Automatic Mathematical Information Retrieval to Perform Translations up to Computer Algebra Systems

Presentation of the Planned Doctoral Thesis

André Greiner-Petter
University of Konstanz
andre.greiner-petter@uni-konstanz.de

ABSTRACT

In mathematics, LaTeX is the de facto standard to prepare documents, e.g., scientific publications. While some formulae are still developed using pen and paper, more complicated mathematical expressions used more and more often with computer algebra systems. Mathematical expressions are often manually transcribed to computer algebra systems. The goal of my doctoral thesis is to improve the efficiency of this workflow. My envisioned method will automatically semantically enrich mathematical expressions so that they can be imported to computer algebra systems and other systems which can take advantage of the semantics, such as search engines or automatic plagiarism detection systems. These imports should preserve essential semantic features of the expression.

CCS CONCEPTS

• **Information systems → Mathematics retrieval**; • **Software and its engineering → Semantics**; • **Computing methodologies** → *Representation of mathematical objects*; Computer algebra systems;

KEYWORDS

LaTeX, computer algebra systems, mathematical information retrieval, semantification

ACM Reference Format:
André Greiner-Petter. 2018. Automatic Mathematical Information Retrieval to Perform Translations up to Computer Algebra Systems: Presentation of the Planned Doctoral Thesis. In *Proceedings of ACM/IEEE Joint Conference on Digital Libraries in 2018 (JCDL 2018).* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.475/123$_4$

1 PROBLEM & MOTIVATION

The general problem of enriching mathematical expressions with semantic information and providing lossless translations to computer algebra systems can be divided into two parts: the translation process and the semantic enrichment process, hereafter called semantification. The first part was the focus of my Master´s thesis (see section 3.1). The focus of my doctoral research will lie on the second part of the problem. Providing extractions automatically for semantic information of mathematical expressions is worthwhile for several different tasks and can be examined from different areas of interests. For example, semantic knowledge strongly improves mathematical search engines [18, 20]. Imagine someone has the famous Pythagorean theorem with different variable names, e.g. $q^2 + p^2 = r^2$. Finding a connection between this formula and the Pythagorean theorem is still a difficult task for nowadays search engines. Another, rather new field of interest is automatic plagiarism detection in mathematical expressions [16]. Detecting plagiarism in a mathematical formula is highly imprecise, if the semantics of the formula is unknown. As seen in the example above, finding the connection between $q^2 + p^2 = r^2$ and the Pythagorean theorem probably remains unrecognized, which makes it difficult to see a potential plagiarism in this formula.

During this extended abstract, I will explain the difficulties of an automatic semantification process on examples related to problems we have for converting mathematical expressions to computer algebra systems. In the following, I will give a brief introduction to this workflow and explain its issues in a more detailed way.

Scientists usually work with word processors to write scientific papers. The very well-known word processor LaTeX[1] has become a de facto standard[2] for this purpose over the last 30 years [8]. Also, numerous other editors, such as the editor for Wikipedia articles or Microsoft Word, entirely or partially support LaTeX expressions. LaTeX, developed by Leslie Lamport, extends the typesetting system TeX, developed by Donald E. Knuth 1977 [14, p. 559], by a set of standard macros. Knuth created the TeX system because he was unsatisfied with the typography of his book, The Art of Computer Programming [14, pp. 5-6, 24]. LaTeX and TeX provide a syntax for printing mathematical formulae in a way a person would write it by hand. During my research, I will primarily focus on LaTeX formats.

Besides that, scientists work with formulae in their papers while they evaluate special values, create diagrams and find or calculate practical solutions. Computer algebra systems (CAS) are software tools that allow such computations on mathematical expressions. Since CAS should be easy to use, they created their representations with the intent of displaying the results as intuitively as possible. That approach for representing formulae is not standardized. That is why most CAS have created their representation, all of which are different.

Mathematical expressions written in LaTeX, which use only standard libraries of macros (hereafter called generic LaTeX), do not provide sufficient semantic information about the used symbols. In

JCDL 2018, June 2018, Fort Worth, TX

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123$_4$

[1] Since LaTeX is more than just a word processor, calling LaTeX that way might not be accurate enough. But during our research, we compared the theoretical concepts behind LaTeX and word processors such as Microsoft Word and we decided to keep this description.
[2] https://www.latex-project.org/

Systems	Representations
Rendered Version	$P_n^{(\alpha,\beta)}(\cos(a\Theta))$
Generic LaTeX	`P_n^{(\alpha, \beta)}(\cos(a\Theta))`
DLMF/DRMF macros	`\JacobiP{\alpha}{\beta}{n}@{\cos@{a\Theta}}`
CAS Maple	`JacobiP(n, alpha, beta, cos(a Theta))`
CAS Mathematica	`JacobiP[n, \[Alpha], \[Beta], Cos[a \[CapitalTheta]]]`

Table 1: Example of different representations for a Jacobi polynomial

contrast to that, a certain level of semantic information is necessary for CAS to offer correct computations on input expressions. The exact semantics concluded by readers is based on their knowledge and in which context the expression is used. Without a translation tool, a typical workflow for scientists contains different representations for the same mathematical expression: a representation in a word processor such as LaTeX, and another representation in a CAS such as Maple or Mathematica. Table 1 in section 2 describes different representations for the same mathematical expression.

The semantic enrichment problem can be made clear with the description of a single symbol expression. Consider the Euler-Mascheroni constant represented by the Greek letter γ. Without any further information, γ is just a Greek letter, often used to represent this mathematical constant, but can also be used to represent curve parametrization, and many other things, such as a variable. In LaTeX, γ is represented as `\gamma`. The equivalent representation in a CAS, such as Maple or Mathematica, depends on the meaning of γ. For example, in Maple and Mathematica, the Euler-Mascheroni constant is represented as `gamma` and `EulerGamma` respectively. Another more complicated example might be Euler´s number e, which is well-known to be the constant used for the basis of the natural logarithm. Translated to Mathematica, it is just a capital `E`, while Maple has no specific symbol set aside to represent the constant, and one has to evaluate the exponential function using `exp(1)`.

The above examples indicate two potential problems. A translation with semantic information might be difficult because one needs to be aware of the details of the CAS. For instance, a scientist who usually works with Mathematica might not know that `gamma` in Maple is not only a Greek letter but also the constant. Therefore, an automated equivalent translation between representations is desirable. The other problem occurs when the semantic information is not sufficient or is essentially absent in the expression. For example, consider the mass-energy equivalence formula $E = mc^2$. In plain LaTeX `E`, `m` and `c` are Latin letters and one has to analyze the context of each symbol to pick the correct semantic information. It is essential to clarify the correct semantics before translating an expression to another representation. Otherwise, `E` might be mistakenly translated as a Latin letter to `"E"` in Mathematica instead of `EE` to represent the energy.

2 STATE OF THE ART

Unfortunately, there is no agreement on how much semantic information is necessary for an expression to be sufficiently semantic. It always depends on the task. For example, in most cases, no semantic information is necessary to render a mathematical expression. For a search engine, providing the name of the formula might be adequate. But for computations on mathematical expressions, detailed knowledge about the definitions becomes necessary. In this context, we suggest that a mathematical expression is sufficiently semantic when a translation to a CAS becomes feasible.

Before we can start a semantification process, we need a system that is capable of carrying such information. There are several approaches for attaching semantic information to symbols or entire expressions. One approach is the content Mathematical Markup Language (content MathML) [24], which tries to organize semantic information in an XML document. MathML and content MathML are widely used for web services because of their simple and easy to parse XML structure. The JOBAD architecture [6, 10] uses content MathML to create web documents with access to semantic information. While XML is an appropriate format for algorithms, it is not convenient for humans.

The National Institute of Standards and Technology (NIST) in Maryland, USA, has developed a set of LaTeX macros to tie specific character sequences to well-defined mathematical objects and thereby providing semantic information within LaTeX expressions. NIST uses these macros for the Digital Library of Mathematical Functions (DLMF) [17] and the Digital Repository of Mathematical Formulae (DRMF) [2, 3], an outgrowth of the DLMF project. Thereby we call these set of macros *DLMF/DRMF macros*.

Table 1 gives an overview of several different representations with and without semantic information for a Jacobi polynomial. The rendered version illustrates how a scientist would write the expression by hand. As previously explained, the generic LaTeX expression and the rendered version does not provide sufficient semantic information in the source, while the DLMF/DRMF macro, Maple and Mathematica representations are tied to specific definitions.

Besides the noticeable visible differences in used parentheses or the ordering of the arguments, more complicated and hidden differences may appear for multi-valued functions. CAS usually define branch cuts to compute principle values for multi-valued functions. Thereby, CAS implementations of multi-valued functions are discontinuous. Moreover, the position of these branch cuts varies from CAS to CAS [5]. It is important that CAS users understand the position of such cuts [7]. For example, the parabolic cylinder function $U(a, z)$ is defined as a solution of a second order differential equation[3]. $U(a, z)$ can also be represented by other functions, such as the modified Bessel function of the second kind $K_\nu(z)$. One relation between U and K_ν is:

$$U(0, z) = \sqrt{\frac{z}{2\pi}} K_{\frac{1}{4}}\left(\frac{1}{4}z^2\right), \quad \text{for } z \in \mathbb{C}. \tag{1}$$

[3] http://dlmf.nist.gov/12.2#SS1.p1

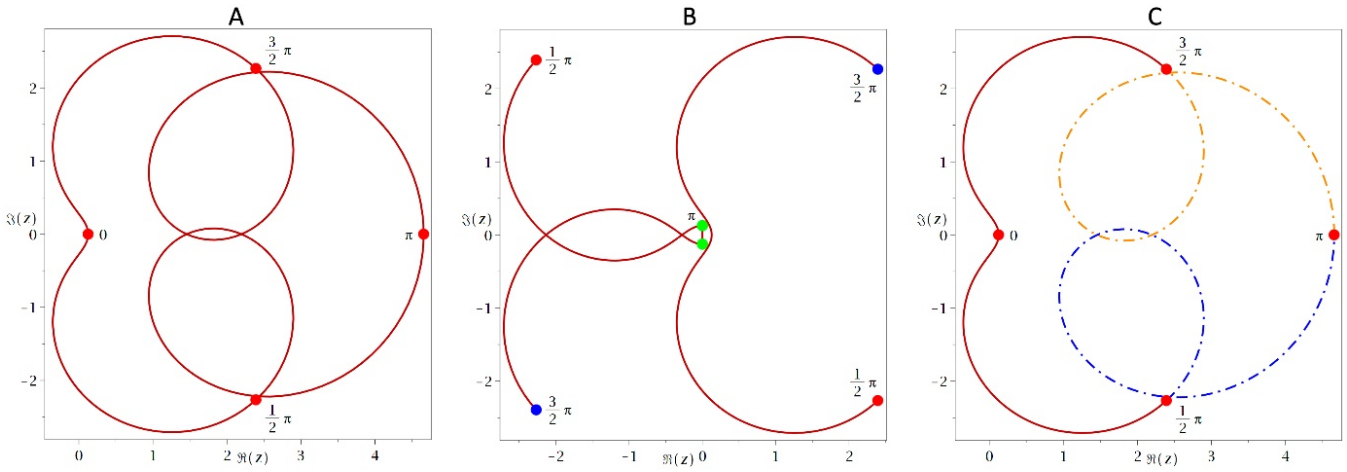

Figure 1: Polar plot for the parabolic cylinder function with $z(\phi) = 2.5e^{i\phi}$ for $\phi \in [0, 2\pi]$. Subfigure A shows $U(0, z(\phi))$. Subfigure B shows the right-hand side of DLMF 12.7.10. Subfigure C uses analytic continuation to allow computations on other branches.

Figure 1 visualizes the problem of this relation using polar plots, for fixed $r = 2.5$ in the polar representation of complex numbers $z = re^{i\phi}$. Figure 1.A draws a curve for $0 \le \phi \le 2\pi$ of $U(0, z(\phi))$. The curve is continuous and closed. Figure 1.B draws the same curve for the principal values of the right-hand side of the equation as computed by a CAS. The function jumps over branch cuts at the angles $\phi \in \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. Based on the relation above, Figure 1.B should yield to the same curve as Figure 1.A. The solution for this problem is analytic continuation which allows computations on other branches. Figure 1.C shows the right-hand side of the equation with analytic continuation for $K_\nu(z)$. The orange $\pi \le \phi \le \frac{3}{2}\pi$ and blue $\frac{1}{2}\pi \le \phi \le \pi$ curves visualize the respective branches, computed with analytic continuation, while the red $0 \le \phi \le \frac{1}{2}\pi$, $\frac{3}{2}\pi \le \phi \le 2\pi$ curve is the principal value. If CAS users use B rather than C for their computations, this leads to incorrect conclusions.

Most CAS provide import and export functions to other representations (see [4, 9, 12]). However, most import functions are only provided for presentations that already carry semantic information, foremost content MathML. While import and export functions rarely take care of system-specific differences [4], such as in case of different branch cut definitions from the example above, the problems become more serious when the semantic in the representation is unclear. In this case, the tool has to be able to extract the semantic information first to allow a translation. To my knowledge, there is no translation tool available that solves this problem. A typical workaround is to use translation tools, that can transform mathematical LaTeX to MathML formats. These tools also contain a lot of problems and cannot handle the mentioned hidden differences. We already studied those tools and their accuracy in a paper [21] (see section 3.2).

However, finding ways of an automatic semantification process is an active topic of research. Nowadays, most semantification algorithms focusing on natural languages, which leads to the well-studied and broad field of natural language processing (NLP). In this respect, a relatively new field of interest is to extend this research for mathematical expressions. Existing approaches (and work in progress projects) try to adapt the approaches from NLP [1, 3, 15, 18–23, 25]. Why this adaption process can be error-prone will be shown by an example I discovered during my research. Our test formula[4] explains the continuity of a two-valued function f:

$$|f(a + \alpha, b + \beta) - f(a, b)| < \epsilon. \tag{2}$$

The surrounding text contains the following part: "*[. . .] for every arbitrarily small positive constant ϵ [. . .]*". Basically, every preceding and following noun of an identifier (in this case ϵ) is a candidate for a definiens[5]. We were using the Stanford CoreNLP[6] to tag words. Unfortunately, the tagger tagged the word "*constant*" as an adjective rather than a noun, which leads to the wrong assumption that ϵ has no definiens in the surrounding text. While this example was an extreme case, the context analyzation process is usually time-consuming and mostly finds a lot of possible but not correct pairs of identifiers and definiens, i.e., a high recall[7] and a low precision[8] value. Note that a high recall value does not mean there would be no space for improvements, seen in the ϵ-example above.

3 PRELIMINARY RESEARCH

My preliminary research is structured in two parts. The research starts with my Master´s thesis, which I describe in the following subsection 3.1. The second part focused on analyzing existing tools, creating a comprehensive gold standard and develope improvement techniques, explained in subsection 3.2.

3.1 Translation from semantic LaTeX to CAS

I completed my Master´s thesis at the TU Berlin in Germany, NIST, USA, and Maplesoft, Canada. I focused my research on the translation between special function and orthogonal polynomial LaTeX

[4] https://dlmf.nist.gov/1.5#E2
[5] Latin for a term that describes another term (in our case symbols or identifiers).
[6] https://stanfordnlp.github.io/CoreNLP/
[7] Fraction of relevant instances that have been retrieved from the sources over the total amount of relevant instances in the sources.
[8] Fraction of relevant instances among the retrieved instances.

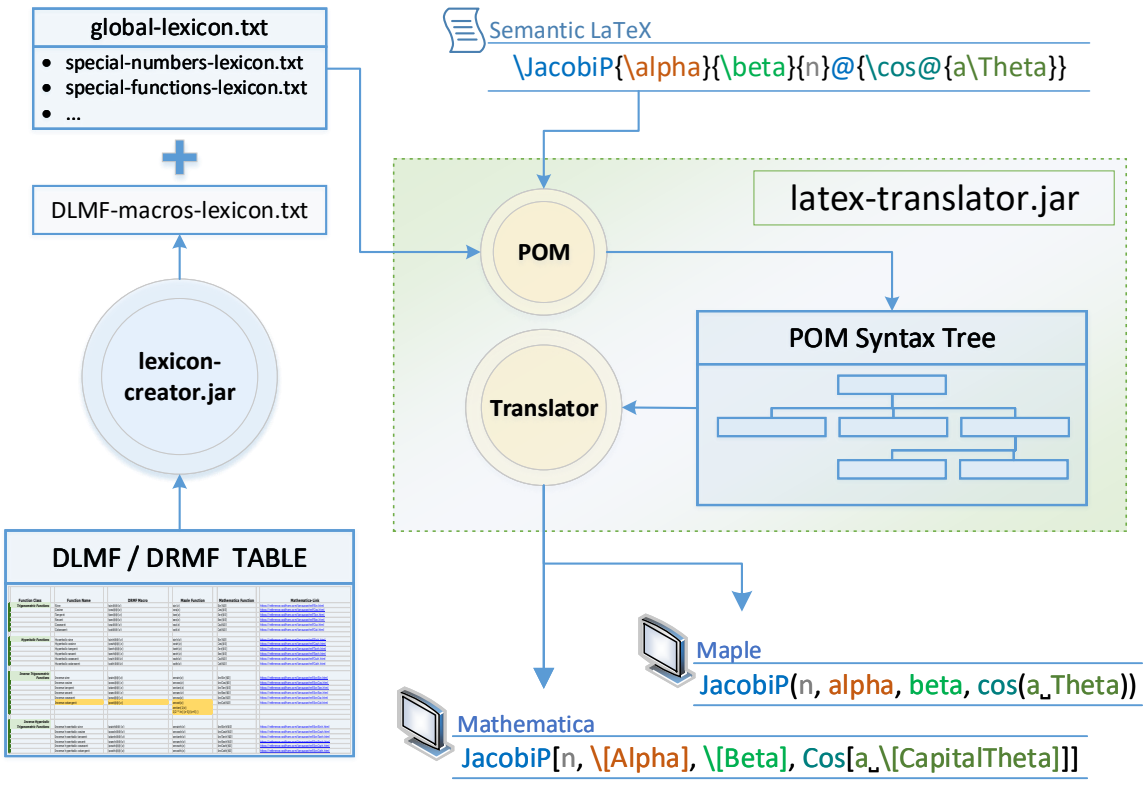

Figure 2: Flow diagram, which explains the translation process between semantic LaTeX and a CAS. The POM-Tagger parses a LaTeX expression based on the lexicon files and creates a syntax tree representation. That syntax tree can be translated node by node to a CAS representation.

expressions with DLMF/DRMF macros and representations in the Maple and Mathematica CAS. The thesis was also published in a conference paper [4].

From a scientific point of view, mathematics in LaTeX is primarily a formal language with an alphabet, words, and rules. LaTeX mathematical expressions can be described by a context-free grammar, to build a syntactical structure. We used the Part-of-Math tagger (POM-Tagger) [25], to build a syntax tree of a mathematical expression written in LaTeX. With a syntax tree, we can translate an expression, node by node without changing the meaning of the expression.

Figure 2 describes the translation process for the Jacobi polynomial example in table 1. The POM-Tagger uses the lexicon files to tag each token in the input expression with additional information. These lexicon files store that information, that describes the meaning of each symbol. I extended the lexicon files to provide translation patterns for the semantic DLMF/DRMF macros. Thereby, the POM-Tagger provides translation patterns for known semantic macros and also semantic but ambiguous information about other symbols. For instance, in the lexicon file, the symbol `\beta` indicates that it is a Greek letter and it is often used to describe the second of three angles in a triangle, or in case of special functions, it could be the Dirichlet beta function, and so on and so forth. In our example from table 1, it is just a variable. Picking the right meaning is essential and one of the goals of my doctoral thesis.

For the Master´s thesis, we only allowed disambiguated expressions with DLMF/DRMF macros, to avoid all of the problems previously described. We also followed the intuitive approach and presumed that the semantic information is already provided. Consequently, an author has to provide this information during the writing process. In my Master´s thesis, I attempted to provide mathematical expression representations with sufficient semantic information to provide a lossless translation to a CAS. Therefore, we refer to a LaTeX expression which uses DLMF/DRMF macros as semantic LaTeX.

3.2 Extract semantics from generic LaTeX

While the POM-Tagger follows a lexicon-based approach to tag tokens with semantic information, other approaches already try to analyze the context, i.e., the surrounding text, of the formula [1, 15, 19, 22]. The Mathematical Language Processor (MLP) extracts definiens and identifier pairs based on context analyzation with NLP algorithms.

During my first stay at NII, I researched on existing tools with the capability of parsing generic LaTeX expressions, such as the POM-Tagger. We have chosen the content MathML (cMML) as the output format because the DLMF/DRMF macros are still not publicly available and thereby not supported by any other tools than the POM-Tagger and LaTeXML, which was originally developed to create the semantic version of the DLMF [11]. We developed a comprehensive gold standard of 300 randomly picked formulae from Wikipedia, the DLMF sources and the NTCIR 12 task [15, 26]. This gold standard contains manually annotated semantic information in manually corrected cMML files, such that the cMMLs can be considered as correct and most accurate for the 300 formulae in the gold standard. We compared the MML files generated from available tools with the cMML files in the gold standard to measure similarities and accuracy in extracting the identifiers and their meanings in the formula. Furthermore, we used the MLP to analyze the context of each formula from the gold standard. As a test state, we implemented some post-processing algorithms to improve the generated MML files with results from the MLP analyzation. For example, when the MLP recognizes a symbol as a function based on the context, we automatically adjust the generated MML based on this conclusion, which significantly improves the similarity and accuracy [21].

I will continue to research on the MLP pipeline and will combine the results from the context analyzation approach with the lexicon-based approach from the POM-Tagger. During our research, we miss a comprehensive API to handle MathML files effectively. Thereby, we are planning to provide such an API for others researchers that they do not need to reinvent the wheel and can follow our already beaten track. The next section will give a brief overview of the general objectives of my doctoral research.

4 THESIS OBJECTIVES

4.1 Research Goal & Research Objectives

The goal of my doctoral research is to:

Accomplish and evaluate an approach for automatic extraction of sufficient semantic information for calculations of mathematical expressions in scientific publications and web pages.

In respect to this, we call semantic information of a mathematical expression sufficient when a translation to a CAS becomes feasible. To achieve this goal, I define the following thesis objectives:

I. Analyze the strengths and weaknesses of existing semantification approaches of mathematical expressions.
II. Develop a new semantification concept that will improve the current approaches based on the identified weaknesses.
III. Implement the system for an automatic semantification of mathematical expressions in real-world scientific documents.
IV. Implement an extension of the system to provide translations to computer algebra systems.
V. Evaluate the developed system by comparing the performance and accuracy, i.e., recall and precision values, with existing approaches and take advantage of evaluation techniques for CAS translations [4].

In the following, I will give a brief overview of the already discovered problems of existing approaches and explain my new semantification concept.

4.2 The Problems

As already mentioned, an automatic semantification process has performance weaknesses and complicated to realize, because the semantics are rarely explicitly given in the expression. Consequently, one has to analyze the context of the expression to find and extract semantic information. While those extraction approaches usually already have high recall values, they still can be improved by improving the used NLP techniques, seen in the ϵ-example from section 2. A possible improvement might take advantage of dictionary-based approaches. For example, consider a "*constant*" as a noun in mathematical scientific publications. This would lead us to specialized NLP algorithms for scientific publications. Another problem is, that existing approaches also ignore the structure of mathematical expressions, which also contains useful information to conclude semantic information. Integrating these techniques as a preprocessing step and use specialized NLP algorithms would improve performances, recall and precision values. The following subsection will present my new approach that improves existing tools in the explained way.

4.3 Multiple-Scan Approach

Consider a dictionary, that contains information about the meanings of each symbol and the structure of the formula in respect of this meaning. The previously described POM-tagger project contains such a dictionary. Assuming the correct semantic information is given in that dictionary, we need to eliminate other possible meanings to conclude the correct information. My approach will take advantage of the existing tools and combine them with the mentioned dictionary-based approach. The concept will be split into the following three research objectives:

(1) narrow down possible meanings only from the expression itself, without referring to the context of the expression (dictionary-based),
(2) refine the process with conclusions from the nearby context of the expressions (dictionary-based preprocessing with context-based refinements), and
(3) improve the previous process by analyzing not only the nearby context but the overall topic of the whole scientific paper or book, its references and other publications by the authors (further expanded context-based refinements).

The behavior of a human reader inspires the idea behind the concept of the combination of context- and dictionary-based approaches shown in objectives (1), (2) and (3). The necessary information to get correct and essential semantics is given somewhere in the context and in the formula itself. If not, it would be difficult even for a scientific reader to understand the given expression. The concept is illustrated in figure 3. In the following, I will explain each objective in a more detailed way.

Since we will achieve these three objectives by scanning the environment multiple times, we call this approach the *multiple-scan approach*. Objective (1) concentrates on the expression itself,

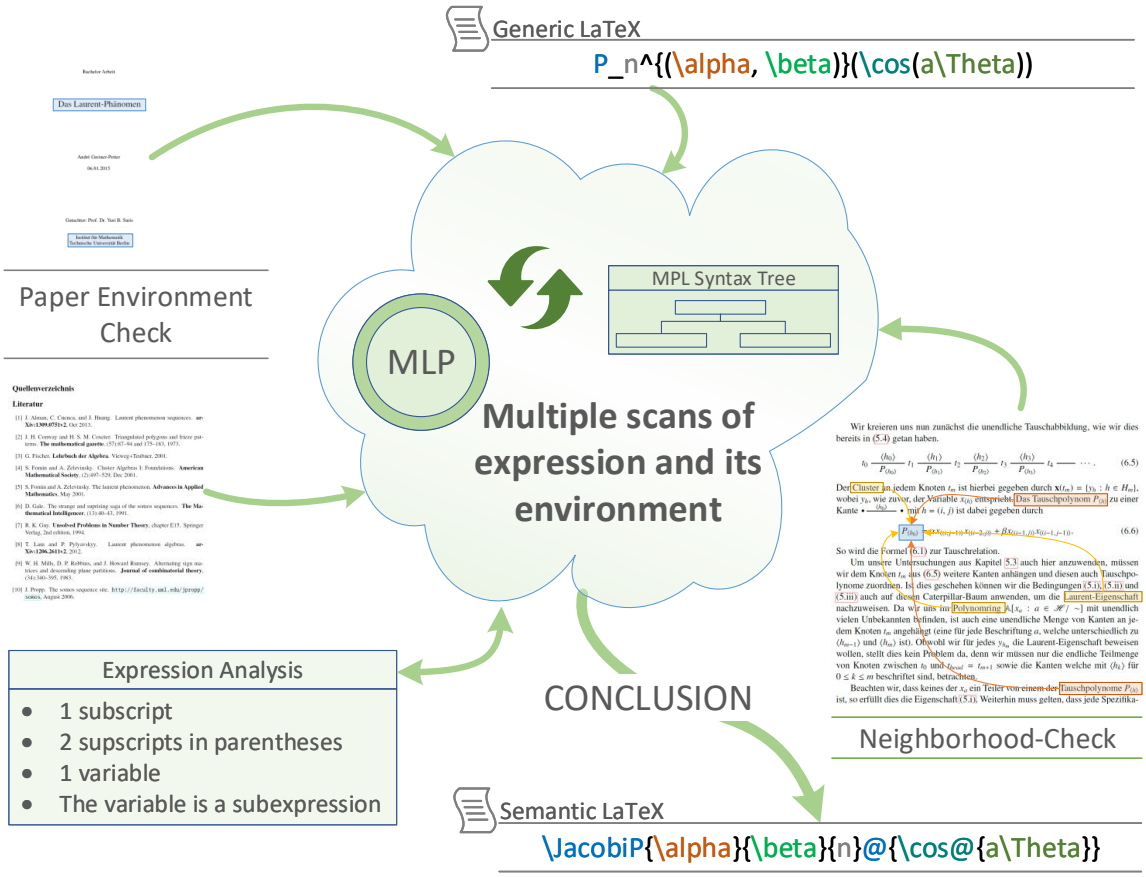

Figure 3: Conceptual explanation of the multiple scan approach.

without extracting information from the context. My proposed approach is to exploit the coherence between the structure of given formula and its meaning, constructing a Markov logic network to deduce possible semantic meanings. Therefore, each meaning gets a probability. If the highest probability is below a given threshold, it would be necessary to use (2) and (3) for improving these probabilities. Otherwise, the probability is sufficiently high for concluding semantic information.

Consider the Jacobi polynomial expression in table 1. The given expression has a superscript, a subscript and the following expression in parentheses. A leading expression in letters with the following expression in parentheses may lead us to the conclusion that the leading expression is the name of a function and the expression in parentheses is its argument. That is also what Maple does, even when the name of the function is unknown. Additionally, the first symbol P has a superscript and a subscript. Note that the Meixner-Pollaczek[9] polynomial $P_n^{(\lambda)}(x;\phi)$ and the associated Legendre[10] function of the first kind $P_\nu^\mu(x)$ are also referenced with P and all of these functions has a superscript, a subscript and an argument. But the Jacobi polynomial assumes a superscript of two parameters, while the Meixner-Pollaczek polynomial and the Legendre function just assume one parameter in the superscript.

Objective (2) based on the finding that around 70 percent of the symbolic elements in scientific papers are denoted in the surrounding text [23]. The mentioned NLP approaches try to exploit this finding to retrieve the semantic information for symbols in a formula. In my thesis, I will primarily work with the MLP approach described in [19, 22]. This approach extracts the symbols from the formula (called identifiers) and retrieves nouns from the surrounding text as candidates for definiens of the identifier. The scoring process assumes that the chance for a correct combination of identifier and definiens depends on the distance between the identifier and its definiens and the distance of the identifier to the closest formula that contains this identifier. I strongly believe we can improve the scoring process with the conclusions from my first objective above.

If the correct semantic information is still unsure, objective (3) is the last way to find a solution. Online compendia, such as arXiv, can be used to discover the overall topic of a scientific paper, the

[9] Meixner-Pollaczek polynomial: $P_n^{(\lambda)}(x;\phi)$, http://dlmf.nist.gov/18.19#E6
[10] Associated Legendre Function of the First Kind: $P_\nu^\mu(x)$, http://dlmf.nist.gov/14.3#E6

references and the area of research of the authors. The MCAT search engine developed by Kristianto, Topic, and Aizawa [15, 18] has performed such and can extract and score information from the document at the document granularity level. I will try to add this engine to our software to solve objective (3).

To achieve the objective (1), we will extend the POM-Tagger and its lexicon files. That is already part of the planned (and currently work in progress) development of the POM-Tagger. Therefore, I will focus on objectives (2) and (3), and support the progress of the POM-Tagger collaboratively with the DRMF project team.

5 PLANNED RESEARCH

5.1 Short Term Project Proposal

Wikipedia as a highly frequently used lexicon has over 17 million edits every month[11]. During the last two years (since May 2016 until January 2018) 7 million different formulae have been rendered via Mathoid in Wikipedia, i.e., there were 7 million edits during the last two years just in mathematical expressions[12]. Wikipedia uses TEX-Markup since 2003 to write and edit mathematical expressions. Therefore, the Wikipedia word processor is a highly suitable test environment to add machine learning algorithms.

The planned recommender system will be embedded to the Wikipedia word processor to learn and train supervised by each editor who modifies or write mathematical expressions in a Wikipedia article. The planned service would work as the usual word processor, that gets additional live updated recommended semantic versions for his mathematical LATEX input. An editor will be able to accept, refuse or just ignore the recommended semantic version for his current input. Based on his acceptance or refusal, the algorithm will adjust the scores for the recommended presentations.

Since every semantic macro has a generic LATEX definition, the backward translation dataset can be used for first training of the algorithm. Furthermore, the entire DLMF website is implemented in semantic LATEX and thus provide an eminently suitable test and training dataset.

5.2 Long Term Plans

As mentioned in the previous subsection, I plan to realize a machine learning algorithm integrated in Wikipedia. I estimate six months (Aug. 2018 - Jan. 2019) for developing the proposed project and additional three months (Feb. 2019 - Apr. 2019) for reviewing and evaluate the results. The results from this project will then be combined with the results from the previously developed and optimized MLP pipeline and the results from the first objective, which will be achieved by the POM-Tagger project.

The following task will be combining the results from the improved MLP project and the enhanced POM-Tagger project and develop an overall combination algorithm that takes advantage of both systems. This phase will finalize objectives (1) and (2) and realize a dictionary-based preprocessing system with context-based enhancements. I estimate three months for realizing this task (May 2019 - Jul. 2019).

With the help of the previously acquired findings, I will investigate the possibility of deducing semantic information from the overall topic of the context of an input expression. I am planning to use NLP approaches and online compendia of scientific papers such as arXiv to explore the topic by the authors and references of the scientific document. Accomplishing objective (3) will give us the field of a mathematical expression, so that we can fine-tune our scoring of objectives (1) and (2). I plan to dedicate six months (Aug. 2019 - Jan. 2020) to this task.

Writing the dissertation for a further six months (Jan. 2020 - Jun. 2020) will complete my doctoral research project.

[11] https://stats.wikimedia.org/v2/#/all-projects (visited Feb. 2018)
[12] https://github.com/physikerwelt/wikiMath17 (visited Feb. 2018)

REFERENCES

[1] A. Author(s). Improving Mathematical Identifier Definition Extraction using Machine Learning. National Institute of Informatics. 2018.

[2] H. S. Cohl et al. Digital Repository of Mathematical Formulae. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2014, pp. 419–422. DOI: 10.1007/978-3-319-08434-3_30.

[3] H. S. Cohl et al. Growing the Digital Repository of Mathematical Formulae with Generic LaTex Sources. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 280–287. DOI: 10.1007/978-3-319-20615-8_18.

[4] H. S. Cohl et al. Semantic Preserving Bijective Mappings of Mathematical Formulae Between Document Preparation Systems and Computer Algebra Systems. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2017, pp. 115–131. DOI: 10.1007/978-3-319-62075-6_9.

[5] R. M. Corless et al. "According to Abramowitz and Stegun" or arccoth needn't be uncouth. In: *ACM SIGSAM Bulletin* 34.2 (June 2000), pp. 58–65. DOI: 10.1145/362001.362023.

[6] C. David, C. Lange, and F. Rabe. Interactive Documents as Interfaces to Computer Algebra Systems: JOBAD and Wolfram|Alpha. In: *Intelligent Computer Mathematics, 10th International Conference, AISC 2010, 17th Symposium, Calculemus 2010, and 9th International Conference, MKM 2010, Paris, France, July 5-10, 2010. Proceedings*. Ed. by S. Autexier et al. Vol. 6167. Lecture Notes in Computer Science. Springer, 2010. DOI: 10.1007/978-3-642-14128-7.

[7] M. England et al. Branch cuts in maple 17. In: *ACM Comm. Computer Algebra* 48.1/2 (2014), pp. 24–27. DOI: 10.1145/2644288.2644293.

[8] A. Gaudeul. Do Open Source Developers Respond to Competition?: The (La)TeX Case Study. In: *SSRN Electronic Journal* (2006). DOI: 10.2139/ssrn.908946.

[9] *Generating and Importing TeX in Mathematica*. https://reference.wolfram.com/language/tutorial/GeneratingAndImportingTeX.html. visited Feb. 2018.

[10] J. Giceva, C. Lange, and F. Rabe. Integrating Web Services into Active Mathematical Documents. 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. 279–293. DOI: 10.1007/978-3-642-02614-0_24.

[11] D. Ginev et al. The LaTeXML Daemon: Editable Math on the Collaborative Web. In: *Intelligent Computer Mathematics - 18th Symposium, Calculemus 2011, and 10th International Conference, MKM 2011, Bertinoro, Italy, July 18-23, 2011. Proceedings*. Ed. by J. H. Davenport et al. Vol. 6824. Lecture Notes in Computer Science. Springer, 2011, pp. 292–294. DOI: 10.1007/978-3-642-22673-1_25.

[12] *Input and Output: Translate LaTeX in Mapel*. https://www.maplesoft.com/support/help/Maple/view.aspx?path=latex. visited Feb. 2018.

[13] N. Kando, T. Sakai, and M. Sanderson, eds. *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016*. National Institute of Informatics (NII), 2016.

[14] D. E. Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms (3rd Edition)*. Addison-Wesley Professional, 1997.

[15] G. Y. Kristianto, G. Topic, and A. Aizawa. MCAT Math Retrieval System for NTCIR-12 MathIR Task. In: *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016*. Ed. by N. Kando, T. Sakai, and M. Sanderson. National Institute of Informatics (NII), 2016.

[16] N. Meuschke et al. Analyzing Mathematical Content to Detect Academic Plagiarism. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*. ACM Press, 2017. DOI: 10.1145/3132847.3133144.

[17] *NIST Digital Library of Mathematical Functions*. http://dlmf.nist.gov/, Release 1.0.17 of 2017-12-22. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds.

[18] S. Ohashi et al. Efficient Algorithm for Math Formula Semantic Search. In: *IEICE Transactions* 99-D.4 (2016), pp. 979–988.

[19] R. Pagel and M. Schubotz. Mathematical Language Processing Project. In: *Joint Proceedings of the MathUI, OpenMath and ThEdu Workshops and Work in Progress track at CICM co-located with Conferences on Intelligent Computer Mathematics (CICM 2014), Coimbra, Portugal, July 7-11, 2014*. Ed. by M. England et al. Vol. 1186. CEUR Workshop Proceedings. CEUR-WS.org, 2014.

[20] M. Schubotz. Augmenting Mathematical Formulae for More Effective Querying & Efficient Presentation. PhD thesis. Technical University of Berlin, 2017.

[21] M. Schubotz et al. Improving the Representation and Conversion of Mathematical Formulae by Considering their Textual Context. Accepted at Joint Conference on Digital Libraries (JCDL) 2018. Mar. 2018.

[22] 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 2016, Pisa, Italy, July 17-21, 2016*. Ed. by R. Perego et al. ACM, 2016, pp. 135–144. DOI: 10.1145/2911451.2911503.

[23] M. Wolska and M. Grigore. Symbol Declarations in Mathematical Writing. In: *Towards a Digital Mathematics Library. Paris, France, July 7-8th, 2010*. 2010, pp. 119–127.

[24] World Wide Web Consortium (W3C). *Content Markup Language (MathML)*. https://www.w3.org/TR/MathML3/Overview.html. visited Feb. 2018.

[25] A. Youssef. Part-of-Math Tagging and Applications. In: *Intelligent Computer Mathematics - 10th International Conference, CICM 2017, Edinburgh, UK, July 17-21, 2017, Proceedings*. Ed. by H. Geuvers et al. Vol. 10383. Lecture Notes in Computer Science. Springer, 2017, pp. 356–374. DOI: 10.1007/978-3-319-62075-6_25.

[26] R. Zanibbi et al. NTCIR-12 MathIR Task Overview. In: *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016*. Ed. by N. Kando, T. Sakai, and M. Sanderson. National Institute of Informatics (NII), 2016.