1: Overview of the top 10 results: The queries that had less than 10 relevant results are not displayed. The symbol ++ denotes relevance, + partially relevant and o not relevant.

#	formula search (FS)						fulltext search (FT)			
	5	8	16	18	20	21	1	8	9	15
1	++	++	++	++	++	++	++	+	+	++
2	++	++	++	++	+	++	++	+	+	++
3	++	++	++	++	++	++	o	+	+	+
4	++	++	++	++	++	++	++	++	+	o
5	++	++	++	++	+	++	++	++	o	++
6	++	+	++	++	+	++	++	+	o	++
7	++	++	++	++	+	++	++	+	o	++
8	++	++	o	++	o	++	++	o	o	o
9	++	++	++	++	o	++	++	o	o	o
10	+	++	++	++	o	++	++	+	o	++

and $\langle\langle P_{++} \rangle\rangle$ increased from 28.5% to 44.4%). This indicates that TFIDF methods have to be checked carefully in the future.

A.2.3 Conclusion

In this section, we present a batch-oriented approach to formula search that is characterized by its short development test cycles. We achieved rank 2 of 13 in the formula search subtask with regard to precision of partially relevant hits. However, the achieved precision of our system is not satisfying yet. In general, our two phase approach of token filtering and structure matching seems to be viable and will be investigated further.

Our main focus for future work will be query expansion. Given a pattern that includes the identifier i , it would be beneficial to consider formulae that include the identifier j , if j is known to be the imaginary unit (cf. Section 3.2 for details). The first requirement for this kind of query expansion is the integration of variable definition detection in the surrounding text which requires deep parsing of the text and analysis of the equation. This way, our approach can incorporate valuable information that would be lost, if only equations are considered. Due to the absence of precomputed structures, such as indexes, this extension is comparatively easy to implement in the next iteration of our system.

The second requirement for query expansion is a clear definition of the actual semantics (cf. Section 3.1) of the query patterns which was not available for this competition.

Additionally, we believe that query patterns should provides means to tag identifiers with meta information, such as ‘ i is the imaginary unit’, so the identifiers in the pattern can be matched better to the identifiers in the corpus.

Replik According to Guidi and Sacerdoti Coen our system

... trades flexibility with performance, and it is essentially math-unaware (for example, it does not normalize the input in any way)[84].

Unfortunately, the authors refrain from defining ‘math-awareness’ of a system. However, we admit that our system was designed with the sole purpose to process data and retrieve documents or mathematical expressions respectively.

A.3 Mathoid: Formulae Processing for Wikipedia

Pixelated images of mathematical formulae, which are inaccessible to screen readers and computer algebra systems, disappeared from Wikipedia as of May 2016. In this section, we describe how **Mathoid** matured from a research prototype to a production service and describe a novel visual similarity image comparison tool designed for regression testing mathematical formulae rendering engines. Currently, updates to math rendering engines that are used in production are infrequent. Due to their high complexity and large variety of special cases, developers are intimidated by the dangers involved in introducing new features and resolving non-critical problems. Today’s hardware is capable of rendering large collections of mathematical contents in a reasonable amount of time. Thus, developers can run their new algorithms before using them in production. However, before now they could not identify the most significant changes in rendering due to the large data volume and necessity for human inspection of the results.

The novel image comparison tool we are proposing will help to identify critical changes in the images and thus lower the bar for improving production level mathematical rendering engines.

In [207] (cf. Section 3.3), we analyzed different ways to improve math rendering in Wikipedia and presented our solution, called Mathoid, to the problem. However, it took more than two years until the majority of users could benefit from the improvements in rendering. During this period many bugs were reported and fixed. Those bugs can be categorized into two main concerns; performance and layout. While improving performance is relatively straightforward to measure, improving the layout is still an open problem [149, 34, 150, 249]. In particular, any work in this area is hindered by the issue of ensuring that improving the layout of some aspect of math

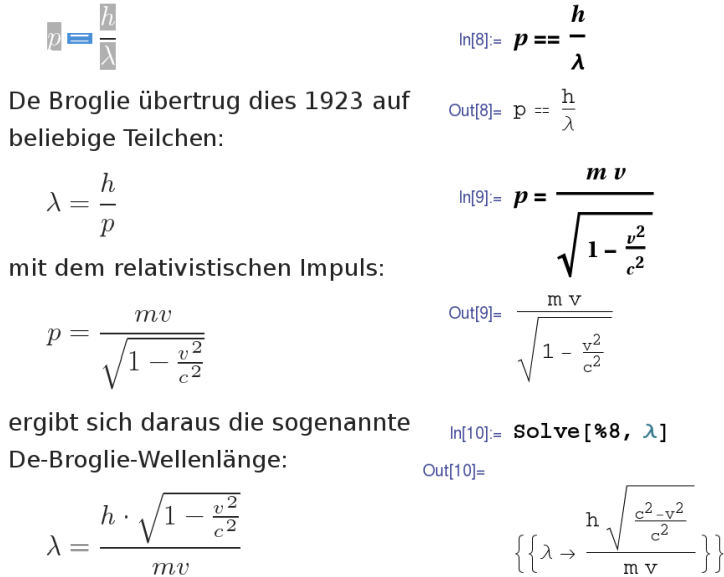
rendering does not negatively impact that of others. When such regression testing requires humans to visually inspect and compare tens of thousands of images, progress on layout can be slow and error prone.

In this section, we present a method to automatically compare images of mathematical formulae generated by different rendering engines and thus automate visual regression testing in this domain. Our section is structured as follows: We begin by presenting an overview of the improvements over the old rendering, then we describe the request flow of the new rendering process in detail and analyze its performance. Thereafter, we describe **Mathpipe**, the tool we built to compare different rendering mechanisms at scale and finally present the current state of our image comparison tool. Since this is work in progress we encourage the reader to visit <http://png.formulasearchengine.com> for the latest updates.

A.3.1 Mathoid’s Improvements over texvc Rendering

As described in [207] (cf. Section 3.3), Mathoid provides ‘Robust, Scalable, Fast and Accessible Math Rendering for Wikipedia.’ In 2014, Mathoid was one of the first Wikimedia nodeJS services supporting the MediaWiki extension Math which takes care of the handling of mathematical expressions used in Wikipedia and other websites running MediaWiki. Today, Mathoid is constructed from a huge number of services such as **Citoid** and **Graphoid** which support the MediaWiki extensions **Cite** and **Graph** respectively. All those service-supported extensions add complex functionality to MediaWiki which would be hard to reimplement with native and efficient PHP code. Moreover, they all share the goal to be ‘Robust, Scalable, Fast and Accessible’. The so called **service template** provides a common ground for robustness and scalability and reduces the maintenance effort. For the math extension that means that no binaries need to be installed on the MediaWiki servers and no files in the file-system are created. Thus, **robustness** is improved with respect to the old approach that relies on the texvc binary and creates local files [207] (cf. Section 3.3) which uses MathJax [45] rather than LaTeX for the rendering. Note, that the set of supported ‘LaTeX like macros’ is exactly the same for the old and the new system. With the service template, scaling the Mathoid service is simple. As an improvement over the original version of Mathoid, presented in [207] (cf. Section 3.3), now all formulae of a page are processed in parallel (cf. Section A.3.2). That way, the page loading time is determined by the formula that takes the longest time to render and no longer by the sum of all the rendering times. See Section A.3.3 for the time measurements of individual formulae. However, most significant to the user are the accessibility component and the change in the layout itself. After the new rendering was tested on beta clusters, it has been enabled on Wikidata and the German version of Wikibooks on May 16th, 2016. The rest of Wikipedia is scheduled to follow shortly after that date.

As visualized in Figure A.7, the MathML code is available to the browser. This



De Broglie übertrug dies 1923 auf beliebige Teilchen:

$$\lambda = \frac{h}{p}$$

mit dem relativistischen Impuls:

$$p = \frac{mv}{\sqrt{1 - \frac{v^2}{c^2}}}$$

ergibt sich daraus die sogenannte De-Broglie-Wellenlänge:

$$\lambda = \frac{h \cdot \sqrt{1 - \frac{v^2}{c^2}}}{mv}$$

Input in Mathematica:

```
In[8]:= p == h/lambda
Out[8]= p == h/lambda

In[9]:= p == m v / Sqrt[1 - v^2/c^2]
Out[9]= m v / Sqrt[1 - v^2/c^2]

In[10]:= Solve[%8, lambda]
Out[10]= {{lambda -> h Sqrt[1 - v^2/c^2] / (m v)}}
```

Figure A.7: Copying MathML expressions from the German version of Wikibooks to Wolfram Mathematica: On the left, there is a screen shot displaying a section of the book on quantum mechanics taken on May'16 2016 (the day MathML rendering became the default rendering mode for mathematical formulae on Wikibooks) as non registered visitor using Firefox version 45, with the native MathML plugin enabled. The photon momentum was selected to demonstrate the bounding boxes of the symbols, and to copy and paste it to the computer algebra system Mathematica (screen shot on the right) as `In[8]`. The same step was performed for the relativistic momentum `In[9]`. Thereafter, an additional `=` sign has been manually inserted to `In[8]` and `Solve[%8,lambda]` was typed to compute the De- Broglie wavelength.

allows screen readers to verbalize the formulae from the additional information that is neither available from the new Scalable Vector Graphics (SVG) nor from the old Portable Network Graphics (PNG). The documentation page of the Math extension contains links to examples of this feature and [50, 47, 141, 217, 220] provide further information on that topic.

However, since MathML requires certain fonts that are not available on all systems, MathML will only be provided to people that have installed a browser extension that indicates that their browser actually provides good MathML rendering. This could be either Mozilla Firefox with the native MathML plugin [242] or Internet Explorer with the MathPlayer plugin [217]. The majority will see SVG, which is already an improvement for people with high resolution displays, or on stations where the formulae are printed.

3. All math elements on a page are collected.
4. A bundle of requests (the size of this bundle is a performance tuning parameter) is sent to Restbase (check request).
5. **Restbase** checks the requests; either the result is cached in the internal Cassandra data store
6. Or it contacts Mathoid which calls Texvcinfo which calls Texvcjs (this will be explained later in detail).
7. Restbase returns the results of the check and other information about the formulae, i.e., the sanitized Tex, a hash and the SVG.
8. After a bundle of check requests is returned to the MediaWiki, the extension evaluates each check response, and either replaces the math tag with an error message in case the check request was not successful or collects the hash of the MathML and SVG.
9. Now, the math extension has collected the hashes of all valid input formulae and a second request bundle is sent to Restbase requesting the MathML rendering.
10. Restbase responds with the MathML rendering from storage. In the header of the request response, the SVG dimensions are stored.
11. Thereafter, the math extension replaces the math tags with MathML and a link to the SVG fallback image. The style sheet is configured in a way that the fallback image is visible by default and the MathML element is invisible.
12. Finally, MediaWiki returns the HTML of the page, including the links to the SVG fallback images.
13. If the browser of the user does not overwrite the visibility information of the MathML and SVG, the browser sends a request for each individual formula.
14. While most of the image requests will be served from either the browser cache or varnish (a caching HTTP reverse proxy [234]), the rest will go directly to Restbase without contacting MediaWiki.

A.3.3 Measuring Mathoid's Performance

To evaluate the absolute performance of the new rendering, we performed measurements with the following setup: We use two identical work stations (Intel(R) Core(tm) i7-3770 CPU @ 3.40GHz 16 GB RAM, Gigabit Ethernet, 2x1 TB HDD, Ubuntu 14 LTS). One system (hereafter referred to as the server) has the Cassandra storage engine, the Restbase client and Mathoid. The other system (client) has MediaWiki with the extension math and MathSearch. It uses as many parallel workers as CPUs are

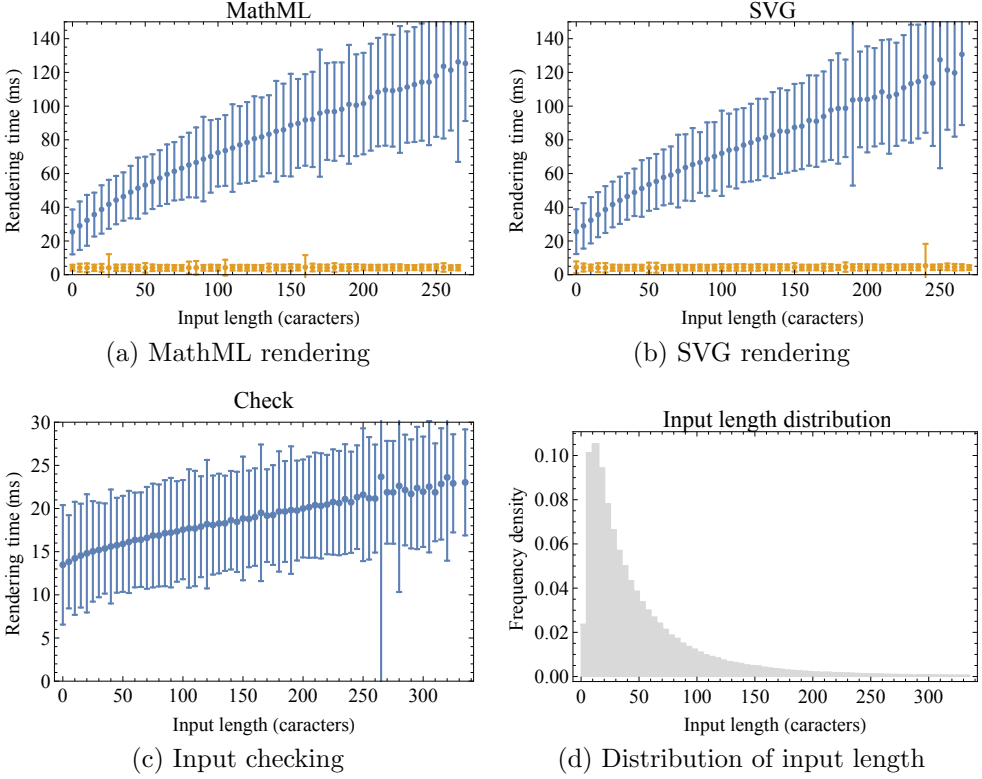


Figure A.9: Rendering and verification time versus input length: Measurements for the test collection, with different rendering modes. (a-c) shows the rendering, times and standard deviations respectively. (a-b) include two series, where the yellow (lower) series is the cached result.

present (8) to call to get the data as would be done in production. Our performance tests script can be downloaded from [githubscript](#).

For each formula, the requests are sent in as described in A.3.2 and the time is measured. Note that, for this test, we did not use the complete endpoint but the MathML or SVG endpoint respectively. For each formula we randomly chose, if it was rendered first in SVG or MathML mode. The measurement results are visualized in Figure A.9. The figures indicate an almost linear relationship between the formula length and the rendering time. It should be noted that formulae whose request could be answered by Restbase from the Cassandra storage without contacting Mathoid were fast, independent of the length of the input Tex string. The average was around 4ms. These measurements indicate that, in the average case, where already rendered formulae get requested again, most of the time is spend at the checking phase (>20 ms). Therefore, the most recent version of Restbase which has not been tested yet, also

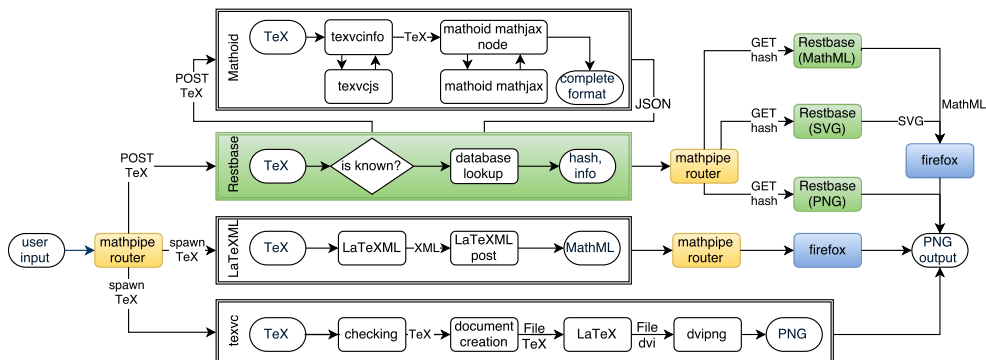


Figure A.10: Mathpipe processing chain

caches the check responses from Mathoid. This is supposed to reduce the checking significantly. Compared to the old LaTeX-based rendering, which created a LaTeX document for each formulae, rendered that document and thereafter converted it to a PNG, this is a significant performance improvement.

A.3.4 Comparing Different Rendering Engines with Mathpipe

Mathpipe is a program specifically designed for testing the different options for rendering mathematics on the web. It starts with user input (currently limited to texvc dialects) and goes via different routes to PNG output. All current routes are visualized in Figure A.10. From there on, we perform the analysis of different generated PNGs and calculate the similarity scores as described below. Mathpipe has two modes of operation. A command line mode that can be used for batch jobs and a web interface. The command line will be used to identify the formulae that have the biggest divergence. The web interface will provide a quick comparison for humans to manually investigate the difference outputs and check the derived results. We encourage the reader to visit <http://png.formulasearchengine.com> and test an instance of our service.

A.3.5 Image Comparison

The lack of practical automatic regression testing on Mathpipe-generated renderings of the huge collection of mathematical formulae in Wikipedia has been a serious hindrance to speedy improvement of rendering quality and performance. Currently, any change made to the rendering pipeline can only be checked visually by humans. Even then, it is very easy for a human inspector to overlook some small but important errors that has arisen in the rendering. The motivation for automating this procedure with a suitably high precision machine-based approach is clear. The problem

to be solved involves comparing two PNGs rendered using different methods from the same input source. These methods vary in their choice of fonts (and therefore, also the thickness of the character strokes), the resolutions of rendered images, the approaches to anti-aliasing and background transparency rendering, spacing between characters and relative sizes of characters. Some of the methods also generate extra empty padding around the images and most generate some form of gray scale or color images, the details of which can vary rather than monochrome. In comparing these images, our ideal is to make judgments about the relative validity of these renderings as an extremely careful human inspector would. Differences insignificant to the readability and correct interpretation of the underlying mathematical formulas should be overlooked. For example:

- Changes of font where corresponding characters from the two images are visually very similar.
- Differences in character spacing in the two images which are within aesthetically reasonable bounds.
- Characters that are spatially discrete but close together in one rendering may touch in another. This is often due to the rendering resolution chosen together with the antialiasing approach used. If the resulting touching characters are perfectly readable, then this should not cause significant concern or trigger reporting of errors unless the user is explicitly looking for such problems.
- With symptoms very similar to the previous case, a single character in one image may be broken into two or more characters in the other. This arises not from characters touching, but by the renderer building a character, often an expendable fence or integral character out of separate character components. The renderer is supposed to make such characters touch, but may fail to do so completely. Such a case rarely interferes with the readability of the image, but should be overlooked or highlighted as the user requires.

Conversely, significant issues should be identified and reported:

- Characters in one image but missing from the other.
- Characters that have significantly different shapes, for example if the character is missing or mis-indexed in the font used in one of the images.
- Significant differences in spacing that may confuse the reader, e.g., a superscript that is rendered on the same baseline as a superscript.
- Touching characters that are unreadably overlapped.
- Broken characters that are so spatially separated that it confuses the reader.

The tolerances used to determine many of the fuzzy quality issues (i.e., ‘**visually similar**’, ‘**aesthetically reasonable**’, etc.) described above, should be selectable by the user to correspond to the user’s purpose at the time, although defaults should be set to practical values for general purpose regression testing.

In general, the system should be biased to not overlook serious issues, even if that means that more insignificant differences are reported as significant (false positives).

Finally, a practical and robust way of reporting issues found is necessary to assist developers in quickly identifying problems and their sources.

A.3.5.1 Approach

The constraints of a solution as described above mean that a purely image-focused approach is unlikely to be successful. Instead we have taken an approach based on a structural analysis of the image into a form where we can deal with the image components and their relationships more abstractly. This form is based on **connected components**. A connected component in a monochrome image is a maximal set of foreground pixels that are horizontally, vertically or diagonally connected. Thus, an equals symbol, ‘=’ and the letter ‘i’ both have two connected components, while an equivalence symbol ‘≡’ and a capital greek letter xi, ‘Ξ’, both have three. In brief, the general approach we have chosen involves binarising the images, decomposing them into sets of connected components, scaling the meta-data (bounding-box information) about the connected components to make them comparable between the images, identifying viable pairings of corresponding connected components in the different images based on relative positions, aspect ratios, and size, verifying the pairings using features of the underlying connected component shape and pixel distributions, and finally treating connected components not successfully paired as possible candidates for touching/broken character analysis. It should be noted that, though we use techniques borrowed from the areas of document analysis and Optical Character Recognition (OCR) [62], we do not attempt to classify characters or interpret the mathematical formulae in any way, and hence we avoid the problems of classifier training, mis-classifications and incompleteness of a model for the structure of mathematical formulae. More precisely, we borrow only methods of binarization, connected component analysis and some simple shape feature extraction methods that are commonly used in document analysis for character classification purposes but here are used only to provide a basis for metric shape similarity measurements between corresponding components of the two images. We shall now discuss each part of the approach in detail.

A.3.5.2 Binarization and Connected Component Analysis

Connected component analysis [175] requires binary choices between foreground and background images. Since these images are generated rather than scanned, a very simple binarization method selecting the pixel class-based only on the RGBA (red-green-blue-alpha) value of the pixel itself is sufficient. The only issue of (minor) concern is the different approaches to background colors and anti-aliasing. Background pixels and anti-aliasing in an effectively monochrome image is best implemented by using the same (foreground) color for all pixels but varying the transparency of the anti-aliased pixels down to fully transparent for background pixels. However, some images do change the gray-scale/color as well as the transparency while others do not use transparency at all but merely blend the foreground into a different background color. Binarization must be aware of these variations and manage them all. Connected component analysis is accomplished using the algorithms described in [92, 93].

A.3.5.3 Cropping and Scaling

Removing extraneous background padding is a simple matter of cropping to the size of the rectangle union of the bounding boxes of the identified connected components. Scaling is slightly more subtle. Since these images can be of relatively low resolution, the size of a pixel relative to the whole connected component can be significant. Hence, scaling the image to a common size can introduce discretization artifacts that impede the analysis. Hence, the images themselves are not scaled but a scaled version of the connected component bounding box information is added. The scale factors are chosen so that one of the images' information is scaled to correspond to an image of width 1.0 (using floating point rather than integer numbers of pixels) and a height that maintains the original aspect ratio. The other image is scaled so that the image is exactly the same size, even if that distorts its aspect ratio. This results in relative positions of connected components in the scale spaces being directly comparable. The underlying image pixels are not changed so shape analysis is not affected by the scaling.

A.3.5.4 Simple Component Pairing

At this point we attempt to pair off connected components in one image with the corresponding ones in another. A simple matching of corresponding positions does not work for a number of reasons:

- While scaling ensures that the leftmost and rightmost characters are in very close to the same horizontal position, variations in spacing may mean that characters in the centre of the image are significantly out of horizontal alignment with

the corresponding character of the other image, and, indeed, may be in exactly the same relative position as a non-corresponding character. Ditto for vertical alignments.

- Multiple characters in one image may be touching and therefore occur as a single connected component while they are separated in the other. Therefore, the correct pairing should be of at least one connected component with a set of connected components.
- Depending on the actual parameters used, a component of one image may viably, but incorrectly, be paired with any one of a number of different components from the other. We call such a case a ‘**multi-match**’.
- Because of variations of character fonts, positioning and sizing, any choice of discrimination parameters that correctly accept/reject a pairing in one part of the image tends to be wrong for another part of the image. An adaptive approach that varies the parameters over different parts of the image might be possible but it is not clear what criterion can be used to accomplish it without more in-depth classification or recognition.

For these reasons we chose an iterative approach, where we start with very tight constraints on acceptable parameters to pair components based on their position, aspect ratio and size (i.e., area of the bounding boxes), with a verification element based on a metric shape similarity measure to guarantee reliability of the pairing. This ensures that any pairings found are robust and can be removed from consideration. Repeating the process with slightly relaxed parameters on the thinned out set of remaining components allows robust pairings to be made where, with the full set of components, an unambiguous pairing would not have been possible. This cycle continues until one of the following holds:

- No unpaired components remain (in which case the two images can be considered to have passed the comparison check) or
- Any further pairings require relaxing the parameters beyond their upper limits or
- There is a component which, at the current parameter setting, could viably be paired with more than one component (a multi-match).

In the latter two cases further analysis is necessary.

A.3.5.5 Touching Component Analysis

At this point, assuming there are multi-matches or unpaired components remaining, there is a set of components from each image that could not be paired with components from the others. The only allowable remaining situation that would not justify

reporting this as an error, is if components are touching in one image but not in the other, and there may be multiple separate cases involved. Simply trying all possible combinations of ways that components could touch is computationally infeasible for our purposes. To find such cases, note that a touching component in one image that corresponds to a group of components from the other is necessarily larger than the individual sub-components. Therefore, we work iteratively starting at the largest remaining unpaired component of **both** images, the **target** component, and select the set of unpaired components (the **candidate** components) from the other image that overlap with an expanded version of the bounding box of the target component. This excludes from consideration components that should not realistically be considered as candidates, but the expansion allows for some distortion of the spacing between the different images. Even if the target does correspond to some of the candidate components, it may not correspond to all. For example, consider the following expressions where the left is from one image and the right from another:

$$\sqrt{x^K} \quad \sqrt{x^K}$$

Here the target would be the touching $\sqrt{x^K}$ component from the right and the candidate set would include all three components from the left, because all three are within the appropriate space. However, the x should not be included in the pairing or it will cause the shape matching to fail.

For these reasons, all combinations of the candidate components are considered by calculating the comparison attributes of the union of each combination, where the attributes involved are; center of the bounding box, aspect ratio, area of bounding box and, only if the checking of those attributes passes, the more expensive shape similarity test. The final shape similarity test ensures that the resulting merged shape is still readable. The limitation of the set of candidate characters to those that overlap the expanded target component bounding box ensures computational feasibility. If a pairing is found, the components are removed and the process repeats on any remaining unpaired components until exhaustion of unpaired components or failure to find a pairing.

A.3.5.6 Reporting of Results

The results of the above analysis is four-fold:

1. **Simple matches:** A set of pairs of single compatible components that are within appropriate matching parameters.
2. **Touching matches:** A set of pairs of sets of components corresponding to target/candidate set matching pairs for touching component cases.
3. **First image unpaired:** A set of components from the first image that could not be paired with corresponding components from the second.

$$P = 3B_0 \left(\frac{1 - \eta}{\eta^2} \right) e^{\frac{3}{2}(\textcolor{blue}{B}'_0 - 1)(1 - \eta)}$$

$$P = 3B_0 \left(\frac{1 - \eta}{\eta^2} \right) e^{\frac{3}{2}(\textcolor{green}{B}'_0 - 1)(1 - \eta)}$$

$$P = \mathbf{3}B_{\mathbb{0}} \left(\frac{1 - \eta}{\eta^2} \right) e^{\frac{3}{2}(\textcolor{blue}{B}'_0 - 1)(1 - \eta)}$$

Figure A.11: Error images from two different renderings of the same formula. The original of the second image has been artificially edited to force the B' characters to touch, as indicated by the green color. In the first image the two characters that form the matching candidates are in blue. The third image shows the result when the two images are scaled and overlaid using the GIMP plugin to demonstrate differences in the spacing and character shapes between the two images.

4. **Second image unpaired:** A set of components from the second image that could not be paired with corresponding components from the first.

The test passes if all except **simple matches** is empty. It can be considered to pass if **touching matches** is also non-empty and the user chooses that option. A short narrative report is generated of the results to a log file or to the standard output stream. However, a textual description of the problems when errors occur is frustratingly difficult to interpret. Hence, we also generate two error images. These are cropped binarised images but with unpaired components drawn in one color, target components of touching matches in another and the corresponding matching candidate components in a third. This is usually sufficient for immediate identification of the problem to the user. However, sometimes it is necessary to investigate more directly the relative positioning or sizing issues that triggered the comparison failure. So we provide a python plugin for the GIMP image processing tool that allows the two error images to be loaded as separate layers, scaled and positioned to exactly the same size and position so that, by varying the transparency of the layers with the GIMP's layer tool, one can precisely see the issues involved. An example of the error results, when touching matches are found, are shown in Figure A.11.

	1	<math>
	2	...
	3	<mfrac id="a" xref="A">
	4	<mi id="b" xref="C"/>
	5	<mi id="c" xref="D"/>
	6	</mfrac>
	7	...
	8	<apply id="A" xref="a">
	9	<divide id="B"/>
	10	<csymbol cd="latexml" id="C" xref="b"
		>
	11	absent</csymbol>
	12	<csymbol cd="latexml" id="D" xref="c"
		>
	13	absent</csymbol>
	14	</apply>
	15	...
	16	\frac{}{}
	17	</math>

$$R_{ix}(t) = M_i A_{ix}(t)$$

$$R_{ix}(t) = M_i A_{ix}(t) \cdot$$

```

1  <math>
2  ...
3  <mfrac id="a" xref="A">
4  <mi id="b" xref="C"/>
5  <mi id="c" xref="D"/>
6  </mfrac>
7  ...
8  <apply id="A" xref="a">
9  <divide id="B"/>
10 <csymbol cd="latexml" id="C" xref="b"
    >
11 absent</csymbol>
12 <csymbol cd="latexml" id="D" xref="c"
    >
13 absent</csymbol>
14 </apply>
15 ...
16 \frac{}{}
17 </math>

```

Figure A.12: First rendering problem discovered by Mathpipe. The LaTeX input $R_{ix}(t) = M_i A_{ix}(t) \frac{}{} \cdot$ which ended with an empty fraction is rendered by LaTeX as $R_{ix}(t) = M_i A_{ix}(t)$. However, LaTeXML translates the empty fraction to MathML which causes that a faction without nominator and denominator is generated. Note, that we have manually modified the id element in the listing to improve the readability.

A.3.6 Further Work

As a work in progress, there is still much work to do in refining and improving the image comparison tool, testing and evaluating it on the various Mathpipe rendering pipelines and, eventually, building it into the Mathpipe construction tool chain

A.3.7 Conclusion

We have presented Wikipedia's new approach to higher performance, higher quality, scalable, accessible mathematical formula rendering and delivery and the work we carried out in performance analysis of its results that demonstrates its huge performance improvement over the previous approach. We have also presented our work on addressing a critical need for speedier and more robust development of further improvement in mathematical formula rendering; namely an image comparison program suited for use in automatic regression testing of mathematical formula rendering software. While this program is still under heavy development, it is already showing promise in providing support for more aggressive development of new Mathoid-based rendering methods.

Appendix B

Data Tables

The following tables lists the topics from NTCIR MIR tasks and the performance indicators of the approaches we analyzed in the respective tasks.

Table B.1: Statistics for NTCIR 10-FS

(1) $\int_0^\infty dx \int_x^\infty F(x, y) dy = \int_0^\infty dy \int_0^y F(x, y) dx$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	1	1	2
<5000	1	30	68	99
Σ	1	31	69	101
$\langle P \rangle$ in %	0.0	3.1		

(3) $x^{\textcolor{red}{n}} + y^{\textcolor{red}{n}} = z^{\textcolor{red}{n}}$

	++	+	o	Σ
>2k	7	2	0	9
5000-2k	2	8	24	34
<5000	4	12	42	58
Σ	13	22	66	101
$\langle P \rangle$ in %	59.6	41.2		

(5) $\frac{f(\textcolor{red}{x}+\textcolor{red}{h})-f(\textcolor{red}{x})}{\textcolor{red}{h}}$

	++	+	o	Σ
>2k	13	4	0	17
5000-2k	2	3	24	29
<5000	23	18	14	55
Σ	38	25	38	101
$\langle P \rangle$ in %	28.0	30.7		

(7) $\sin(\textcolor{red}{x})/\textcolor{red}{x}$

	++	+	o	Σ
>2k	5	11	0	16
5000-2k	0	6	9	15
<5000	5	10	59	74
Σ	10	27	68	105
$\langle P \rangle$ in %	23.5	55.0		

(2) $X(\textcolor{red}{i}\omega)$

	++	+	o	Σ
>2k	0	0	16	16
5000-2k	0	0	33	33
<5000	0	2	53	55
Σ	0	2	102	104
$\langle P \rangle$ in %	-	0.0		

(4) $\int_{-\infty}^\infty e^{-\textcolor{red}{x}^2} d\textcolor{red}{x}$

	++	+	o	Σ
>2k	3	0	0	3
5000-2k	6	21	21	48
<5000	5	15	31	51
Σ	14	36	52	102
$\langle P \rangle$ in %	27.7	35.8		

(6) $\sqrt{2} = 1 + \frac{1}{3} + \textcolor{red}{x} - \textcolor{red}{y}$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	8	36	44
<5000	0	17	41	58
Σ	0	25	77	102
$\langle P \rangle$ in %	-	5.0		

(8) $\textcolor{red}{a}x^2 + \textcolor{red}{b}x + \textcolor{red}{c}$

	++	+	o	Σ
>2k	19	3	3	25
5000-2k	1	0	18	19
<5000	25	3	29	57
Σ	45	6	50	101
$\langle P \rangle$ in %	36.8	42.5		

Table B.2: Statistics for NTCIR 10-FS (cont.)

(9) $\frac{e^x+y}{z}$

	++	+	o	Σ
>2k	0	28	13	41
5000-2k	0	2	5	7
<5000	0	10	45	55
Σ	0	40	63	103
$\langle P \rangle$ in %	-	46.3		

(11) $\int_{g \neq 0} |\nabla f|^q dx \leq c \int g \neq 0 |\nabla(f+g)|^q dx$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	9	4	13
<5000	0	33	54	87
Σ	0	42	58	100
$\langle P \rangle$ in %	-	16.2		

(13) $N_k(r, \frac{1}{f-a})$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	0	46	46
<5000	2	0	52	54
Σ	2	0	98	100
$\langle P \rangle$ in %	0.0	0.0		

(15) $\wp(z; \Lambda)$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	2	0	24	26
<5000	1	0	74	75
Σ	3	0	98	101
$\langle P \rangle$ in %	55.6	55.6		

(10) $f^n(z)f^{(k)}(az) \neq c$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	1	44	45
<5000	0	12	43	55
Σ	0	13	87	100
$\langle P \rangle$ in %	-	0.3		

(12) $q_n|a_n - a| \sim_{n \rightarrow +\infty} q_n|\frac{p_n}{q_n} - a|$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	8	0	8
<5000	0	18	74	92
Σ	0	26	74	100
$\langle P \rangle$ in %	-	30.8		

(14) $\ddot{u}(x, t) = u''(x, t)$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	1	29	0	30
<5000	0	5	65	70
Σ	1	34	65	100
$\langle P \rangle$ in %	4.3	85.7		

(16) $\wp(z; \omega_1, \omega_2)$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	11	1	15	27
<5000	8	1	66	75
Σ	19	2	81	102
$\langle P \rangle$ in %	48.1	45.8		

Table B.3: Statistics for NTCIR 10-FS (cont. 2)

(18) $O(\textcolor{red}{n} \log \textcolor{red}{n})$

	++	+	o	Σ
>2k	25	1	2	28
5000-2k	0	13	7	20
<5000	19	18	19	56
Σ	44	32	28	104
$\langle P \rangle$ in %	54.6	49.2		

(20) $|G : H| = \frac{|G|}{|H|}$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	4	5	9	18
<5000	28	22	32	82
Σ	32	27	41	100
$\langle P \rangle$ in %	4.8	11.4		

(22) $\textcolor{red}{A}_n = \frac{1}{\pi} \int_{-\pi}^{\pi} \textcolor{red}{F}(x) \cos(nx) dx$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	13	0	13
<5000	0	59	29	88
Σ	0	72	29	101
$\langle P \rangle$ in %	-	18.1		

(19) $Rf(L) = \int_L f(\mathbf{x}) |d\mathbf{x}|$

	++	+	o	Σ
>2k	0	4	14	18
5000-2k	0	0	0	0
<5000	0	20	62	82
Σ	0	24	76	100
$\langle P \rangle$ in %	-	11.0		

(21) $H^n(X) = Z^n(X)/B^n(X)$

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	6	0	0	6
<5000	21	12	61	94
Σ	27	12	61	100
$\langle P \rangle$ in %	22.2	15.4		

$$\langle \langle P_{++} \rangle \rangle = 31.5\%, \text{MAP}_+ = 28.5\%$$

Table B.4: Statistics for NTCIR 10-FT

(1) \wp Points derivative vanishes

	++	+	o	Σ
>2k	28	2	26	56
5000-2k	4	0	0	4
<5000	18	2	20	40
Σ	50	4	46	100
$\langle P \rangle$ in %	39.9	39.4		

(4) $\prod_{N=1}^{\infty} (1 + Z/N)$ diverges diverge

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	13	12	25
<5000	0	27	48	75
Σ	0	40	60	100
$\langle P \rangle$ in %	-	22.7		

(6) $\sum_{n=1}^{\infty} \frac{\sin(n)}{n}$ infinite series conditionally convergent

	++	+	o	Σ
>2k	0	2	2	4
5000-2k	0	16	26	42
<5000	0	21	33	54
Σ	0	39	61	100
$\langle P \rangle$ in %	-	14.8		

(8) $y^2 = x^3 + ax + b$ mod modulo

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	5	10	4	19
<5000	17	21	43	81
Σ	22	31	47	100
$\langle P \rangle$ in %	6.0	24.4		

(2) $\int_b^a f^2(x)dx$ NOT(Parseval)

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	1	12	45	58
<5000	1	14	27	42
Σ	2	26	72	100
$\langle P \rangle$ in %	0.9	16.5		

(5) $\sum \frac{n!x^n}{n^n}$ radius of convergence

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	18	30	48
<5000	0	38	14	52
Σ	0	56	44	100
$\langle P \rangle$ in %	-	18.8		

(7) $8x^3 + 4x^2 - 4x - 1$ root

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	12	36	48
<5000	5	18	29	52
Σ	5	30	65	100
$\langle P \rangle$ in %	0.0	6.8		

(9) p -adic diophantine equation

	++	+	o	Σ
>2k	5	7	31	43
5000-2k	0	0	0	0
<5000	16	30	11	57
Σ	21	37	42	100
$\langle P \rangle$ in %	2.2	11.0		

Table B.5: Statistics for NTCIR 10-FT (cont.)

(10) $r_k(C_4)$ estimated multicolor Ramsey number

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	2	0	32	34
<5000	8	1	57	66
Σ	10	1	89	100
$\langle P \rangle$ in %	20.0	18.2		

(12) $\frac{\partial u}{\partial t} - \Delta u + \frac{\langle D^2 u Du, Du \rangle}{1+|Du|^2} = 0$
uniqueness of solutions

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	3	7	10
<5000	0	17	73	90
Σ	0	20	80	100
$\langle P \rangle$ in %	-	6.7		

(14) $\sum p_n a_n$ convergence

	++	+	o	Σ
>2k	1	1	2	4
5000-2k	5	4	63	72
<5000	1	5	18	24
Σ	7	10	83	100
$\langle P \rangle$ in %	19.6	19.7		

(11) $x'(t) + \sum_{j=1}^N B_j(t)x(t - \tau_j(t)) = F(t)$
conditions boundedness

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	0	2	2
<5000	0	7	91	98
Σ	0	7	93	100
$\langle P \rangle$ in %	-	0.0		

(13) $x_{k+1} = \frac{A_1}{x_k^{p_1}} + \frac{A_2}{x_{k-1}^{p_2}} + \dots + \frac{A_n}{x_{k-n+1}^{p_n}}$
stability

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	0	2	20	22
<5000	0	9	69	78
Σ	0	11	89	100
$\langle P \rangle$ in %	-	1.6		

(15) $dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t$
solution

	++	+	o	Σ
>2k	0	0	0	0
5000-2k	12	4	12	28
<5000	21	22	29	72
Σ	33	26	41	100
$\langle P \rangle$ in %	16.2	16.6		

$$\langle \langle P_{++} \rangle \rangle = 16.7\%, \text{ MAP}_+ = 15.5\%$$

Table B.6: Query data. This table first lists query IDs followed by the queries, where the **qvar** elements (universal variables cf. <https://trac.mathweb.org/MWS/wiki/MwsQuery>) are listed in red. The columns $v = 0 - 4$ represent the relevance ranking (from the non-relevant to the most relevant). Columns F_1 through F_5 correspond to similarity measure factors 1 – 5. The number of Content Dictionary matches is F_1 , the number of data type matches is F_2 , the number of exact matches at any depth is F_3 , the average coverage is F_4 , and the number of formulae (as opposed to expressions) is F_5 .

ID	query	$v = 0$	$v = 1$	$v = 2$	$v = 3$	$v = 4$	F_1	F_2	F_3	F_4	F_5
1f1.0	$\text{square}(\text{phi}) = id$	447	207	784	209	11	12	13	1	0.19	441
1f1.1	$\text{phi} \neq id$	447	207	784	209	11	44	8	8	0.19	441
2f1.0	$ImP_\gamma^+ = C_\mu^+(\gamma)$	399	219	19	0	14	0	0	0	0.07	106
3f1.0	$L'd_{-k} = L_k$	764	80	99	10	0	0	0	0	0.12	239
4f1.0	$B\sigma_3 B = \sigma_3$	1307	19	72	11	7	5	5	0	0.11	383
5f1.0	$S_{EH} = \frac{1}{G_3} od^3 x \sqrt{-g^{(3)}}$	75	187	313	48	73	0	0	0	0.14	198
6f1.0	$S = -T_p \int d^{p+1} x \sqrt{g}$	936	249	118	90	205	0	0	0	0.36	445
7f1.0	$x \frac{y}{z} - u \frac{v}{w}$	847	232	12	42	0	0	0	0	1.00	246
8f1.0	$x \leq \frac{6}{2^n} + 12\epsilon$	864	119	14	24	0	0	0	0	0.15	225
9f1.0	$x_n^i = (1 - \epsilon)f + \frac{\epsilon}{2}[g + h]$	1726	79	224	0	0	0	0	0	0.18	527
10f1.0	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} z$	458	160	68	8	60	0	0	0	0.17	231
11f1.0	$p^2 + x^2(ix)^\epsilon$	649	466	9	25	84	11	11	0	0.22	369
12f1.0	L_∞	515	135	66	47	36	196	205	122	0.30	117
13f1.0	(D)	645	7	17	60	6	723	727	281	0.38	103
14f1.0	$-tr(x \ln x)$	427	147	42	7	17	0	0	0	0.23	164
15f1.0	$\frac{1}{n^s}$	666	10	115	0	43	87	78	52	0.32	157
16f1.0	$f(x) = x$	909	66	4	72	9	22	76	0	1.00	322
17f1.0	$f(z) = z^d + c$	579	50	14	83	70	8	12	0	0.35	211
18f1.0	$\frac{az+b}{cz+d}$	531	30	0	44	290	34	34	33	0.27	285
19f1.0	$H_{n-k}(X)$	245	60	10	213	431	13	23	0	0.25	175
20f1.0	$x^2 - x - 1 = 0$	1562	11	47	44	123	6	6	5	0.29	502
21f1.0	$f(ax + by) < af(x) + bf(y)$	418	265	59	35	24	0	0	0	0.32	195
22f1.0	$\int_M f dS$	183	109	97	52	328	29	31	0	0.23	182
23f1.0	$\langle \cdot, \cdot \rangle$	114	108	174	239	112	141	141	0	0.04	92
24f1.0	$\widehat{CH}^p(A) \cong Y$	324	48	11	80	236	1	1	0	0.16	72
25f1.0	$\widehat{\deg}(x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n})$	1016	15	14	10	6	0	0	0	0.30	256
26f1.0	$\det(a_1 b_2 - a_2 b_1 + c)$	738	193	12	20	0	0	0	0	0.28	249
27f1.0	$E(\lambda) = -m_{\text{dyn}}^2(\lambda)$	484	496	19	12	99	3	3	0	0.10	323
28f1.0	Φ^4	426	21	8	181	130	207	347	0	0.09	146
29f1.0	$\sum_{n=0}^\infty t^m a_k(x)$	373	665	13	222	61	0	0	0	0.28	349
30f1.0	$\mathbb{C}P^n$	240	15	0	16	319	123	128	0	0.13	80
31f1.0	$k + 1/(3k + c)$	877	44	96	38	0	0	0	0	0.43	397

Table B.6: Query data. This table first lists query IDs followed by the queries, where the **qvar** elements (universal variables cf. <https://trac.mathweb.org/MWS/wiki/MwsQuery>) are listed in red. The columns $v = 0 - 4$ represent the relevance ranking (from the non-relevant to the most relevant). Columns F_1 through F_5 correspond to similarity measure factors 1–5. The number of Content Dictionary matches is F_1 , the number of data type matches is F_2 , the number of exact matches at any depth is F_3 , the average coverage is F_4 , and the number of formulae (as opposed to expressions) is F_5 .

ID	query	$v = 0$	$v = 1$	$v = 2$	$v = 3$	$v = 4$	F_1	F_2	F_3	F_4	F_5
32f1.0	$ u \cdot v \leq u v $	729	74	27	14	0	2	2	0	0.00	227
33f1.0	$ fg _1 \leq f _p g _q$	878	34	25	11	42	2	4	0	0.22	259
34f1.0	$\lim_{n \rightarrow \infty} \int_{\mathbf{X}} f_n du = \int_{\mathbf{X}} \lim_{n \rightarrow \infty} f_n du$	833	57	69	51	0	0	0	0	0.24	242
35f1.0	$ x - a \leq \frac{1}{ a^{-1} }$	1089	82	27	9	6	0	0	0	0.38	328
36f1.0	$\rho(A) = \lim_{n \rightarrow \infty} A^n ^{1/n}$	384	210	22	38	90	10	10	0	0.22	217
37f1.0	$A = USV^T$	521	249	85	60	151	10	10	0	0.30	293
38f1.0	$ x + y _p \leq x _p + y _p$	260	337	70	169	33	7	6	1	1.00	264
39f1.0	$\mathbb{P}[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$	196	303	52	14	37	1	1	0	0.05	132
40f1.0	$\lim_{n \rightarrow \infty} \mathbb{P}[A_n - \mathbb{E}[X]] > e] = 0$	443	381	39	6	3	0	0	0	0.12	194
41f1.0	$\mathbb{P}[\lim_{n \rightarrow \infty} A_n = \mathbb{E}[X]] = 1$	103	255	158	3	17	0	0	0	0.19	123
42f1.0	$E = \bigoplus_{i=0}^{\infty} E_i$	197	152	340	42	22	14	14	0	0.22	156
43f1.0	$\oint_{\mathcal{C}} \mathbf{B} \cdot d\ell = \mu_0 I$	1000	1251	78	16	31	0	0	0	0.03	606
44f1.0	$x^n + y^n = z^n$	1250	16	3	58	24	5	7	0	1.00	292
44f1.1	$x, y, z, n \in \mathbb{N}$	1250	16	3	58	24	0	0	0	0.00	292
45f1.0	$\frac{1+\sqrt{5}}{2}^n$	565	103	40	5	77	9	10	9	0.30	198
46f1.0 ¹	$1024k^{10} - 2560k^9 + 3840k^8 - 4480k^7 + 4096k^6 - 2944k^5 + 1696k^4 - 760k^3 + 236k^2 - 40k$	890	19	23	19	2	0	0	0	0.16	220
47f1.0	$P_n = 2P_{n-1} + P_{n-2}$	948	142	10	18	82	7	7	3	0.39	383
48f1.0	$\dot{x}(t) = Ax(t) + Bu(t)$	78	759	45	0	25	3	3	0	0.22	162
48f1.1	$t \in \mathbb{R}$	78	759	45	0	25	63	63	0	0.15	162
48f1.2	$x(t) \in \mathbb{R}^n$	78	759	45	0	25	4	4	0	0.18	162
48f1.3	$u(t) \in \mathbb{R}^m$	78	759	45	0	25	4	4	0	0.16	162
49f1.0 ¹	$\sum_{n=1}^{2*k-1} (-1)^n * \cos(1/4 * \pi) * n^{2/k} = R$	1074	155	9	0	0	0	0	0	0.25	348
50f1.0	$\chi'_a(G) \leq \Delta(G) + 6$	462	7	121	84	228	4	3	0	0.02	223

¹We added line breaks to the query 46f1.0 and 49f1.0 to improve readability.

Table B.7: Submitted results and aggregated relevance score (0-4) in parentheses. A question mark in parentheses (?) denotes that no relevance score was given.

#	Topic	Result and Comment
1	what symbol is ζ	Riemann zeta function(4) other results may exist
2	define \iff in $A \iff B$	Logical equivalence(4), If and only if(?)
3	definition $a \oplus b$	Direct sum(4), \oplus (?)
4	$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$	Determinant ⁺ (4)
5	define notation ${}_2F_1(a, b; c; z)$	Hypergeometric function ⁺ (4)
6	$a_0 + \frac{b_1}{a_1+} \frac{b_2}{a_2+} \dots$ define pattern	Continued fraction ⁺ (4)
7*	$PY(*1*)$	Pitman-Yor process(?), Classical general equilibrium model ⁺ (0)
8	$n = \frac{c}{*1*}$ light	Refractive index(4) $*1*$ denotes the frequency ν .
9	$*1*_{n+1} = r*1*_n(1 - *1*_n)$ recurrence relation	Bifurcation diagram ⁺ (4)
10	$g(x) = \frac{1}{1+e^{-x}}$	Logistic function(4), Multimodal learning(0)
11	$F = ma$	Newton's laws of motion ⁺ (2) Comment: The authors consider Newton's second law as a relevant result for the query $F = ma$., Force(4)
12	Legendre $\left(\frac{a}{p}\right)$	Legendre symbol(4)
13	find $ax^2 + bx + c = (*1*)^2 + *2*$	Quadratic equation(4), Quadratic function(2), Binomial theorem(?)
14*	convert $\log_2(*1*)$ to $\ln(*1*)$	Binary logarithm ⁺ (3)
15	compute value for $\binom{n}{k}$	Binomial coefficient(4)
16	solve $x_n^2 - x_{n-1}x_{n+1} = c$ for x_n	No result! At first glance, $x_n = \sqrt{c - x_{n-1}x_{n+1}}$ appears to be related to the Mandelbrot set(?), but the $n + 1$ index is hard to interpret.
17	factor $x^3 + Dy^3 + D^2z^3 - 3Dxyz$ multiple variables	Hessian form of an elliptic curve(2)
18	$\lim_{x \rightarrow 0} \frac{2 - \cos(3x) - \cos(4x)}{x}$ solve limit	L'Hôpital's rule(3)
19	sequence name 1, 2, 2, 3, 3, 4, 4, 4, 5, 5, ...	Golomb sequence(4)
20	why $*1*^2 - 7*1* + 2$ polynomial but $\frac{*1*^2 - 7*1* + 2}{*1* + 2}$ not polynomial	Polynomial(4)

Table B.7: Submitted results and aggregated relevance score (0-4) in parentheses. A question mark in parentheses (?) denotes that no relevance score was given.

#	Topic	Result and Comment
21	difference between $\text{Log } *1*$ and $\log *1*$	Common logarithm(0) Comment: This article elaborates on the difference between <i>Log</i> and <i>log</i> in detail.
22*	explanation intuition $\nabla \times E = -\frac{\delta B}{\delta t}$	Faraday's law of induction(2) Comment: The article explains the Maxwell-Faraday equation $\nabla \times E = -\frac{\partial B}{\partial t}$ in detail
23*	is $P = NP$ possible	P versus NP problem(4)
24*	what is gamma $\int_0^{*1*} e^{-x} dx$	Gamma function(4)
25	prove $(f \circ g)' = (f' \circ g) \cdot g'$	Chain rule(4)
26	prove $x^2 + y^2 = z^2$	Pythagorean theorem(4)
27	$*1*^2 + *2*^2 = *3*^2$ uses	Pythagorean triple(3), Polar coordinate system(3), Cauchy-Schwarz inequality(3), Euclidean vector ⁺ (2), Parallelogram law(2)
28	example uses $f(a_1x_1 + \dots + a_nx_n) \leq a_1f(x_1) + \dots + a_nf(x_n)$ where $x_i \in \mathbb{R}$ $a_i \geq 0$ $\sum_{i=1}^n a_i = 1$	Sublinear function(1), Subadditivity ⁺ (1)
29*	application growth $\frac{dP}{dt} = r(1 - \frac{P}{K})$ P epidemiology biology	Logistic function ⁺ (4), R/K selection theory(4)
30	applications $f \star g$ where $(f \star g)[*1*] := \sum_{*2*=-\infty}^{\infty} f(*2*)g(*1* - *2*)$	Convolution ⁺ (2) Comment: The formula in the query is neither an exact match for convolution $(f * g)[*1*] := \sum_{*2*=-\infty}^{\infty} f[*2*]g[*1* + *2*]$ nor for cross correlation(4) $(f \star g)[*1*] := \sum_{*2*=-\infty}^{\infty} f^*[*2*]g[*1* - *2*]$. Note that for f non Hermitian $f \star g = f^*(-t) * g$.

*) Note that the incorrect typesetting was given in the topics. ⁺) The link points to a specific section of the Wikipedia article.

Table B.8: Compiled gold standard: Our original results are stated in boldface in case we still consider them as relevant, or stroked through in case we reconsidered our decision given the assessors' feedback. Normal font indicates results of other participants that we included in the gold standard.

#	Topic	Result and Comment
1	what symbol is ζ	Riemann zeta function , Damping ratio, Hurwitz zeta function, 1s Slater-type function, Jerk (physics), Oblate spheroidal coordinates, Routhian mechanics Comment: 6 new hits.
2	define \iff in $A \iff B$	Monoidal t-norm logic, Logical equivalence , If and only if , Logical biconditional Contraposition, (Tautology (logic) or Exportation (logic)) Comment: 4 new hits; the assessors rated 1 hit as irrelevant that we consider relevant.
3	definition $a \oplus b$	Direct sum , \oplus , Exclusive or Comment: 1 new hit.
4	$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$	Determinant , Laplace expansion Comment: 1 new hit.
5	define notation ${}_2F_1(a, b; c; z)$	Hypergeometric function Comment: no new hits.
6	$a_0 + \frac{b_1}{a_1+} \frac{b_2}{a_2+} \dots$ define pattern	Generalized continued fraction, Continued fraction Comment: 1 new hit.
8	$n = \frac{c}{*1*}$ light	Refractive index Comment: no new hits.
9	$*1*_{n+1} = r*1*_n(1 - *1*_n)$ recurrence relation	Logistic map Bifurcation diagram , Lyapunov fractal Comment: 2 new hits (one more and one less relevant than our original hit).
10	$g(x) = \frac{1}{1+e^{-x}}$	Logistic function , Multimodal learning Comment: No new hits; we consider the second hit as a duplicate.
11	$F = ma$	Newton's laws of motion , Force Comment: no new hits.
12	Legendre $\left(\frac{a}{p}\right)$	Legendre symbol Comment: no new hits.
13	find $ax^2 + bx + c = (*1*)^2 + *2*$	Quadratic equation , Quadratic function , Binomial theorem Comment: no new hits.
14	convert $\log_2(*1*)$ to $\ln(*1*)$	Binary logarithm Comment: no new hits.
15	compute value for $\binom{n}{k}$	Binomial coefficient Comment: no new hits.
18	$\lim_{x \rightarrow 0} \frac{2 - \cos(3x) - \cos(4x)}{x}$ solve limit	L'Hôpital's rule , List of limits Comment: 1 new hit.
19	sequence name 1, 2, 2, 3, 3, 4, 4, 4, 5, 5, ...	Golomb sequence Comment: no new hits.

Table B.8: Compiled gold standard: Our original results are stated in boldface in case we still consider them as relevant, or stroked through in case we reconsidered our decision given the assessors' feedback. Normal font indicates results of other participants that we included in the gold standard.

#	Topic	Result and Comment
20	why $*1*^2 - 7*1* + 2$ polynomial but $\frac{*1*^2 - 7*1* + 2}{*1* + 2}$ not polynomial	Polynomial Comment: no new hits.
21	difference between Log $*1*$ and $\log *1*$	Common logarithm Comment: no new hits.
22	explanation intuition $\nabla \times E = -\frac{\delta B}{\delta t}$	Faraday's law of induction Comment: no new hits.
23	is $P = NP$ possible	P versus NP problem Comment: no new hits.
24	what is gamma $\int_0^{*1*} e^{-x} dx$	Gamma function , Incomplete gamma function Comment: 1 new hit.
25	prove $(f \circ g)' = (f' \circ g) \cdot g'$	Chain rule Comment: no new hits.
26	prove $x^2 + y^2 = z^2$	Pythagorean theorem Comment: no new hits.
27	$*1*^2 + *2*^2 = *3*^2$ uses	Pythagorean triple , Polar coordinate system , Cauchy-Schwarz inequality , Euclidean vector , Parallelogram law , Crossed ladders problem, Cauchy-Schwarz inequality, Law of cosines, Isosceles triangle, Slant height, Special right triangles, Triangle Comment: 7 new hits (no particular order of relevance).
28	example $f(a_1x_1 + \dots + a_nx_n) \leq a_1f(x_1) + \dots + a_nf(x_n)$ where $x_i \in \mathbb{R}$ $a_i \geq 0$ $\sum_{i=1}^n a_i = 1$ uses	Jensen's inequality , Sublinear function , Subadditivity Comment: 1 new hit that we consider significantly more relevant than our hits.
29	application $\frac{dP}{dt} = r(1 - \frac{P}{K})P$ growth epi-demiology biology	Logistic function , R/K selection theory , Ecology Comment: 1 new hit.
30	applications $f \star g$ where $(f \star g)[*1*] := \sum_{*2*=-\infty}^{\infty} f(*2*)g(*1* - *2*)$	Convolution , Cross-correlation Comment: 1 new hit; both hits are equally relevant and very similar.

List of Figures

1.1	A stem publication with formulae	2
2.1	Overview of the Content Augmentation process	9
2.2	Abstraction level of formulae	11
2.3	Example of a scanned book.	13
2.4	Highlighting mathematical expressions in browsers	19
2.5	Visualizations of different possible syntactical structures	20
3.1	DRMF seeding data flow	34
3.2	Semantic macro breakdown	36
3.3	MLP pipeline overview	42
3.4	Data flow of the Stratosphere program	45
3.5	Illustration of the identifier-document matrix	54
3.6	Distribution of identifier counts.	59
3.7	Selected entries from the gold standard.	61
3.8	MediaWiki extension Math 2.0 overview	74
3.9	Comparison of different math renderings.	76
4.1	Frequency distribution of formulae in Wikipedia	82
4.2	WMC topic characteristics	84
4.3	Page-centric evaluation	86

4.4	Overview of our experimental setup	89
4.5	Overview of the assessments	90
4.6	Comparison of our results to the average of the other systems	92
4.7	Example content dictionary entries for a search pattern	98
4.8	Match Depth	100
4.9	Query coverage similarity measure	101
5.1	An English sentence from a proof	107
A.1	The MediaWiki math rendering process (2012)	119
A.2	Schema of MediaWiki extension MathSearch	121
A.3	MathSearch at the MIR happening in 2012	122
A.4	Formula pre-filtering based on $\text{T}_{\text{E}}\text{X}$ tokens	127
A.5	Mathosphere data flow	128
A.6	Topic analysis page	135
A.7	Copying MathML expressions from Wikibooks to Wolfram Mathematica . . .	140
A.8	Overview of the Wikimedia infrastructure	141
A.9	Rendering and verification time	143
A.10	Mathpipe processing chain	144
A.11	Error images from two different renderings	150
A.12	First rendering problem discovered by Mathpipe	151

List of Tables

3.1	DRMF overview	35
3.2	Semantic LaTeX macros	36
3.3	Implemented static patterns	44
3.4	Most common definition relations	47
3.5	Evaluation results.	48
3.6	Identifier definitions	63
3.6	Identifier definitions	64
3.7	Comparison of different math rendering engines	78
4.1	Overview of participant results	83
4.2	Assessment matrix for our results	91
4.3	Sister duplicates.	94
4.4	Parent-child duplicates.	104
4.5	Match depth similarity measure	105
4.6	Query coverage similarity measure	105
A.1	Overview of the top 10 NTCIR 10 results	137
B.1	Statistics for NTCIR 10-FS	154
B.2	Statistics for NTCIR 10-FS (cont.)	155
B.3	Statistics for NTCIR 10-FS (cont. 2)	156

B.4	Statistics for NTCIR 10-FT	157
B.5	Statistics for NTCIR 10-FT (cont.)	158
B.6	Query data.	159
B.6	Query data.	160
B.7	Submitted results	161
B.7	Submitted results	162
B.8	Compiled gold standard	163
B.8	Compiled gold standard	164

Bibliography

- [1] A. Grbin and C. Maloney. **svgtex**. <https://github.com/agrbin/svgtex>. Mar. 20, 2014.
- [2] M. Abramowitz and I. S. (eds.) **Handbook of Mathematical Functions**. Vol. 55. National Bureau of Standards Applied Mathematics Series U.S. Government Printing Office, Washington, DC, 1964.
- [3] M. Adeel, H. S. Cheung, and S. H. Khiyal. “Math GO! prototype of a content based mathematical formula search engine”. In: **Journal of Theoretical and Applied Information Technology** 4.10 (2008), pp. 1002–1012.
- [4] C. C. Aggarwal and C. Zhai. “A survey of text clustering algorithms”. In: **Mining Text Data**. Springer, 2012, pp. 77–128.
- [5] S. A. Ahmadi and A. Youssef. “Lexical Error Compensation in Handwritten-Based Mathematical Information Retrieval”. In: **Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008**. Brno: Masaryk University, 2008, pp. 43–54. URL: <http://eudml.org/doc/220930>.
- [6] A. Aizawa, M. Kohlhase, and I. Ounis. “NTCIR-10 Math Pilot Task Overview”. In: **Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies** (2013), pp. 654–661.
- [7] A. Aizawa et al. “NTCIR-11 Math-2 Task Overview”. In: **Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies**. National Institute of Informatics (NII), 2014, pp. 88–98.
- [8] A. Aizawa et al. “NTCIR-12 Math-3 Task Overview”. In: **NTCIR**. National Institute of Informatics (NII), 2016.
- [9] M. A. Alkalai et al. “Improving Formula Analysis with Line and Mathematics Identification”. In: **2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013**. IEEE Computer Society, 2013, pp. 334–338. DOI: 10.1109/ICDAR.2013.74.

- [10] M. E. Altamimi and A. Youssef. "A more canonical form of content MathML to facilitate math search". In: **Proc. Extreme Markup Languages**. Citeseer. 2007.
- [11] M. E. Altamimi and A. S. Youssef. "A Math Query Language with an Expanded Set of Wildcards". English. In: **Mathematics in Computer Science** 2.2 (2008), pp. 305–331. ISSN: 1661-8270. DOI: 10.1007/s11786-008-0056-4.
- [12] M. E. Altamimi and A. S. Youssef. "Wildcards in Math Search, Implementation Issues". In: **Proceedings of the ISCA 20th International Conference on Computer Applications in Industry and Engineering, CAINE 2007, November 7-9, 2007, San Francisco, California, USA**. Ed. by G. Hu. ISCA, 2007, pp. 90–96.
- [13] G. M. Amdahl. "Validity of the single processor approach to achieving large scale computing capabilities". In: **American Federation of Information Processing Societies: Proceedings of the AFIPS '67 Spring Joint Computer Conference, April 18-20, 1967, Atlantic City, New Jersey, USA**. Vol. 30. AFIPS Conference Proceedings. AFIPS, 1967, pp. 483–485. DOI: 10.1145/1465482.1465560.
- [14] American Mathematical Society. **AMS Mathematics Subject Classification 2010**. 2009.
- [15] American Physical Society. **PACS 2010 Regular Edition**. 2009.
- [16] G. E. Andrews, R. Askey, and R. Roy. **Special functions**. Vol. 71. Encyclopedia of Mathematics and its Applications. Cambridge: Cambridge University Press, 1999, pp. xvi+664.
- [17] S. Arunachalam. "Open access-current developments in India". In: **Proceedings Berlin 4 Open Access: From Promise to Practice, Potsdam-Golm, Germany**. March. 2006, pp. 1–9. URL: <http://1.formulasearchengine.com/arunachalam>.
- [18] A. Asperti and M. Selmi. "Efficient Retrieval of Mathematical Statements". In: **Mathematical Knowledge Management, Third International Conference, MKM 2004, Bialowieza, Poland, September 19-21, 2004, Proceedings**. Ed. by A. Asperti, G. Bancerek, and A. Trybulec. Vol. 3119. Lecture Notes in Computer Science. Springer, 2004, pp. 17–31. DOI: 10.1007/978-3-540-27818-4_2.
- [19] A. Asperti and S. Zacchiroli. "Searching Mathematics on the Web: State of the Art and Future Developments". In: **In: Proceedings of New Developments in Electronic Publishing of Mathematics, p. 9-18, Edited by FIZ**. Citeseer. 2004.

- [20] A. Asperti et al. “A Content Based Mathematical Search Engine: Whelp”. In: **Types for Proofs and Programs: International Workshop, TYPES 2004, Jouy-en-Josas, France, December 15-18, 2004, Revised Selected Papers**. Ed. by J.-C. Filliâtre, C. Paulin-Mohring, and B. Werner. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 17–32. ISBN: 978-3-540-31429-5. DOI: 10.1007/11617990_2.
- [21] A.-M. Awal, H. Mouchère, and C. Viard-Gaudin. “A global learning approach for an online handwritten mathematical expression recognition system”. In: **Pattern Recognition Letters** 35 (Jan. 2014), pp. 68–77. DOI: 10.1016/j.patrec.2012.10.024.
- [22] J. B. Baker. “A linear grammar approach for the analysis of mathematical documents”. PhD thesis. University of Birmingham, UK, 2012. URL: <http://etheses.bham.ac.uk/3377/>.
- [23] J. B. Baker, A. P. Sexton, and V. Sorge. “A Linear Grammar Approach to Mathematical Formula Recognition from PDF”. In: **Intelligent Computer Mathematics, 16th Symposium, Calculemus 2009, 8th International Conference, MKM 2009, Held as Part of CICM 2009, Grand Bend, Canada, July 6-12, 2009. Proceedings**. Ed. by J. Carette et al. Vol. 5625. Lecture Notes in Computer Science. Springer, 2009, pp. 201–216. DOI: 10.1007/978-3-642-02614-0_19.
- [24] J. B. Baker, A. P. Sexton, and V. Sorge. “Faithful mathematical formula recognition from PDF documents”. In: **The Ninth IAPR International Workshop on Document Analysis Systems, DAS 2010, June 9-11, 2010, Boston, Massachusetts, USA**. Ed. by D. S. Doermann et al. ACM International Conference Proceeding Series. ACM, 2010, pp. 485–492. DOI: 10.1145/1815330.1815393.
- [25] J. B. Baker, A. P. Sexton, and V. Sorge. “MaxTract: Converting PDF to LaTeX, MathML and Text”. 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**. Ed. by J. Jeuring et al. Vol. 7362. Lecture Notes in Computer Science. Springer, 2012, pp. 422–426. ISBN: 978-3-642-31373-8. DOI: 10.1007/978-3-642-31374-5_29.
- [26] J. B. Baker et al. “Comparing Approaches to Mathematical Document Analysis from PDF”. In: **2011 International Conference on Document Analysis and Recognition, ICDAR 2011, Beijing, China, September 18-21, 2011**. IEEE Computer Society, 2011, pp. 463–467. DOI: 10.1109/ICDAR.2011.99.
- [27] G. Bancerek. “Information Retrieval and Rendering with MML Query”. In: **Mathematical Knowledge Management: 5th International Conference, MKM 2006, Wokingham, UK, August 11-12, 2006. Proceedings**. Ed. by J. M. Borwein and W. M. Farmer. Berlin, Heidelberg: Springer

- Berlin Heidelberg, 2006, pp. 266–279. ISBN: 978-3-540-37106-9. DOI: 10.1007/11812289_21.
- [28] G. Bancerek and P. Rudnicki. “Information Retrieval in MML”. In: **Mathematical Knowledge Management, Second International Conference, MKM 2003, Bertinoro, Italy, February 16-18, 2003, Proceedings**. Ed. by A. Asperti, B. Buchberger, and J. H. Davenport. Vol. 2594. Lecture Notes in Computer Science. Springer, 2003, pp. 119–132. DOI: 10.1007/3-540-36469-2_10.
- [29] G. Bancerek and J. Urban. “Integrated Semantic Browsing of the Mizar Mathematical Library for Authoring Mizar Articles”. In: **Mathematical Knowledge Management: Third International Conference, MKM 2004, Białowieża, Poland, September 19-21, 2004. Proceedings**. Ed. by A. Asperti, G. Bancerek, and A. Trybulec. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 44–57. ISBN: 978-3-540-27818-4. DOI: 10.1007/978-3-540-27818-4_4.
- [30] D. Battré et al. “Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical Processing”. In: **Proceedings of the 1st ACM symposium on Cloud computing**. SoCC ’10. Indianapolis, Indiana, USA: ACM, 2010, pp. 119–130. ISBN: 978-1-4503-0036-0. DOI: 10.1145/1807128.1807148.
- [31] P. Baumgartner and U. Furbach. “Automated Deduction Techniques for the Management of Personalized Documents”. In: **Ann. Math. Artif. Intell.** 38.1-3 (2003), pp. 211–228. DOI: 10.1023/A:1022976016809.
- [32] J. Beel et al. “Research Paper Recommender Systems: A Literature Survey”. In: **International Journal on Digital Libraries** (2015), pp. 1–34. ISSN: 1432-5012. DOI: 10.1007/s00799-015-0156-0.
- [33] D. Blostein and R. Zanibbi. “Processing Mathematical Notation”. In: **Handbook of Document Image Processing and Recognition**. Ed. by D. Doermann and K. Tombre. London: Springer London, 2014, pp. 679–702. ISBN: 978-0-85729-859-1. DOI: 10.1007/978-0-85729-859-1_21.
- [34] E. Börjesson and R. Feldt. “Automated System Testing Using Visual GUI Testing Tools: A Comparative Study in Industry”. In: **5th IEEE Int. Conf. on Software Testing, Verification and Validation, ICST 2012**. Ed. by G. Antoniol, A. Bertolino, and Y. Labiche. IEEE Computer Society, 2012, pp. 350–359. DOI: 10.1109/ICST.2012.115.
- [35] L. Bornmann and R. Mutz. “Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references”. In: **Journal of the Association for Information Science and Technology** 66.11 (2015), pp. 2215–2222. DOI: 10.1002/asi.23329.
- [36] B. Bos. **The W3C MathML software list**. Mar. 20, 2014. URL: <http://www.w3.org/Math/Software/>.

- [37] T. Bouche. “Digital Mathematics Libraries: The Good, the Bad, the Ugly”. In: **Mathematics in Computer Science 3.3** (2010), pp. 227–241. ISSN: 1661-8289. DOI: 10.1007/s11786-010-0029-2.
- [38] Y. A. Brychkov. **Handbook of Special Functions: Derivatives, Integrals, Series and Other Formulas**. Boca Raton-London-New York: Chapman & Hall/CRC Press, 2008, pp. xx+680. ISBN: 978-1-58488-956-4.
- [39] P. F. Byrd and M. D. Friedman. **Handbook of elliptic integrals for engineers and physicists**. Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete. Bd LXVII. Berlin: Springer-Verlag, 1954, pp. xiii+355.
- [40] P. A. Cairns. “Informalising Formal Mathematics: Searching the Mizar Library with Latent Semantics”. In: **Mathematical Knowledge Management, Third International Conference, MKM 2004, Bialowieza, Poland, September 19-21, 2004, Proceedings**. Ed. by A. Asperti, G. Bancerek, and A. Trybulec. Vol. 3119. Lecture Notes in Computer Science. Springer, 2004, pp. 58–72. DOI: 10.1007/978-3-540-27818-4_5.
- [41] F. Cajori. **A history of mathematical notations. 1. Notations in elementary mathematics**. Chicago, Ill.: Open Court Publ. Co., 1928, —Ill., graph. Darst.
- [42] O. Caprotti, M. Dewar, and D. Turi. “Mathematical Service Matching Using Description Logic and OWL”. In: **Mathematical Knowledge Management, Third International Conference, MKM 2004, Bialowieza, Poland, September 19-21, 2004, Proceedings**. Ed. by A. Asperti, G. Bancerek, and A. Trybulec. Vol. 3119. Lecture Notes in Computer Science. Springer, 2004, pp. 73–87. DOI: 10.1007/978-3-540-27818-4_6.
- [43] J. Carette and W. Farmer. “A Review of Mathematical Knowledge Management”. In: **MKM/Calculemus Proceedings**. Ed. by J. Carette et al. LNAI 5625. Springer Verlag, 2009, pp. 233–246. ISBN: 9783642026133.
- [44] D. Carlisle. “OpenMath, MathML, and XSL”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 6–11. ISSN: 0163-5824. DOI: 10.1145/362001.362011.
- [45] D. Cervone. “MathJax: A Platform for Mathematics on the Web”. In: **Notices of the AMS** 59.2 (2012), pp. 312–316. URL: <http://www.ams.org/journals/notices/201202/rtx120200312p.pdf>.
- [46] S. Chaudhuri et al. “Probabilistic information retrieval approach for ranking of database query results”. In: **ACM Transactions on Database Systems** 31.3 (Sept. 2006), pp. 1134–1168. ISSN: 0362-5915. DOI: 10.1145/1166074.1166085.
- [47] W. Chisholm, G. Vanderheiden, and I. Jacobs. “Web Content Accessibility Guidelines 1.0”. In: **Interactions** 8.4 (July 2001), pp. 35–54. ISSN: 1072-5520. DOI: 10.1145/379537.379550.

- [48] H. S. Cohl et al. “Digital Repository of Mathematical Formulae”. In: **Intelligent Computer Mathematics, Lecture Notes in Artificial Intelligence 8543**. Ed. by S. M. Watt, J. H. Davenport, A. P. Sexton, P. Sojka, J. Urban. Vol. 8543. LNCS. Springer, 2014, pp. 419–422. DOI: 10.1007/978-3-319-08434-3_30.
- [49] H. S. Cohl et al. “Growing the Digital Repository of Mathematical Formulae with Generic LaTeX Sources”. In: **Intelligent Computer Mathematics, Lecture Notes in Artificial Intelligence 9150**. Ed. by M. Kerber et al. Vol. 9150. LNCS. Springer, 2015, pp. 280–287. DOI: 10.1007/978-3-319-20615-8_18.
- [50] M. Cooper, T. Lowe, and M. Taylor. “Access to Mathematics in Web Resources for People with a Visual Impairment”. In: **Computers Helping People with Special Needs**. Ed. by K. Miesenberger et al. Vol. 5105. Lecture Notes in Computer Science. Springer, 2008, pp. 926–933. ISBN: 978-3-540-70539-0. DOI: 10.1007/978-3-540-70540-6_139.
- [51] R. M. Corless et al. “‘According to Abramowitz and Stegun’ or Arccoth Needn’T Be Uncouth”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 58–65. ISSN: 0163-5824. DOI: 10.1145/362001.362023.
- [52] N. R. Council. **Developing a 21st Century Global Library for Mathematics Research**. Washington, DC: The National Academies Press, 2014. ISBN: 978-0-309-29848-3. DOI: 10.17226/18619.
- [53] D. Crystal. **The Cambridge Encyclopedia of Language**. Second. Cambridge: Cambridge University Press, 1997.
- [54] D. Czepita and E. Lodygowska. “[Role of the organ of vision in the course of developmental dyslexia].” In: **Klinika oczna** 108.1-3 (Jan. 2006), pp. 110–3. ISSN: 0023-2157. URL: <http://www.ncbi.nlm.nih.gov/pubmed/16883955>.
- [55] J. H. Davenport. “On Writing OpenMath Content Dictionaries”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 12–15. ISSN: 0163-5824. DOI: 10.1145/362001.362012.
- [56] J. H. Davenport. **Conferences on Intelligent Computer Mathematics 2012. Notes by J.H. Davenport**. Tech. rep. University of Bath, July 13, 2012.
- [57] J. Dean and S. Ghemawat. “MapReduce: simplified data processing on large clusters”. In: **Commun. ACM** 51.1 (Jan. 2008), pp. 107–113. ISSN: 0001-0782. DOI: 10.1145/1327452.1327492.
- [58] S. C. Deerwester et al. “Indexing by latent semantic analysis”. In: **JAsIs** 41.6 (1990), pp. 391–407.
- [59] D. Delahaye. “Information Retrieval in a Coq Proof Library Using Type Isomorphisms”. In: **Types for Proofs and Programs, International Workshop TYPES’99, Lökeberg, Sweden, June 12-16, 1999, Selected Papers**. Ed. by T. Coquand et al. Vol. 1956. Lecture Notes in Computer Science. Springer, 1999, pp. 131–147. DOI: 10.1007/3-540-44557-9_8.

- [60] M. Dewar. “OpenMath: An Overview”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 2–5. ISSN: 0163-5824. DOI: 10.1145/362001.362008.
- [61] **NIST Digital Library of Mathematical Functions**. Release 1.0.10 of 2015-08-07. Online companion to [178]. 2016. URL: <http://dlmf.nist.gov>.
- [62] D. Doermann and K. Tombre, eds. **Handbook of Document Image Processing and Recognition**. Springer, 2014.
- [63] E. Duval et al. “Metadata principles and practicalities”. In: **D-lib Magazine** 8.4 (2002), p. 16.
- [64] K. T. Ellen M.D. Friedman. **Introduction to Apache Flink**. O’Reilly Media, Inc, USA, Nov. 11, 2016. 110 pp. ISBN: 1491976586. URL: http://www.ebook.de/de/product/27802371/ellen_m_d_friedman_kostas_tzoumas_introduction_to_apache_flink.html.
- [65] A. Erdélyi et al. **Higher transcendental functions. Vols. 1-3**. Melbourne, Fla.: Robert E. Krieger Publishing Co. Inc., 1981, pp. xiii+302. ISBN: 048644614X.
- [66] A. Erdélyi et al. **Tables of integral transforms. Vols. 1-2**. McGraw-Hill Book Company, Inc., New York-Toronto-London, 1954, pp. xx+391.
- [67] D. Formánek et al. “Normalization of Digital Mathematics Library Content”. In: **Joint Proceedings of the 24th OpenMath Workshop, the 7th Workshop on Mathematical User Interfaces (MathUI), and the Work in Progress Section of the Conference on Intelligent Computer Mathematics**. (Bremen, Germany, July 9–13, 2012). Ed. by J. Davenport et al. CEUR Workshop Proceedings 921. Aachen, 2012, pp. 91–103. URL: <http://ceur-ws.org/Vol-921/wip-05.pdf>.
- [68] W. Foundation. **MediaWiki**. 2001. URL: <http://www.mediawiki.org/wiki/MediaWiki> (visited on 10/05/2012).
- [69] G. Frege. “Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens”. In: **Nebert, Louis** (1879).
- [70] F. Furukori et al. “An OCR System with OCRopus for Scientific Documents Containing Mathematical Formulas”. In: **2013 12th International Conference on Document Analysis and Recognition**. Aug. 2013, pp. 1175–1179. DOI: 10.1109/ICDAR.2013.238.
- [71] L. Gao et al. “ICST Math Retrieval System for NTCIR-11 Math-2 Task”. 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**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/02-NTCIR11-MATH-GaoL.pdf>.

- [72] G. Gasper and M. Rahman. **Basic hypergeometric series**. Second. Vol. 96. Encyclopedia of Mathematics and its Applications. With a foreword by Richard Askey. Cambridge: Cambridge University Press, 2004, pp. xxvi+428. ISBN: 0-521-83357-4.
- [73] T. Gauthier and C. Kaliszyk. “Matching Concepts across HOL Libraries”. In: **Intelligent Computer Mathematics - International Conference, CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings**. Ed. by S. M. Watt et al. Vol. 8543. Lecture Notes in Computer Science. Springer, 2014, pp. 267–281. DOI: 10.1007/978-3-319-08434-3_20.
- [74] D. Ginev et al. “The LaTeXML Daemon: Editable Math on the Collaborative Web”. In: (2011). Ed. by J. H. Davenport et al., pp. 292–294. DOI: 10.1007/978-3-642-22673-1_25.
- [75] L. Glasser et al. “The integrals in Gradshteyn and Ryzhik. Part 22: Bessel- K functions”. In: **Scientia. Series A. Mathematical Sciences. New Series** 22 (2012), pp. 129–151.
- [76] A. Gliozzo and C. Strapparava. **Semantic domains in computational linguistics**. Springer Science & Business Media, 2009.
- [77] B. Goetz. “The Lucene search engine: Powerful, flexible, and free”. In: **JavaWorld**. Available <http://www.javaworld.com/javaworld/jw-09-2000/jw-0915-lucene.html> (2000).
- [78] J. Gosling et al. **The Java Language Specification. Java SE 8 Edition**. Addison-Wesley Professional, 2015.
- [79] I. S. Gradshteyn and I. M. Ryzhik. **Table of integrals, series, and products**. Seventh. Elsevier/Academic Press, Amsterdam, 2007, pp. xlviii+1171.
- [80] P. Graf. **Term Indexing**. Vol. 1053. Lecture Notes in Computer Science. Springer, 1996. ISBN: 3-540-61040-5. DOI: 10.1007/3-540-61040-5.
- [81] M. Grigore, M. Wolska, and M. Kohlhase. “Towards context-based disambiguation of mathematical expressions”. In: **ASCM**. 2009, pp. 262–271.
- [82] A. Grigorev. “Identifier Namespaces in Mathematical Notation”. In: **arXiv preprint arXiv:1601.03354** (2016).
- [83] J. Grimm. **Producing MathML with Tralics**. In: P. Sojka. **Towards a Digital Mathematics Library**. Masaryk University, 2010, pp. 105–117.
- [84] F. Guidi and C. Sacerdoti Coen. “A Survey on Retrieval of Mathematical Knowledge”. English. In: **Intelligent Computer Mathematics**. Ed. by M. Kerber et al. Vol. 9150. Lecture Notes in Computer Science. Springer International Publishing, 2015, pp. 296–315. ISBN: 978-3-319-20614-1. DOI: 10.1007/978-3-319-20615-8_20.
- [85] F. Guidi and C. Sacerdoti Coen. “A Survey on Retrieval of Mathematical Knowledge”. In: **Mathematics in Computer Science** 10.4 (2016), pp. 409–427. ISSN: 1661-8289. DOI: 10.1007/s11786-016-0274-0.

- [86] F. Guidi and I. Schena. “A Query Language for a Metadata Framework about Mathematical Resources”. In: **Mathematical Knowledge Management, Second International Conference, MKM 2003, Bertinoro, Italy, February 16-18, 2003, Proceedings**. Ed. by A. Asperti, B. Buchberger, and J. H. Davenport. Vol. 2594. Lecture Notes in Computer Science. Springer, 2003, pp. 105–118. DOI: 10.1007/3-540-36469-2_9.
- [87] H. Hagino and H. Saito. “Partial-match Retrieval with Structure-reflected Indices at the NTCIR-10 Math Task”. In: **Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-10, National Center of Sciences, Tokyo, Japan, June 18-21, 2013**. Ed. by N. Kando and T. Kato. National Institute of Informatics (NII), 2013. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/07-NTCIR10-MATH-HaginoH.pdf>.
- [88] N. Halko, P. Martinsson, and A. Tropp. “Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions.” In: 2009. URL: https://amath.colorado.edu/faculty/martinss/Pubs/2009%5C_HMT%5C_random%5C_review.pdf.
- [89] R. Hambasan, M. Kohlhase, and C. Prodescu. “MathWebSearch at NTCIR-11”. 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**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/05-NTCIR11-MATH-HambasanR.pdf>.
- [90] Y. Haralambous and P. Quaresma. “Querying Geometric Figures Using a Controlled Language, Ontological Graphs and Dependency Lattices”. In: **Intelligent Computer Mathematics - International Conference, CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings**. Ed. by S. M. Watt et al. Vol. 8543. Lecture Notes in Computer Science. Springer, 2014, pp. 298–311. DOI: 10.1007/978-3-319-08434-3_22.
- [91] H. Hashimoto, Y. Hijikata, and S. Nishida. “Incorporating breadth first search for indexing MathML objects”. In: **Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12-15 October 2008**. IEEE, 2008, pp. 3519–3523. DOI: 10.1109/ICSMC.2008.4811843.
- [92] L. He, Y. Chao, and K. Suzuki. “A run-based two-scan labeling algorithm”. In: **IEEE Transactions on Image Processing** 17.5 (2008), pp. 749–756.
- [93] L. He et al. “Fast connected-component labeling”. In: **Pattern Recognition** 42.9 (2009), pp. 1977–1987.
- [94] A. Heise et al. “Meteor/Sopremo: An Extensible Query Language and Operator Model”. In: **International Workshop on End-to-end Management of Big Data (BigData 2012)** (2012). URL: <http://www.cse.buffalo.edu/faculty/tkosar/bigdata2012/program.php>.

- [95] X. Hu et al. “WikiMirs: a mathematical information retrieval system for wikipedia”. In: **13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13, Indianapolis, IN, USA, July 22 - 26, 2013**. Ed. by J. S. Downie et al. ACM, 2013, pp. 11–20. DOI: 10.1145/2467696.2467699.
- [96] I. Iglezakis, T.-E. Synodinou, and S. Kapidakis, eds. **E-Publishing and Digital Libraries**. IGI Global, 2011. DOI: 10.4018/978-1-60960-031-0.
- [97] P. Ion. “The Effort to Realize a Global Digital Mathematics Library”. In: **Mathematical Software – ICMS 2016: 5th International Conference, Berlin, Germany, July 11-14, 2016, Proceedings**. Ed. by G.-M. Greuel et al. Cham: Springer International Publishing, 2016, pp. 458–466. ISBN: 978-3-319-42432-3. DOI: 10.1007/978-3-319-42432-3_59.
- [98] M. E. H. Ismail. **Classical and Quantum Orthogonal Polynomials in One Variable**. Vol. 98. Encyclopedia of Mathematics and its Applications. With two chapters by Walter Van Assche, With a foreword by Richard A. Askey. Cambridge: Cambridge University Press, 2005, pp. xviii+706. ISBN: 978-0-521-78201-2.
- [99] A. Jackson. “The Digital Mathematics Library”. In: **Notices of the American Mathematical Society** 50.8 (2003), pp. 918–924.
- [100] D. Jurafsky and J. H. Martin. **Speech & language processing**. Pearson Education India, 2000.
- [101] S. Kamali and F. W. Tompa. “A new mathematics retrieval system”. In: **Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010**. Ed. by J. Huang et al. ACM, 2010, pp. 1413–1416. DOI: 10.1145/1871437.1871635.
- [102] S. Kamali and F. W. Tompa. “Improving mathematics retrieval”. In: **Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009** (2009), pp. 37–48.
- [103] S. Kamali and F. W. Tompa. “Structural Similarity Search for Mathematics Retrieval”. In: **Intelligent Computer Mathematics - MKM, Calculemus, DML, and Systems and Projects 2013, Held as Part of CICM 2013, Bath, UK, July 8-12, 2013. Proceedings**. Ed. by J. Carette et al. Vol. 7961. Lecture Notes in Computer Science. Springer, 2013, pp. 246–262. DOI: 10.1007/978-3-642-39320-4_16.
- [104] D. E. Knuth. “Computer science and its relation to mathematics”. In: **American Mathematical Monthly** (1974), pp. 323–343.
- [105] R. Koekoek, P. A. Lesky, and R. F. Swarttouw. **Hypergeometric Orthogonal Polynomials and Their q -Analogues**. Ed. by T. H. Koornwinder. Springer Monographs in Mathematics. With a foreword by Tom H. Koornwinder. Berlin [u.a.]: Springer-Verlag, 2010, pp. xx+578. XIX, 578. ISBN: 978-3-642-05013-8. DOI: 10.1007/978-3-642-05014-5.

- [106] A. Kohlhase. “Search Interfaces for Mathematicians”. In: **Intelligent Computer Mathematics - International Conference, CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings**. Ed. by S. M. Watt et al. Vol. 8543. Lecture Notes in Computer Science. Springer, 2014, pp. 153–168. DOI: 10.1007/978-3-319-08434-3_12.
- [107] A. Kohlhase. “Search interfaces for mathematicians”. In: **Intelligent Computer Mathematics**. Springer, 2014, pp. 153–168.
- [108] A. Kohlhase and M. Kohlhase. “Re examining the MKM Value Proposition: From Math Web Search to Math Web Re Search”. In: **Towards Mechanized Mathematical Assistants, 14th Symposium, Calculemus 2007, 6th International Conference, MKM 2007, Hagenberg, Austria, June 27-30, 2007, Proceedings**. Ed. by M. Kauers et al. Vol. 4573. Lecture Notes in Computer Science. Springer, 2007, pp. 313–326. DOI: 10.1007/978-3-540-73086-6_25.
- [109] A. Kohlhase and M. Kohlhase. “Towards a Flexible Notion of Document Context”. In: **Proceedings of the 29th ACM International Conference on Design of Communication**. SIGDOC ’11. Pisa, Italy: ACM, 2011, pp. 181–188. ISBN: 978-1-4503-0936-3. DOI: 10.1145/2038476.2038512.
- [110] M. Kohlhase. **Formats for Topics and Submissions for the Math2 Task at NTCIR-11**. Tech. rep. 2014. URL: <http://ntcir-math.nii.ac.jp/wp-content/blogs.dir/13/files/2014/05/NTCIR11-Math-topics.pdf>.
- [111] M. Kohlhase, ed. **Joint Proceedings of the MathUI, OpenMath and ThEdu Workshops and Work in Progress track at CICM**. (Bialistok, Poland, July 19, 2016). CEUR Workshop Proceedings ? Aachen, 2016. URL: <http://ceur-ws.org/>.
- [112] M. Kohlhase. **Math Information Retrieval Happening**. Tech. rep. July 8, 2012.
- [113] M. Kohlhase. “The Flexiformalist Manifesto”. In: **2012 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing**. IEEE. Sept. 2012, pp. 30–35. DOI: 10.1109/SYNASC.2012.78.
- [114] M. Kohlhase. “Using LaTeX as a Semantic Markup Format”. In: **Mathematics in Computer Science 2.2 (2008)**, pp. 279–304. ISSN: 1661-8289. DOI: 10.1007/s11786-008-0055-5.
- [115] M. Kohlhase and C. C. Prodescu. “Scaling an Open Formula Search Engine”. In: **Challenge (2012)**, pp. 1–15.
- [116] M. Kohlhase, C. Prodescu, and C. Liguda. “XLSearch: A Search Engine for Spreadsheets”. In: **Proceedings of the EuSpRIG 2013 Conference “Spreadsheet Risk Management”. July 4-5, London, United Kingdom**. Ed. by S. T. et. al. Five Star Printing Ltd, Claydon, 2013, pp. 47–58. ISBN: 978-1-9054045-1-3.

- [117] M. Kohlhase and C. Prodescu. “MathWebSearch at NTCIR-10”. In: **Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-10, National Center of Sciences, Tokyo, Japan, June 18-21, 2013**. Ed. by N. Kando and T. Kato. National Institute of Informatics (NII), 2013. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/04-NTCIR10-MATH-KohlhaseM.pdf>.
- [118] M. Kohlhase and I. Sucan. “A Search Engine for Mathematical Formulae”. In: **Proceedings of Artificial Intelligence and Symbolic Computation, AISC’2006**. Ed. by T. Ida, J. Calmet, and D. Wang. Vol. 4120. Lecture Notes in Computer Science 4120. Springer Berlin Heidelberg, 2006, pp. 241–253. DOI: 10.1007/11856290_21.
- [119] M. Kohlhase et al. “The Planetary System: Web 3.0 & Active Documents for STEM”. In: **Proceedings of the International Conference on Computational Science, ICCS 2011, Nanyang Technological University, Singapore, 1-3 June, 2011**. Ed. by M. Sato et al. Vol. 4. Procedia Computer Science. Elsevier, 2011, pp. 598–607. DOI: 10.1016/j.procs.2011.04.063.
- [120] T. H. Koornwinder. “Additions to the formula lists in “Hypergeometric orthogonal polynomials and their q -analogues” by Koekoek, Lesky and Swarttouw”. In: **arXiv:1401.0815v2** (2014).
- [121] G. Y. Kristianto, G. Topic, and A. Aizawa. “Extracting Textual Descriptions of Mathematical Expressions in Scientific Papers”. In: **D-Lib Magazine** 20.11/12 (2014), p. 9. DOI: 10.1045/november14-kristianto.
- [122] G. Y. Kristianto et al. “The MCAT Math Retrieval System for NTCIR-11 Math Track”. 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**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/06-NTCIR11-MATH-KristiantoGY.pdf>.
- [123] C. Lange et al. “The Planetary System: Executable Science, Technology, Engineering and Math Papers”. In: **The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011, Proceedings, Part II**. Ed. by G. Antoniou et al. Vol. 6644. Lecture Notes in Computer Science. Springer, 2011, pp. 471–475. DOI: 10.1007/978-3-642-21064-8_37.
- [124] C. Larman. **Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development**. Pearson Education India, 2005.

- [125] R. R. Larson, C. Reynolds, and F. C. Gey. "The Abject Failure of Keyword IR for Mathematics Search: Berkeley at NTCIR-10 Math". In: **Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-10, National Center of Sciences, Tokyo, Japan, June 18-21, 2013**. Ed. by N. Kando and T. Kato. National Institute of Informatics (NII), 2013. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/02-NTCIR10-MATH-LarsonRR.pdf>.
- [126] M. Leich et al. "Applying Stratosphere for Big Data Analytics". In: **Datenbanksysteme für Business, Technologie und Web (BTW), 15. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), 11.-15.3.2013 in Magdeburg, Germany. Proceedings**. Ed. by V. Markl et al. Vol. 214. LNI. GI, 2013, pp. 507–510. URL: <http://www.btw-2013.de/proceedings/Applying%20Stratosphere%20for%20Big%20Data%20Analytics.pdf>.
- [127] P. Libbrecht. "Escaping the Trap of Too Precise Topic Queries". In: **Intelligent Computer Mathematics - MKM, Calculemus, DML, and Systems and Projects 2013, Held as Part of CICM 2013, Bath, UK, July 8-12, 2013. Proceedings**. Ed. by J. Carette et al. Vol. 7961. Lecture Notes in Computer Science. Springer, 2013, pp. 296–309. DOI: 10.1007/978-3-642-39320-4_20.
- [128] P. Libbrecht and E. Melis. "Methods to Access and Retrieve Mathematical Content in ActiveMath". In: **Mathematical Software - ICMS 2006, Second International Congress on Mathematical Software, Castro Urdiales, Spain, September 1-3, 2006, Proceedings**. Ed. by A. Iglesias and N. Takayama. Vol. 4151. Lecture Notes in Computer Science. Springer, 2006, pp. 331–342. DOI: 10.1007/11832225_33.
- [129] P. Libbrecht et al. "Cross-Curriculum Search for Intergeo". In: **Intelligent Computer Mathematics, 9th International Conference, AISC 2008, 15th Symposium, Calculemus 2008, 7th International Conference, MKM 2008, Birmingham, UK, July 28 - August 1, 2008. Proceedings**. Ed. by S. Autexier et al. Vol. 5144. Lecture Notes in Computer Science. Springer, 2008, pp. 520–535. DOI: 10.1007/978-3-540-85110-3_42.
- [130] X. Lin et al. "A Text Line Detection Method for Mathematical Formula Recognition". In: **2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013. IEEE Computer Society, 2013, pp. 339–343. DOI: 10.1109/ICDAR.2013.75**.
- [131] X. Lin et al. "Mathematical formula identification and performance evaluation in PDF documents". In: **International Journal on Document Analysis and Recognition (IJDAR) 17.3 (2014), pp. 239–255. ISSN: 1433-2825. DOI: 10.1007/s10032-013-0216-1**.

- [132] A. Lipani et al. "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**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/09-NTCIR11-MATH-LipaniA.pdf>.
- [133] X. Liu et al. "Hybrid Clustering of Text Mining and Bibliometrics Applied to Journal Sets". In: **Proceedings of the SIAM International Conference on Data Mining**. Sparks, NV, USA, 2009, pp. 49–60.
- [134] M. Líška. "Evaluation of Mathematics Retrieval". Master's thesis. Masaryk University, Faculty of Informatics, Brno, 2013. URL: http://is.muni.cz/th/255768/fi_m.
- [135] M. Líška. **Vyhledávání v matematickém textu (in Slovak), Searching Mathematical Texts, 2010**. Bachelor Thesis, Masaryk University, Brno, Faculty of Informatics (advisor: Petr Sojka). Bachelor's thesis. Masaryk University, Faculty of Informatics, Brno, 2010. URL: http://is.muni.cz/th/255768/fi_b/ (visited on 10/02/2016).
- [136] M. Líška, P. Sojka, and M. Ruzicka. "Math Indexer and Searcher Web Interface - Towards Fulfillment of Mathematicians' Information Needs". In: **Intelligent Computer Mathematics - International Conference, CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings**. Ed. by S. M. Watt et al. Vol. 8543. Lecture Notes in Computer Science. Springer, 2014, pp. 444–448. DOI: 10.1007/978-3-319-08434-3_36.
- [137] M. Líška, P. Sojka, and M. Ruzicka. "Similarity Search for Mathematics: Masaryk University Team at the NTCIR-10 Math Task". In: **Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-10, National Center of Sciences, Tokyo, Japan, June 18-21, 2013**. Ed. by N. Kando and T. Kato. National Institute of Informatics (NII), 2013. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/06-NTCIR10-MATH-LiskaM.pdf>.
- [138] M. Líška et al. "Web Interface and Collection for Mathematical Retrieval : WebMIaS and MREC". eng. In: **DML 2011: Towards a Digital Mathematics Library**. Ed. by P. Sojka and T. Bouche. Brno: Masaryk University, 2011, pp. 77–84. URL: <https://is.muni.cz/publication/946673/en>.
- [139] D. W. Lozier. "NIST Digital Library of Mathematical Functions". In: **Annals of Mathematics and Artificial Intelligence** 38 (1-3 2003).
- [140] K. Ma, S. C. Hui, and K. Chang. "Feature extraction and clustering-based retrieval for mathematical formulas". In: **Software Engineering and Data Mining (SEDM)**. IEEE. 2010, pp. 372–377.

- [141] S. Maddox. “Mathematical equations in Braille”. In: **Maths, Stats and Operations Research (MSOR) Connections** 7.2 (May 2007), pp. 45–48. ISSN: 1473-4869. DOI: 10.11120/msor.2007.07020045.
- [142] W. Magnus, F. Oberhettinger, and R. P. Soni. **Formulas and theorems for the special functions of mathematical physics**. Third enlarged edition. Die Grundlehren der mathematischen Wissenschaften, Band 52. Springer-Verlag New York, Inc., New York, 1966, pp. viii+508.
- [143] C. D. Manning et al. “The Stanford CoreNLP Natural Language Processing Toolkit”. In: **Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations**. The Association for Computer Linguistics, 2014, pp. 55–60. URL: <http://aclweb.org/anthology/P/P14/P14-5010.pdf>.
- [144] M. Marchiori. “The Mathematical Semantic Web”. In: **Mathematical Knowledge Management, MKM’03**. Ed. by A. Asperti, B. Buchberger, and J. H. Davenport. LNCS 2594. Springer Verlag, 2003.
- [145] D. Mayer. **Help:Displaying a formula — Meta, discussion about Wikimedia projects**. Mar. 20, 2014. URL: http://meta.wikimedia.org/w/index.php?title=Help:Displaying_a_formula%5C&oldid=15233.
- [146] D. McKain. **SnuggleTeX**.
- [147] MediaWiki. **Failed to parse (Cannot store math image on filesystem.)** Mar. 20, 2014. URL: https://www.mediawiki.org/w/index.php?title=Manual_talk:Math&oldid=782277#Failed_to_parse_.28Cannot_store_math_image_on_filesystem_.29.
- [148] MediaWiki. **Manual: Troubleshooting math display errors — MediaWiki, The Free Wiki Engine**. Mar. 20, 2014. URL: https://www.mediawiki.org/wiki/Manual:Troubleshooting_math_display_errors?oldid=798098.
- [149] A. Memon, A. Nagarajan, and Q. Xie. “Automating regression testing for evolving GUI software”. In: **J. of Software Maintenance** 17.1 (2005), pp. 27–64. DOI: 10.1002/smr.305.
- [150] A. M. Memon. “Automatically repairing event sequence-based GUI test suites for regression testing”. In: **ACM Trans. Softw. Eng. Methodol.** 18.2 (2008). DOI: 10.1145/1416563.1416564.
- [151] G. O. Michler. “Report on the retrodigitization project ”Archiv der Mathematik””. In: **Archiv der Mathematik** 77.1 (2001), pp. 116–128.
- [152] I. Mierswa et al. “YALE: Rapid Prototyping for Complex Data Mining Tasks”. In: **Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’06**. Philadelphia, PA, USA: ACM, 2006, pp. 935–940. ISBN: 1-59593-339-5. DOI: 10.1145/1150402.1150531.

- [153] A. Miles et al. "SKOS Core: Simple Knowledge Organisation for the Web". In: **Proceedings of the 2005 International Conference on Dublin Core and Metadata Applications: Vocabularies in Practice**. DCMi '05. Dublin Core Metadata Initiative, 2005, pp. 1–9. ISBN: 9788489315440. URL: <http://dl.acm.org/citation.cfm?id=1383465.1383467>.
- [154] B. Miller. "Drafting DLMF Content Dictionaries". English. Talk presented at the OpenMath workshop of the 9th Conference on Intelligent Computer Mathematics CICM 2016. Bialystok, Poland, July 25, 2016. URL: <http://cicm-conference.org/2016/cicm.php?event=&menu=talks#02> (visited on 10/03/2016).
- [155] B. Miller. "LaTeXML: A LaTeX to XML Converter". In: (2010). URL: <http://dlmf.nist.gov/LaTeXML/>.
- [156] B. R. Miller. "Three Years of DLMF: Web, Math and Search". In: **Intelligent Computer Mathematics - MKM, Calculemus, DML, and Systems and Projects 2013, Held as Part of CICM 2013, Bath, UK, July 8-12, 2013. Proceedings**. Ed. by J. Carette et al. Vol. 7961. Lecture Notes in Computer Science. Springer, 2013, pp. 288–295. DOI: 10.1007/978-3-642-39320-4_19.
- [157] B. R. Miller and A. Youssef. "Augmenting Presentation MathML for Search". In: **Intelligent Computer Mathematics, 9th International Conference, AISC 2008, 15th Symposium, Calculemus 2008, 7th International Conference, MKM 2008, Birmingham, UK, July 28 - August 1, 2008. Proceedings**. Ed. by S. Autexier et al. Vol. 5144. Lecture Notes in Computer Science. Springer, 2008, pp. 536–542. DOI: 10.1007/978-3-540-85110-3_43.
- [158] B. R. Miller and A. Youssef. "Technical Aspects of the Digital Library of Mathematical Functions". In: **Ann. Math. Artif. Intell.** 38.1-3 (2003), pp. 121–136. DOI: 10.1023/A:1022967814992.
- [159] R. Miner. "The Importance of MathML to Communication". In: **Notices of the AMS** 52.5 (2005).
- [160] R. Miner and R. Munavalli. "An Approach to Mathematical Search Through Query Formulation and Data Normalization". In: **Towards Mechanized Mathematical Assistants, 14th Symposium, Calculemus 2007, 6th International Conference, MKM 2007, Hagenberg, Austria, June 27-30, 2007, Proceedings**. Ed. by M. Kauers et al. Vol. 4573. Lecture Notes in Computer Science. Springer, 2007, pp. 342–355. DOI: 10.1007/978-3-540-73086-6_27.
- [161] C. Mishra and N. Koudas. "Interactive query refinement". In: **Proceedings of the 12th International Conference on Extending Database Technology Advances in Database Technology - EDBT '09**. New York, New York, USA: ACM Press, 2009, p. 862. ISBN: 9781605584225. DOI: 10.1145/1516360.1516459.

- [162] J. Mišutka and L. Galamboš. “Extending Full Text Search Engine for Mathematical Content”. In: **Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008**. Brno: Masaryk University, 2008, pp. 55–67. URL: <http://eudml.org/doc/220081>.
- [163] J. Mišutka and L. Galamboš. “Mathematical Extension of Full Text Search Engine Indexer”. In: **Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on**. Apr. 2008, pp. 1–6. DOI: 10.1109/ICTTA.2008.4530006.
- [164] J. Mišutka and L. Galamboš. “System Description: EgoMath2 As a Tool for Mathematical Searching on Wikipedia.org”. In: (2011). Ed. by J. H. Davenport et al., pp. 307–309. DOI: 10.1007/978-3-642-22673-1_30.
- [165] R. Morris. **Bug 54367 - intermittent texvc problems**. Mar. 20, 2014. URL: https://bugzilla.wikimedia.org/show_bug.cgi?id=5436.
- [166] R. Munavalli and R. Miner. “MathFind: A Math-aware Search Engine”. In: **Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. SIGIR ’06. Seattle, Washington, USA: ACM, 2006, pp. 735–735. ISBN: 1-59593-369-7. DOI: 10.1145/1148170.1148348.
- [167] F. Á. Muñoz. “Mathematical Expression Recognition based on Probabilistic Grammars”. PhD thesis. 2015. DOI: 10.4995/thesis/10251/51665.
- [168] S. Murugan. **Bug 54456 - Failed to parse (Cannot store math image on filesystem.)** Mar. 20, 2014. URL: https://bugzilla.wikimedia.org/show_bug.cgi?id=54456.
- [169] G. Navarro. “A guided tour to approximate string matching”. In: **ACM computing surveys** 33.1 (2001), pp. 31–88.
- [170] S. B. Needleman and C. D. Wunsch. “A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins”. In: **Journal of Molecular Biology** 48.3 (1970), pp. 443–453. DOI: 10.1016/0022-2836(70)90057-4.
- [171] Netheril96@gmail.com. **Option “MathML if possible” doesn’t work**. Mar. 20, 2014. URL: https://bugzilla.wikimedia.org/show_bug.cgi?id=25646 (visited on 11/28/2011).
- [172] M.-Q. Nghiem et al. “Mining coreference relations between formulas and text using Wikipedia”. In: August (2010), pp. 69–74.
- [173] M. Nghiem et al. “Which One Is Better: Presentation-Based or Content-Based Math Search?” In: **Intelligent Computer Mathematics - International Conference, CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings**. Ed. by S. M. Watt et al. Vol. 8543. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 200–212. DOI: 10.1007/978-3-319-08434-3_15.

- [174] T. T. Nguyen, K. Chang, and S. C. Hui. “A Math-aware Search Engine for Math Question Answering System”. In: **Proceedings of the 21st ACM International Conference on Information and Knowledge Management**. CIKM '12. Maui, Hawaii, USA: ACM, 2012, pp. 724–733. ISBN: 978-1-4503-1156-4. DOI: 10.1145/2396761.2396854.
- [175] A. Nomura et al. “Detection and segmentation of touching characters in mathematical expressions”. In: **Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings**. Aug. 2003, 126–130 vol.1. DOI: 10.1109/ICDAR.2003.1227645.
- [176] I. Normann and M. Kohlhase. “Extended Formula Normalization for **epsilon**-Retrieval and Sharing of Mathematical Knowledge”. In: **Towards Mechanized Mathematical Assistants, 14th Symposium, Calculemus 2007, 6th International Conference, MKM 2007, Hagenberg, Austria, June 27-30, 2007, Proceedings**. Ed. by M. Kauers et al. Vol. 4573. Lecture Notes in Computer Science. Springer, 2007, pp. 356–370. DOI: 10.1007/978-3-540-73086-6_28.
- [177] N. Oikonomakou and M. Vazirgiannis. “A review of web document clustering approaches”. In: **Data mining and knowledge discovery handbook**. Springer, 2005, pp. 921–943.
- [178] F. W. J. Olver et al., eds. **NIST handbook of mathematical functions**. Cambridge: Cambridge University Press, 2010, pp. xvi+951. ISBN: 978-0-521-14063-8.
- [179] R. Pagel. **MLP Project Repository**. <https://github.com/rbzn/project-mlp>. 2013.
- [180] R. Pagel and M. Schubotz. “Mathematical Language Processing Project”. In: URL: <http://ceur-ws.org/Vol-1186/paper-23.pdf>.
- [181] N. Pattaniyil and R. Zanibbi. “Combining TF-IDF Text Retrieval with an Inverted Index over Symbol Pairs in Math Expressions: The Tangent Math Search Engine at NTCIR 2014”. 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**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/08-NTCIR11-MATH-PattaniyilN.pdf>.
- [182] F. Pedregosa et al. “scikitLearn: Machine Learning in Python”. In: **JMLR 12** (2011), pp. 2825–2830.
- [183] K. M. Phan et al. “An incremental recognition method for online handwritten mathematical expressions”. In: **2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)**. Nov. 2015, pp. 171–175. DOI: 10.1109/ACPR.2015.7486488.

- [184] J. M. G. Pinto, S. Barthel, and W. Balke. “QUALIBETA at the NTCIR-11 Math 2 Task: An Attempt to Query Math Collections”. 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**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/03-NTCIR11-MATH-GonzalezPintoJM.pdf>.
- [185] J. Pitman and C. Lynch. “Planning a 21st century global library for mathematics research”. In: **Notices of the AMS** 61.7 (2014), pp. 776–777.
- [186] H. Prieto, S. Dalmas, and Y. Papegay. “Mathematica As an OpenMath Application”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 22–26. ISSN: 0163-5824. DOI: 10.1145/362001.362016.
- [187] A. P. Prudnikov, Y. A. Brychkov, and O. I. Marichev. **Integrals and series. Vols. 1-5**. New York: Gordon & Breach Science Publishers, 1986.
- [188] F. Rabe. “A Query Language for Formal Mathematical Libraries”. 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**. Ed. by J. Jeuring et al. Vol. 7362. Lecture Notes in Computer Science. Springer, 2012, pp. 143–158. DOI: 10.1007/978-3-642-31374-5_10.
- [189] M. S. Reichenbach, A. Agarwal, and R. Zanibbi. “Rendering expressions to improve accuracy of relevance assessment for math search”. In: **Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14** (2014), pp. 851–854. DOI: 10.1145/2600428.2609457.
- [190] A. H. Renear and C. L. Palmer. “Strategic Reading, Ontologies, and the Future of Scientific Publishing”. In: **Science** 325.5942 (2009), pp. 828–832. DOI: 10.1126/science.1157784. eprint: <http://www.sciencemag.org/content/325/5942/828.full.pdf>.
- [191] M. N. Riem. “The OpenMath Guide. A practical guide on using OpenMath”. In: **Available from the Research Institute for Applications of Computer Algebra** (2004).
- [192] B. Rous. “Major Update to ACM’s Computing Classification System”. In: **Communications of the ACM** 55.11 (Nov. 2012), p. 12. ISSN: 0001-0782. DOI: 10.1145/2366316.2366320.

- [193] J. Rowley. “The wisdom hierarchy: representations of the DIKW hierarchy”. In: **Journal of Information Science** 33.2 (2007), pp. 163–180. DOI: 10.1177/0165551506070706. eprint: <http://jis.sagepub.com/content/33/2/163.full.pdf+html>. URL: <http://jis.sagepub.com/content/33/2/163.abstract>.
- [194] M. Růžicka, P. Sojka, and M. Líska. “Math Indexer and Searcher under the Hood : Fine-tuning Query Expansion and Unification Strategies”. In: **Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies**. Ed. by N. Kando et al. Tokyo Japan: National Institute of Informatics, June 7, 2016. ISBN: 0000000205.
- [195] M. Růžicka, P. Sojka, and M. Líska. “Math Indexer and Searcher under the Hood: History and Development of a Winning Strategy”. 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**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/07-NTCIR11-MATH-RuzickaM.pdf>.
- [196] G. Salton and M. J. McGill. **Introduction to Modern Information Retrieval**. New York, NY, USA: McGraw-Hill, Inc., 1986. ISBN: 0070544840.
- [197] U. Schöneberg and W. Sperber. “POS Tagging and its Applications for Mathematics”. In: **Intelligent Computer Mathematics**. Springer, 2014, pp. 213–223.
- [198] E. Schröder. “Über Pasigraphie, ihren gegenwärtigen Stand und die pasigraphische Bewegung in Italien”. In: **Verhandlungen des Ersten Internationalen Mathematiker-Kongresses in Zürich vom 9 (1898)**, pp. 147–162. URL: <http://www.mathunion.org/ICM/ICM1897/Main/icm1897.0147.0162.ocr.pdf>.
- [199] M. Schubotz. **Extension:MathSearch - MediaWiki**. 2012. URL: <http://www.mediawiki.org/wiki/Extension:MathSearch> (visited on 10/05/2012).
- [200] M. Schubotz. **Formulasearchengine. mlp.formulasearchengine.com**. 2012. URL: <http://mlp.formulasearchengine.com> (visited on 04/01/2016).
- [201] M. Schubotz. “Full Counting Statistics. A quantum master equation approach”. English. Diplomarbeit. Berlin: Institut für theoretische Physik an der Fakultät für Mathematik und Naturwissenschaften an der Technische Universität Berlin, 2011, p. 189. unpublished.
- [202] M. Schubotz. “Making Math Searchable in Wikipedia”. In: **CoRR** abs/1304.5475 (July 30, 2012). DOI: 10.14279/depositonce-5034. arXiv: 1304.5475.
- [203] M. Schubotz and T. Brandes. “Random backaction in tunneling of single electrons through nanostructures”. In: **Physical Review B** 84.7 (Aug. 2011), pp. 1–8. ISSN: 1098-0121. DOI: 10.1103/PhysRevB.84.075340.

- [204] M. Schubotz, M. Leich, and V. Markl. “Querying Large Collections of Mathematical Publications: NTCIR10 Math Task”. In: **Querying large Collections of Mathematical Publications. Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-10, National Center of Sciences, Tokyo, Japan, June 18-21, 2013**. Ed. by N. Kando and T. Kato. National Institute of Informatics (NII), 2013, pp. 667–674. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/03-NTCIR10-MATH-SchubotzM.pdf>.
- [205] M. Schubotz and A. P. Sexton. “A Smooth Transition to Modern mathoid-based Math Rendering in Wikipedia with Automatic Visual Regression Testing”. In: **Joint Proceedings of the MathUI, OpenMath and ThEdu Workshops and Work in Progress track at CICM**. (Bialistok, Poland, July 19, 2016). Ed. by M. Kohlhase. CEUR Workshop Proceedings ? Aachen, 2016. DOI: 10.13140/RG.2.1.3961.1124. URL: <http://ceur-ws.org/>.
- [206] M. Schubotz, D. Veenhuis, and H. S. Cohl. “Getting the units right”. In: **Joint Proceedings of the MathUI, OpenMath and ThEdu Workshops and Work in Progress track at CICM**. (Bialistok, Poland, July 19, 2016). Ed. by M. Kohlhase. CEUR Workshop Proceedings ? Aachen, 2016. DOI: 10.13140/RG.2.1.2508.0561. URL: <http://ceur-ws.org/>.
- [207] M. Schubotz and G. Wicke. “Mathoid: Robust, Scalable, Fast and Accessible Math Rendering for Wikipedia”. English. In: **Intelligent Computer Mathematics**. Ed. by S. Watt et al. Vol. 8543. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 224–235. ISBN: 978-3-319-08433-6. DOI: 10.1007/978-3-319-08434-3_17.
- [208] M. Schubotz et al. “Challenges of Mathematical Information Retrieval in the NTCIR-11 Math Wikipedia Task”. In: **Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval**. Ed. by R. A. Baeza-Yates et al. SIGIR ’15. Santiago, Chile: ACM, 2015, pp. 951–954. ISBN: 978-1-4503-3621-5. DOI: 10.1145/2766462.2767787.
- [209] M. Schubotz et al. “Evaluation of Similarity-Measure Factors for Formulae Based on the NTCIR-11 Math Task”. In: **Evaluation of Similarity-Measure Factors for Formulae. Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014**. Ed. by N. Kando, H. Joho, and K. Kishida. National Institute of Informatics (NII), 2014. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/04-NTCIR11-MATH-SchubotzM.pdf>.
- [210] M. Schubotz et al. “Exploring the One-Brain-Barrier: a Manual Contribution to the NTCIR -12 Math Task”. In: **Exploring the One-Brain-Barrier**.

- Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR -12)**. 2016.
- [211] M. Schubotz et al. “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. Pisa, Italy: ACM, 2016, pp. 135–144. ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2911503.
 - [212] C. Schulman-Galambos and R. Galambos. “Cortical responses from adults and infants to complex visual stimuli.” eng. In: **Electroencephalography and clinical neurophysiology** 45 (4 Oct. 1978), pp. 425–35.
 - [213] M. Schwarzer et al. “Evaluating Link-based Recommendations for Wikipedia”. In: **Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, JCDL 2016, Newark, NJ, USA, June 19 - 23, 2016**. Ed. by N. R. Adam et al. ACM, 2016, pp. 191–200. DOI: 10.1145/2910896.2910908.
 - [214] A. P. Sexton. “Abramowitz and Stegun - A Resource for Mathematical Document Analysis”. In: **Conferences on Intelligent Computer Mathematics (CICM 2012)**. Ed. by J. Jeuring et al. Vol. 7362. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2012, pp. 159–168. ISBN: 978-3-642-31373-8. DOI: 10.1007/978-3-642-31374-5_11.
 - [215] M. Shatnawi and A. Youssef. “Equivalence detection using parse-tree normalization for math search”. In: **Second IEEE International Conference on Digital Information Management (ICDIM), December 11-13, 2007, Lyon, France, Proceedings**. IEEE, 2007, pp. 643–648. DOI: 10.1109/ICDIM.2007.4444297.
 - [216] L. A. Sobreviela. “A Reduce-based OpenMath ↔ MathML Translator”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 31–32. ISSN: 0163-5824. DOI: 10.1145/362001.362018.
 - [217] N. Soiffer. “MathPlayer”. In: **Proceedings of the 7th international Association for Computing Machinery Special Interest Group on Accessible Computing Conference on Computers and Accessibility – ASSETS 2005**. New York, New York, USA: ACM Press, 2005, p. 204. ISBN: 1595931597. DOI: 10.1145/1090785.1090831.
 - [218] P. Sojka and M. Líska. “Indexing and Searching Mathematics in Digital Libraries – Architecture, Design and Scalability Issues”. eng. In: **Intelligent Computer Mathematics Lecture Notes in Computer Science**. Ed. by J. H. Davenport et al. Vol. 6824. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 228–243. URL: <https://is.muni.cz/publication/945754/en>.

- [219] P. Sojka and M. Líska. “The Art of Mathematics Retrieval”. In: **Proceedings of the 11th ACM Symposium on Document Engineering**. DocEng ’11. Mountain View, California, USA: ACM, 2011, pp. 57–60. ISBN: 978-1-4503-0863-2. DOI: 10.1145/2034691.2034703.
- [220] V. Sorge et al. “Towards making mathematics a first class citizen in general screen readers.” In: **11th Web for All Conference**. Seoul, Korea, 6–9 April 2014: ACM, 2014.
- [221] H. Stamerjohanns et al. “MathML-aware article conversion from LaTeX, A comparison study”. In: **Towards Digital Mathematics Library, DML 2009 workshop**. Ed. by P. Sojka. Masaryk University, Brno, 2009, pp. 109–120. URL: <http://kwarc.info/kohlhase/submit/dml09.pdf>.
- [222] H. Stamerjohanns et al. “Transforming Large Collections of Scientific Publications to XML”. English. In: **Mathematics in Computer Science 3.3** (2010), pp. 299–307. ISSN: 1661-8270. DOI: 10.1007/s11786-010-0024-7.
- [223] C. Stokoe, M. P. Oakes, and J. Tait. “Word sense disambiguation in information retrieval revisited”. In: **SIGIR**. ACM. 2003, pp. 159–166.
- [224] A. Strotmann and L. Kohout. “OpenMath: Compositionality Achieved at Last”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 66–72. ISSN: 0163-5824. DOI: 10.1145/362001.362024.
- [225] M. Suzuki and K. Yamaguchi. “Recognition of E-Born PDF Including Mathematical Formulas”. In: **Computers Helping People with Special Needs: 15th International Conference, ICCHP 2016, Linz, Austria, July 13–15, 2016, Proceedings, Part I**. Ed. by K. Miesenberger, C. Bühler, and P. Penaz. Cham: Springer International Publishing, 2016, pp. 35–42. ISBN: 978-3-319-41264-1. DOI: 10.1007/978-3-319-41264-1_5.
- [226] M. Suzuki et al. “INFTY: An integrated OCR system for Mathematical Documents”. In: **Proceedings of the 2003 ACM symposium on Document engineering**. ACM. 2003, pp. 95–104.
- [227] W. Sylwestrzak et al. **EuDML-Towards the European Digital Mathematics Library**. In: **Towards a Digital Mathematics Library**. Ed. by P. Sojka. Masaryk University Press, 2010, pp. 11–26.
- [228] E. Tapia and R. Rojas. “Recognition of on-line handwritten mathematical expressions in the E-Chalk system - an extension”. In: **Eighth International Conference on Document Analysis and Recognition (ICDAR’05)**. Institute of Electrical and Electronics Engineers (IEEE), 2005. DOI: 10.1109/icdar.2005.197.
- [229] E. M. Taranta II et al. “A Dynamic Pen-Based Interface for Writing and Editing Complex Mathematical Expressions With Math Boxes”. In: **ACM Trans. Interact. Intell. Syst.** 6.2 (July 2016), 13:1–13:25. ISSN: 2160-6455. DOI: 10.1145/2946795.
- [230] S. Thorpe, D. Fize, C. Marlot, et al. “Speed of processing in the human visual system”. In: **nature** 381.6582 (1996), pp. 520–522.

- [231] G. Topic et al. “The MCAT Math Retrieval System for NTCIR-10 Math Track”. In: **Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-10, National Center of Sciences, Tokyo, Japan, June 18-21, 2013**. Ed. by N. Kando and T. Kato. National Institute of Informatics (NII), 2013. URL: <http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/05-NTCIR10-MATH-TopicG.pdf>.
- [232] A. Triulzi. “OpenMath Support Under CSL-hosted REDUCE”. In: **SIGSAM Bull.** 34.2 (June 2000), pp. 27–30. ISSN: 0163-5824. DOI: 10.1145/362001.362017.
- [233] User:Nageh. **User:Nageh/mathJax**. Mar. 20, 2014. URL: <https://en.wikipedia.org/w/index.php?title=User:Nageh/mathJax%5C&oldid=400482894>.
- [234] **Varnish HTTP Cache**. <https://varnish-cache.org/>. Accessed: 25 June 2016. 2016.
- [235] B. Vibber. **Disable the partial HTML and MathML rendering options for Math extension**. Mar. 20, 2014. URL: <https://github.com/wikimedia/mediawiki-extensions-Math/commit/09679f2f39e6c6c00e87757292421b26bfa7022a>.
- [236] B. Vibber. **Experimental option \$wgMathUseMathJax to have Extension:Math load things via MathJax**. Mar. 20, 2014. URL: <https://github.com/wikimedia/mediawiki-extensions-Math/commit/1042006fd4c2cbe6c62619b860e2e234d04d6d38>.
- [237] A. Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. In: **IEEE Transactions on Information Theory** 13.2 (Apr. 1967), pp. 260–269. ISSN: 0018-9448. DOI: 10.1109/TIT.1967.1054010.
- [238] E. M. Voorhees. “The TREC-8 Question Answering Track Report.” In: **TREC** (1999). URL: http://trec.nist.gov/pubs/trec8/papers/qa%5C_report.pdf.
- [239] E. M. Voorhees, D. K. Harman, et al. **TREC: Experiment and evaluation in information retrieval**. Vol. 1. MIT press Cambridge, 2005.
- [240] D. Vrandečić. “Wikidata: a new platform for collaborative data collection”. In: **Proceedings of the 21st international conference companion on World Wide Web**. ACM. 2012, pp. 1063–1064.
- [241] R. Wagner and M. Fischer. “The string-to-string correction problem”. In: **Journal of the ACM (JACM)** 21.1 (Jan. 1974), pp. 168–173. ISSN: 0004-5411. DOI: 10.1145/321796.321811.
- [242] F. Wang. **Native MathML**. seen May, 2016. May 20, 2016. URL: <https://addons.mozilla.org/en-US/firefox/addon/native-mathml>.

- [243] K. D. V. Wangari, R. Zanibbi, and A. Agarwal. “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 - SIGIR '14** (2014), pp. 947–950. DOI: 10.1145/2600428.2609481.
- [244] T. Wegrzanowski. **Texvc: TEX Validator and Converter**. 2001. URL: <http://en.wikipedia.org/wiki/Texvc> (visited on 10/05/2012).
- [245] P. Willett. “The Porter stemming algorithm: then and now”. In: **Program: electronic library and information systems** 40.3 (2006), pp. 219–223.
- [246] M. Wolska and M. Grigore. “Symbol Declarations in Mathematical Writing”. In: **Towards a Digital Mathematics Library. Paris, France, July 7-8th, 2010**. Brno, Czech Republic: Masaryk University Press, 2010, pp. 119–127. URL: <http://eudml.org/doc/221086>.
- [247] K. Yokoi and A. Aizawa. “An Approach to Similarity Search for Mathematical Expressions using MathML”. In: **Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009**. Brno: Masaryk University Press, 2009, pp. 27–35. URL: <http://eudml.org/doc/221460>.
- [248] K. Yokoi et al. “Contextual Analysis of Mathematical Expressions for Advanced Mathematical Search”. In: **CICLing**. Vol. 43. 2010, pp. 20–26. URL: <http://polibits.cidetec.ipn.mx/ojs/index.php/polibits/article/view/43-11/1774>.
- [249] S. Yoo and M. Harman. “Regression testing minimization, selection and prioritization: a survey”. In: **Softw. Test., Verif. Reliab.** 22.2 (2012), pp. 67–120. DOI: 10.1002/stv.430.
- [250] R. Youngen. “Toward a Mathematical Markup Language”. In: **Notices of the AMS** (1997), pp. 1107–1109.
- [251] A. Youssef. “Mathematical Knowledge Management: 5th International Conference, MKM 2006, Wokingham, UK, August 11-12, 2006. Proceedings”. In: ed. by J. M. Borwein and W. M. Farmer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 2–16. ISBN: 978-3-540-37106-9. DOI: 10.1007/11812289_2.
- [252] A. Youssef. “Methods of Relevance Ranking and Hit-content Generation in Math Search”. In: **Towards Mechanized Mathematical Assistants, 14th Symposium, Calculemus 2007, 6th International Conference, MKM 2007, Hagenberg, Austria, June 27-30, 2007, Proceedings**. Ed. by M. Kauers et al. Vol. 4573. Lecture Notes in Computer Science. Springer, 2007, pp. 393–406. DOI: 10.1007/978-3-540-73086-6_31.
- [253] A. Youssef. “Roles of Math Search in Mathematics”. In: **Mathematical Knowledge Management, 5th International Conference, MKM 2006, Wokingham, UK, August 11-12, 2006, Proceedings**. Ed. by J. M. Borwein and W. M. Farmer. Vol. 4108. Lecture Notes in Computer Science. Springer, 2006, pp. 2–16. DOI: 10.1007/11812289_2.

- [254] A. Youssef. "Search of Mathematical Contents: Issues And Methods". In: **Proceedings of the ISCA 14th International Conference on Intelligent and Adaptive Systems and Software Engineering, July 20-22, 2005, Novotel Toronto Centre, Toronto, Canada**. Ed. by R. T. Hurley and W. Feng. ISCA, 2005, pp. 100–105.
- [255] A. S. Youssef. "Relevance Ranking and Hit Description in Math Search". In: **Mathematics in Computer Science 2.2** (2008), pp. 333–353. DOI: 10.1007/s11786-008-0057-3.
- [256] A. S. Youssef and M. E. Altamimi. "An extensive math query language". In: **16th International Conference on Software Engineering and Data Engineering (SEDE-2007), July 9-11, 2007, Imperial Palace Hotel Las Vegas, Las Vegas, Nevada, USA, Proceedings**. Ed. by H. Al-Mubaid and M. Garbey. ISCA, 2007, pp. 57–63.
- [257] A. Youssef and M. Shatnawi. "Math search with equivalence detection using parse-tree normalization". In: **The 4th International Conference on Computer Science and Information Technology**. 2006.
- [258] E. Zachte. **SquidReportClients@stats.wikimedia.org**. Mar. 20, 2014. URL: http://stats.wikimedia.org/archive/squid_reports/2013-10/SquidReportClients.htm.
- [259] R. Zanibbi and D. Blostein. "Recognition and retrieval of mathematical expressions". In: **International Journal on Document Analysis and Recognition (IJDAR)** 15.4 (2012), pp. 331–357. ISSN: 1433-2825. DOI: 10.1007/s10032-011-0174-4.
- [260] R. Zanibbi et al. "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. ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2911512.
- [261] Q. Zhang and A. Youssef. "An Approach to Math-Similarity Search". English. In: **Intelligent Computer Mathematics - International Conference, CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings**. Ed. by S. Watt et al. Vol. 8543. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 404–418. ISBN: 978-3-319-08433-6. DOI: 10.1007/978-3-319-08434-3_29.
- [262] J. Zhao, M.-Y. Kan, and Y. L. Theng. "Math Information Retrieval: User Requirements and Prototype Implementation". In: **Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries**. JCDL '08. Pittsburgh PA, PA, USA: ACM, 2008, pp. 187–196. ISBN: 978-1-59593-998-2. DOI: 10.1145/1378889.1378921.

Acronyms

- BMP first** The Bateman Manuscript Project deals about the digitalization of the Bateman manuscripts. See resolver.caltech.edu/CaltechAUTHORS:20140123-101456353 for further information. 13, 32, 34
- CICM** The Conference on Intelligent Computer Mathematics is an annual international conference. See cicm-conference.org/ for further information. 29, 30, 40, 78, 85, 110–112, 115, 121
- DLMF** The NIST Digital Library of Mathematical Functions is the digital version of the book ‘NIST Handbook of Mathematical Functions’. See dlmf.nist.gov/ for further information. 12, 30–35, 37–39
- DML** Digital Mathematics Library is a term that describes the digitalization efforts on mathematical literature. 3, 9, 12, 25
- DRMF** The NIST Digital Repository of Mathematical Formulae is a follow up project from the DLMF project. See Section 3.1 for further information.. ix, 5, 12, 29–39, 77, 93, 107, 110, 116, 151, 153
- FDM** Mathematical Formula Data Management is a classification schema for the computer science research problems occurring in Mathematical Information Retrieval. See Chapter 2 for further information. 4, 7, 8, 27–29, 106, 109, 110
- FHP** Formula Home Page is a concept to represent context-free formula semantics. See Section 3.1 for further information. v, 5, 9, 30–32, 34, 38, 39, 93, 110
- FOL** First Order Logic is a mathematical logic model that uses quantified variables. 26
- IN** Information Need describes the desire of a human to retrieve information. v, vi, 3, 4, 22, 27, 46, 67, 77, 79, 85, 91, 107, 109, 110, 112, 122
- IR** Information Retrieval is a subfield of computer science. v, 2, 4, 12, 19, 22, 84, 85, 105, 106, 108, 115, 116, 118
- JCDL** Joint Conference on Digital Libraries is a major conference in computer science. See www.jcdl.org for further information. 113
- LaTeXML** LaTeXML is an LaTeX to HTML converter. See dlmf.nist.gov/LaTeXML/ for further information. 19, 20, 31, 32, 38, 56, 69, 70, 72–76, 79, 81, 117–119, 149

- LRS** Literature Recommender Systems automatically recommend literature. 2
- MAP** Mean Average Precision is an Information Retrieval measure. 87, 132, 134
- MathML** Mathematical Markup Language is a XML-based language for describing mathematical expressions. See Section 2.1.3.1 for further information. 9, 11, 15–20, 24, 26, 31, 32, 37–39, 67–76, 79, 84, 100, 115, 117–119, 121, 126, 127, 129, 130, 134, 137, 138, 140, 141, 149, 152
- MediaWiki** MediaWiki is a Wiki software provided by the Wikimedia Foundation. See www.mediawiki.org for further information. 29–33, 38, 39, 68–76, 84, 86, 106–108, 113, 117–120, 137, 139, 140, 151, 152
- MIR** Mathematical Information Retrieval deals with the application of Information Retrieval methods to STEM texts. v, 2–5, 7, 8, 18, 22, 24–27, 29, 39–41, 55, 66, 67, 77, 78, 80, 81, 83, 85, 92, 93, 100, 106–109, 111–113, 115, 121, 152, 183
- MKM** Mathematical Knowledge Management is a research area at the intersection between mathematics and computer science. See www.mkm-ig.org/ for further information. 3
- MLP** Mathematical Language Processing is an extension of Natural Language Processing for STEM texts containing mathematical expressions. See Section 3.2 for further information. v, 5, 29, 39–43, 45, 47, 48, 50, 56, 57, 59, 63–65, 107, 111, 151
- MRR** The Mean Reciprocal Rank is a typical evaluation measure for ‘known item’ information retrieval tasks. 82–84
- MWS** MathWebSearch is a formula search engine. See search.mathweb.org/ for further information. 23, 26, 27, 95, 106, 115, 119, 122
- NII** National Institute of Informatics is the national informatics research institution of Japan. See www.nii.ac.jp for further information. 81, 83
- NIST** The National Institute of Standards and Technology is an US government research institution. See nist.gov for further information. ix, 13, 29–33, 36, 37, 69, 76, 107
- NLP** Natural Language Processing is an Information Retrieval subtopic. vi, 40, 41, 45, 53, 56, 105, 107, 127
- NTCIR** NII Testbeds and Community for Information access Research is a major international information retrieval task collection, organized by the National Institute of Informatics in Japan. See research.nii.ac.jp/ntcir/ for further information. 22, 23, 26, 57, 58, 65, 77–87, 90–94, 97, 100, 101, 106–109, 112, 113, 115, 116, 122–125, 130, 153, 183
- OCR** Optical Character Recognition deals with the disambiguation of characters in a scan. 9, 10, 13, 14, 18, 33, 39, 144
- OPSF** Orthogonal Polynomials and Special Functions is a field of mathematics. 30–34, 37–39
- PNG** Portable Network Graphics is an raster image format. 67, 68, 70, 71, 75, 76, 138, 142, 143
- POS** Part Of Speech tags are used in Natural Language Processing. 42, 43, 50, 51, 56

- SIGIR first** Special Interest Group on Information Retrieval is a computer science conference. ix, 40, 77, 111, 112
- STEM** Science, Technology, Engineering and Mathematics v, 2, 3, 10, 12, 18, 40, 85, 94, 109
- SVG** Scalable Vector Graphics is a standard data format. See www.w3.org/TR/SVG11/ for further information. 18, 19, 67, 68, 70–76, 138, 140, 141
- TFIDF** Term Frequency Inverse Document Frequency is a method to normalize word frequencies in Natural Language Processing. 26, 45, 129, 131, 135
- WMC** The Math Wikipedia Task is a subtask of the NTCIR math tasks. See ntcir11-wmc.nii.ac.jp/ for further information. v, 57, 78, 80, 83, 85, 86, 94, 101, 109, 111, 112, 151
- XML** Extensible Markup Language is a standard data format. See www.w3.org/TR/xml/ for further information. 15, 26, 34, 38, 81, 125–127, 130