Towards Grounding of Formulae

Takuto Asakura[1], **André Greiner-Petter**[2], **Akiko Aizawa**[3], **Yusuke Miyao**[4]
The University of Tokyo[1,4], University of Wuppertal[2], National Institute of Informatics[3]
{takuto,yusuke}@is.s.u-tokyo.ac.jp[1,4], andre.greiner-petter@zbmath.org[2],
aizawa@nii.ac.jp[3]

Abstract

A large amount of scientific knowledge is represented within mixed forms of natural language texts and mathematical formulae. Therefore, a collaboration of natural language processing and formula analyses, so-called mathematical language processing, is necessary to enable computers to understand and retrieve information from the documents. However, as we will show in this project, a mathematical notation can change its meaning even within the scope of a single paragraph. This flexibility makes it difficult to extract the exact meaning of a mathematical formula. In this project, we will propose a new task direction for grounding mathematical formulae. Particularly, we are addressing the widespread misconception of various research projects in mathematical information retrieval, which presume that mathematical notations have a fixed meaning within a single document. We manually annotated a long scientific paper to illustrate the task concept. Our high inter-annotator agreement shows that the task is well understood for humans. Our results indicate that it is worthwhile to grow the techniques for the proposed task to contribute to the further progress of mathematical language processing.

1 Introduction

In modern research, scientific progress is often solely shared in digital form. Especially in technical research fields, such as in Science, Technology, Engineering, and Mathematics (STEM), it is a crucial aspect to access data and new results in a quick and uniform way. Nevertheless, mathematical formulae, which transport essential information in scientific documents, often remain semantically unutilized in large databases such as Digital Library of Mathematical Functions (DLMF)[1] (Lozier, 2003) and the pre-print archive arXiv.org[2] (hereafter referred to as arXiv). By applying Math Information Retrieval (MathIR) techniques based on natural language processing (NLP), we are able to utilize this extra knowledge of mathematical formulae to build scientific knowledge bases (KBs) (Koprucki and Tabelow, 2016), improve mathematical search engines (Aizawa et al., 2013; Davila and Zanibbi, 2017; Ohashi et al., 2016), or even convert entire scientific papers into executable formats (Kohlhase and Iancu, 2014).

Formulae often express key ideas in scientific documents. Consequently, working with STEM documents requires to grasp the meaning and intention of the respective formulae. In other words, grounding of formulae is crucial for processing STEM documents and developing its applications. However, the grounding is not a trivial process because of the flexibility of mathematical notation and the impreciseness of natural languages. First, generally, formulae in documents are not independent content that can be understood separately from surrounding texts. For this reason, some initiative projects, e.g., the mathematical language processing (MLP) project (Pagel and Schubotz, 2014), the Mathcat project (Kristianto et al., 2014), and the Part-of-Math (POM) tagger (Youssef, 2017), have been undertaken to integrate NLP techniques into formula analysis. We also follow this MLP direction. Second, there is a necessity of disambiguation of mathematical notation because a letter or symbol in formulae is not used in a constant single meaning in a document (Greiner-Petter et al., 2020a,b). The usage of notation is highly-flexible and, as we will show in this paper, a notation can be used for several meanings even in a paragraph. This notation flexibility is not a problem for tasks that targeting short fragments of text, e.g., the ARQMath task (Mansouri et al., 2020; Zanibbi et al., 2020) for question posts from a question answering website. However, it is necessary to perform the grounding in consider-

[1] https://dlmf.nist.gov/
[2] https://arxiv.org/

Proceedings of the First Workshop on Scholarly Document Processing, pages 138–147

https://doi.org/10.18653/v1/P17

ation of the flexibility for processing longer STEM literature.

It is difficult to perform such a grounding in the scopes of existing tasks or MLP tools because many of the tools and approaches are not capable of carrying multiple meanings for a single symbol in a document (Greiner-Petter et al., 2020b; Kristianto et al., 2014; Krstovski and Blei, 2018; Yasunaga and Lafferty, 2019; Schubotz et al., 2016). Therefore, we will propose a new task direction of *grounding of formulae* (Figure 1) in order to take the flexibility into account. In short, the grounding is procedures to identify smallest groups of letters and symbols in formulae, i.e., *math words*, that independently refer to a mathematical concept and associate the math words with a corresponding text description or an entry in an external KB. For example in Figure 1, the first and the third y are parts of math words $\mathrm{y}(\cdot)$ (where $\cdot$ represents an arbitrary argument) and associated with a description of a function while the second y is an independent math word and associated with a description for an output vector. In our grounding, instead of directly associating each math word to a text description, we put an intermediate procedure: making groups each of which consists of math words referring to an identical mathematical concept. For instance in Figure 1, the first and the third y belong to a group because they both refer to the same function, while the second one is in another group because it refers to a vector. It is notable that the grounding is similar to some established tasks, namely coreference resolution (Sukthanker et al., 2020) and named entity recognition (Bunescu and Pasca, 2006).

In this work, we checked the feasibility of the proposing task direction for the grounding. For this purpose, we made a long annotated scientific paper in which all formulae are annotated with math word spans and text descriptions of the corresponding mathematical concepts. The math words in the annotated paper which refer to the same mathematical concept are tied together in a group. We did the annotation for an entire paper rather than small fragments of texts to disclose the flexibility of math word usage. Through the analysis of our annotated paper, we revealed that the meanings of math words can be changed even within the scope of a single paragraph in actual STEM literature. In addition, we did the annotation by multiple human annotators and calculated the inner-annotator agreements so that to confirm that our task design can be well-understood, at least for human beings, and can be performed without individual differences.

2 Related Work

A few tasks that are similar to our grounding of formulae have been proposed in the community of MathIR. The NTCIR project run several shared tasks in the past (Aizawa et al., 2013, 2014; Zanibbi et al., 2016). The task designs of these shared tasks focuses on applications, namely searching and question-answering, while our grounding focuses rather basic parts and is regarded as preprocess for numerous MLP tasks including formulae searching, question-answering, and conversions to formal languages. Notably, the NTCIR-10 MathIR Test Collection (Aizawa et al., 2013), which is a specific dataset for their shared task, contains textual descriptions for formulae components, and thus the data are similar to our annotated paper. The dataset includes 10 papers chosen from the arXMLiv dataset using a manually annotated description for each formula element. For instance, in the dataset, a formula $\log(x)$ is annotated with a description like "a function that computes the natural logarithm of the value x". Though their purpose is close to ours, we annotated not only descriptions but also a few pieces of additional information, i.e., affix types and group information (what concept the word refer to). In the terms of linguistics, these two can be regarded respectively as word spans and coreference information. Additionally, we did the annotation for a longer document than their target papers with coherency. We were especially interested in longer documents so that we can analyze how diverse meanings of mathematical concepts can be.

The *variable typing* task (Stathopoulos et al., 2018) is also closely relevant to our goal. Their task is simply associating *mathematical type* (technical terms referring to mathematical concepts) to each variable in STEM documents. For example, for a sentence

> Let P be a parabolic subgroup of $\mathrm{GL}(n)$ with Levi decomposition $P = MN$, where N is the unipotent radical. (Stathopoulos et al., 2018)

they assigned the "parabolic subgroup" and "unipotent radical" respectively to variables P and N as their mathematical types. Based on arXMLiv, they introduced their own dataset, which includes 33,524 labeled variables in 7,803 sentences. Their work re-

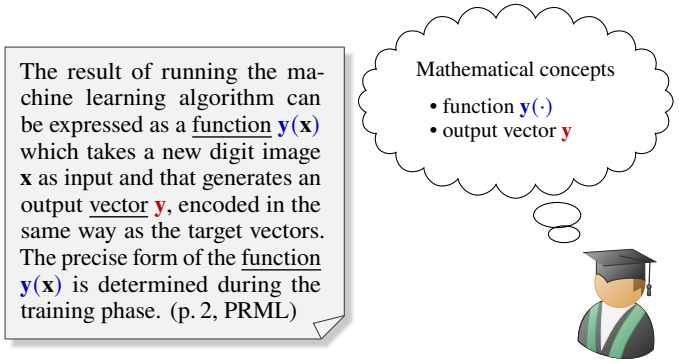

Figure 1: The *grounding of formulae*. It is a 3-step procedure: (1) detecting the span of math words—$y(\cdot)$ is a word for the first and third y, and the sole y is a word for the second y in the above quote (where $\cdot$ can be arbitrary argument), (2) making groups based on the corresponding mathematical description—the first and third y are elements of a group because they both refer to the same function, while the second one is an element of another group because it refers to a vector, (3) associating each group to text description or external knowledge.

sembles ours, but with three major differences. First, they annotated only mathematical types, which are partial components of mathematical concepts. Second, they targeted only those variables appearing as a single token in natural language texts, but we annotated all identifiers including those appearing in complex formulae. Third, they randomly selected sentences in the documents in arXMLiv to create their dataset. They did not attempt to annotate all variables in a full paper.

The POM Tagger (Youssef, 2017) is an initiative for token-level analyses of formulae in documents. The tagger plays the role of formulae akin to Part-of-Speech (POS) taggers in NLP. The tagger is intended to function with multiple scans. The function for the first scan was already implemented. The first scan recognizes lexical math terms (e.g., indexes, functions, left-delimiter, etc.) in formulae. For this tagging process, Youssef built a KB with more than 2,800 entries of symbols (tokens), each of which is associated with typical usage (role and category) and additional information of several kinds, such as the mathematical domain in which the term is used. In the planned following scans (second and third), the tagger will perform disambiguations of various types and extract further semantic information using NLP techniques. Our work is expected to be useful to implement and improve those features.

3 Grounding of Formulae

In order to describe the grounding of formulae precisely, we introduce a few linguistic terms for formulae, roughly borrowing from morphology in natural languages (Nida, 1949):

- A *math morpheme* (also known as *token*) is the smallest unit in formulae. In terms of Presentation MathML (Ausbrooks et al., 2014), this corresponds to an element, i.e., a tag. A morpheme can be a single letter or symbol (e.g., x, θ, $\prime$, $\times$, $=$, and $\sum$) or strings consisting of a few letters (e.g., log and argmax). All characters (both letters and symbols) that appear in formulae must belong to a math morpheme.

- A *math word* is a minimal group of morphemes that refer to a mathematical concept independently. Math words consist of one or more morphemes, typically one or a few morphemes. For instance, x, x', $\stackrel{\text{def}}{=}$, and $\log(\cdot)$ are words. Every word has a *base* morpheme, which reflects a core meaning of the word, and optionally has one or more *affixes*. That is to say, in a word x', x is the base morpheme; $\prime$ is a suffix, which is a type of affix.

Every math word has a corresponding mathematical concept such as the sign function, the set of all natural numbers, and (real) intervals. In the actual data and applications, the math words are associated with textual descriptions. Those descriptions can be taken either from the surrounding text of the formulae or from an external KB. Though some combinations of words, notably an entire formula, also refer to a mathematical concept, we stick to the scope of math words for the grounding. The process of combining the math words and interpreting the constituted concepts will be a subsequent task to our grounding. It is also notable that the concept of math word is close to mathematical objects of interests (MOIs) (Greiner-Petter et al., 2020a), sub

expressions in formulae that can be identified as 'important' components, but not exactly the same. While MOIs can contain other MOIs, math words are *minimal* groups of morphemes that can refer to mathematical concepts, so they naturally cannot contain other math words.

With the terms we introduced in the above, we can describe the proposing grounding of formulae as the following 3-step procedure that simulates the processes of understanding formulae by human beings (Figure 1):

1. Identifying spans of *math words*, a minimal group of math morphemes that refer to a mathematical concept. This step is necessary because the morphemes are not always used singly. For instance, in a text "A variable $\hat{x}$", two morphemes x and $\hat{}$ do not independently refer to mathematical concepts but refer to a variable as a group.

2. Grouping the math words based on the corresponding mathematical descriptions. As we mentioned, a notation can be used in several meanings in the scope of a single paragraph. With this step, we will obtain the groups each of which consists of math words appearances used in the same meaning. In the example of Figure 1, the first and the third y belongs to a group, each math word of which refers to a function, while the second y is in another group, each of which refers to a vector.

3. Associating the groups with text descriptions or external knowledge. It will be easier to associate all the groups to text descriptions in the same document, but in the actual literature, notations can be used without any description. For instance, π is often used for Archimedes' constant without explicit description. Thus, external knowledge will be required in addition to text descriptions in each document.

Although the grounding is a fundamental step for MLP, the processes, even merely recognizing the word spans, are not easy for computers because of various linguistic phenomena in formulae. The following paragraphs show the two most important phenomena: both make the grounding highly challenging. In short, because of these phenomena, the grounding demands disambiguation of math words. For disambiguation, integration of NLP and analyses for formulae are inescapable. More phenomena are discussed in detail elsewhere in the literature (Ganesalingam, 2013; Kohlhase and Iancu, 2014). All quotes presented in this section are from a textbook *Pattern Recognition and Machine Learning* (PRML) (Bishop, 2006).

Integration of Formulae and Texts Formulae in scientific documents are generally deeply integrated into narrative texts and inseparable from natural languages. For instance, in the following sentence, the equation is the passive subject in the grammar of English.

> For the case of a single real-value variable x, the Gaussian distribution is defined by
>
> $$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$
>
> which is governed by two parameters: μ, called the *mean*, and σ^2, called the *variance*. (p. 24, PRML)

From the viewpoint of the formula, meanings of some math words (x, μ, and σ^2) are described in the surrounding natural language texts. Moreover, natural language texts or their fragments can appear in formulae, as shown in the following:

> Given this definition of likelihood, we can state Beyes' theorem in words
>
> $$\text{posterior} \propto \text{likelihood} \times \text{prior}$$
>
> where all of these quantities are viewed as functions of $\mathbf{w}$. (p. 22, PRML)

The natural language parts of the above are just nouns, but they can be sentences, e.g., $\{n \in \mathbb{N} \mid n \text{ is even}\}$.

Because of this integration phenomenon, the grounding cannot be done merely by analyzing formulae. For faithful grounding, one must look into natural language texts to observe information such as the contexts to which formulae belong and to definitions, descriptions, or assumptions for identifiers in formulae.

Ambiguity Various ambiguities arise in formulae. Token-level analyses are responsible for disambiguation (Wang et al., 2016; Youssef, 2017). Different from natural languages, the meanings of words in formulae invariably depend on the context. It is common in STEM documents that a notation has multiple meanings even in a single document. For example, in Figure 1, a letter $\mathbf{y}$ is used in two meanings. Herein, in the first appearance, $\mathbf{y}$ is a function, but it is a vector in the second appearance.

We human beings can see that **y** is used in different meanings in these two appearances by the apposition nouns (in this example, the words "function" and "vector" immediately before the formulae) and the usages of the notations. In the first formula, the morpheme **y** is not used alone, but with affixes; it constitutes a word $\mathbf{y}(\cdot)$. However, the second **y** is used as a single word with no affixes. There are also syntactic ambiguities in formulae. For instance, a formula $f(a+b)$ can be interpreted in two ways: it means $f \times (a+b)$ if f is a variable, or applying the value of $a+b$ to a function f. Ambiguities of these types can be resolved by disambiguating the meanings of math words (the meanings of f and the parentheses in this case).

4 Manual Annotation for the Grounding

We performed manual annotation for the proposed grounding task so that to show the feasibility of the task with sufficient reproducibility. Moreover, language resources are indispensable to automate the task to contribute to the development of MLP tools. The annotated document for this work should play an initiative role to develop larger language resources for developing and evaluating the grounding technology. Collections of natural languages and corresponding formal expressions are not enough for this purpose, because it does not directly provide token-level information for each formula in natural language texts. It is still difficult to extract such information from simple parallel translations.

In order to reveal the flexibility of the math word usage, we annotated an entire long paper rather than small fragments of multiple texts. As we described, the grounding can be regarded as a three-step process. Corresponding to the first two steps of the grounding, the reference data should also have two types of information for each math morpheme in documents: (1) which word the morpheme belongs to and (2) which mathematical concept the word is referring to. We annotated these two pieces of information coherently for all formulae in a document. We also annotated text description by an annotator for each group, that is the information corresponding to the result of Step 3, but it is not literally extracted from the paper itself nor from external knowledge bases. These textual description are rather for the convenience for the annotators.

The classification-based annotation made it possible to evaluate the quality of annotated data. Moreover, the annotation is expected to be beneficial for numerous future applications. First, by studying the appearance pattern of mathematical concepts corresponding to formula words in a document, one can obtain linguistic statistics for formulae. For instance, it can help us to infer the form of the *scopes* for variables in documents. Secondly, the annotated information for each mathematical concept group can be extended easily in accordance with applications. We annotated referential descriptions for each math word as well, but these descriptions can be improved at any point.

Table 1: Basic statistics of the paper (Simeone, 2018).

#words in texts	10,616	#<mi> tags	937
#sections	7	#inline math	331
#pages (in PDF)	20	#display math	23

4.1 Targets

We took the original XHTML documents from the arXMLiv:08.2018 dataset (Ginev, 2018). The XHTML documents in the dataset were generated automatically by converting LaTeX document sources of scientific papers from arXiv with LaTeXML[3] (Ginev et al., 2011; Miller, 2018). In the dataset, we selected a paper *A Very Brief Introduction to Machine Learning With Applications to Communication Systems*[4] (Simeone, 2018) for our annotation because it has suitable numbers of words and formulae. This paper works with the topic with which the authors are familiar. In addition, the paper is easy to read and includes a reasonable number of formulae (Table 1). We annotated the group information to all identifiers in the paper. The analysis for the annotation are described in Section 5.

As the first attempt for performing such an annotation, we narrowed down a target to identifiers, which are one type of math morpheme. An identifier is a letter (e.g., x, **y**, and θ) or a string consisting of a few letters (e.g., sin and log) commonly representing variables, functions, and constants in formulae. We chose identifiers as targets because they are the most major class in standard formulae. In Presentation MathML, an identifier is placed in a <mi> tag, where `mi` stands for "math identifier".

4.2 Annotation Procedure

Before manual annotation, we preprocessed the target XHTML. First, several inappropriate MathML

[3] https://dlmf.nist.gov/LaTeXML/
[4] https://arxiv.org/abs/1808.02342

markups that were originally from unsuitable LaTeX markups by the author of the target paper were fixed to the right markups. This fixing was achieved by simple rule-based replacements, e.g., replacing `<mtext>E</mtext>` (`\text{E}` in the original LaTeX source by the author) to `<mi>E</mi>` (which corresponds to `\mathrm{E}` in LaTeX). Since our targets are only the `<mi>` tags for this time, this replacement was useful to annotate all the tokens that should be grounding manually. For the target document, we defined seven replacement rules. Secondly, lists of two types for annotation were generated by extracting information from the target XHTML file. The first one is a list of identifiers. It had entries for all letters and strings (case-sensitive and typeface-sensitive) with blank description fields. This list played the role of a *dictionary*. In other words, it was an extremely detailed "index to notations" (Table 2). The second list was a simple list of all identifiers' appearances in the document: the list of ids for all `<mi>` tags in the XHTML file. We took this dictionary-based approach to clearly show the groups of math words which are used in the same meaning.

The annotation process was done by manually modifying these two lists. When reading the target paper, the annotator added items to the entries when an identifier is defined or used in a new meaning. As presented in Table 2, each item was given a few fields: a description for the identifier usage, types of affixes in the corresponding words. Then, the annotator associates each identifier's appearance to the corresponding item in the dictionary within the list of identifiers (Figure 2). This task was accomplished efficiently with a GUI application we developed. We associate all identifiers in the document to the items even if the identifier morphemes appeared in a word as an affix.

Our language resource and all programs developed for this project are available with annotation via our repository[5].

5 Analysis on the Annotated Paper

5.1 Agreements and Mismatch Analyses

To verify our annotations, three persons annotated the same target article independently, and the agreements were calculated. First, Annotator 1 performed the whole process of the annotation (Section 4.2). The annotator created both a dictionary and the annotation file, which is a list of identifier appearances associated with the corresponding items in the dictionary. Then the dictionary, that includes all possible mathematical concepts that can be referred to in the paper, was sent to the other two annotators (Annotator 2 is a coauthor of this work. Annotator 3 is not). They performed the step of annotation that associates each identifier's appearance to a dictionary item. They needed about a day to complete the annotation. We shared the common dictionary this time for the ease of annotation work, but all annotators should create their own dictionaries in future work. Table 3 presents results of our experiment. With the given dictionary, the inter-annotator agreements were 96.48% (between Annotator 1 and Annotator 2) and 87.94% (between Annotator 1 and Annotator 3). Of 937 appearances, 132 (14.09%) are identifiers, each of which has a single candidate item. These are included in these agreements. In addition, mismatches of affix types between annotators are important because the numbers of such affix type mismatches reflect disagreements on the *math word spans*. A few examples are explained below. Therefore, we also counted affix mismatches. The numbers are presented in the second column of Table 3.

The two affix type mismatches by Annotator 2 were simply mistakes. The other 31 mismatches were all attributable to a single disagreement on the mathematical concept for $\mathcal{D}$ in the document. In the target paper, the identifier $\mathcal{D}$ refers training datasets for the learning tasks they are discussing, but the assumptions for the dataset (e.g., whether or not the data points follow a true distribution) differ among sections. For example, in §3.1, $\mathcal{D}$ is introduced for the first time as:

> we are given a training set $\mathcal{D}$ of N training points (x_n, t_n), with $n = 1, \ldots, N$, where the variables x_n are the inputs (Simeone, 2018)

Moreover, some times, there is no clear declaration about the assumption, e.g.,

> Under this assumption, the data set $\mathcal{D}$ is not necessary, ... (Simeone, 2018)

and it engenders disagreement. Because the meanings of some identifiers depend on the meanings of others, mismatches might have cascading effects. Results show that we obtained 31 mismatches from a single disagreement of the referring to mathematical concept of a math word.

The agreement of Annotator 3 was lower than

[5] https://sigmathling.kwarc.info/resources/grounding-dataset/

Table 2: Excerpt from the dictionary. In the actual dictionary file, all identifiers appear in the descriptions are also associated to the corresponding items in the same dictionary.

Identifier	Description	Affixes
t (*italic*)	an output of a regression or classification problem in general	(NONE)
	an output of a regression problem, generated by $p(x, t)$	(NONE)
	n-th output in the training set $\mathcal{D}$	subscript
	a predicator which takes an input x and return a predicated value	over, parentheses
	⋮	
t (roman)	a random variable for a test output for regression problem	(NONE)
	⋮	

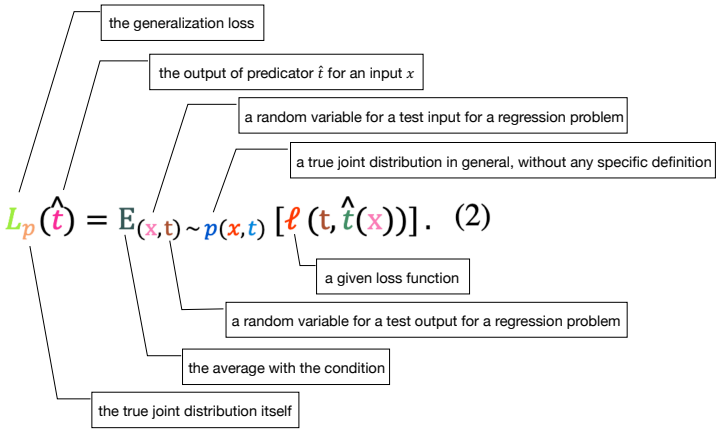

Figure 2: Example of annotated mathematical concepts in a bit complex formula. Due to the space limitation, only the descriptions for some of the identifiers are shown, but all of the identifiers are annotated with mathematical concepts in the actual data. The same letter in the same color is associated with the same mathematical concept.

Table 3: Performances of the annotators. The second column shows the inter-annotator agreements (compared to the golden data created by Annotator 1). The third column shows the number and ratio of identifiers annotated with an item that has different affix types (patterns) out of all disagreements.

	Agreements	Affixes mismatches
Annotator 2	904/937 (96.48%)	2/33 (6.06%)
Annotator 3	824/937 (87.94%)	60/113 (53.10%)

that of Annotator 2. Closer examination of the 113 mismatches reveals many duplications of the mismatch patterns. For example, the annotator marked a word $p(\cdot \mid \cdot, \cdot)$, which refers to "a parameterized predictive distributions" in the correct annotation as $p(\cdot \mid \cdot)$, which refers to "a parameterized true distribution" 19 times. By excluding such duplications, we found that the 113 mismatches can be categorized into 40 patterns, of which 25 patterns were affix type mismatches. Most of them can be distinguished easily by their appearance (e.g., annotating $p(\cdot \mid \cdot, \cdot)$ as $p(\cdot \mid \cdot)$). Apparently, many of these cases are mistakes or are the result of a misunderstanding of the concept of the affix types for Annotator 3. In the remaining 15 patterns, 10 are exactly the same mismatches made by Annotator 2. This finding indicates that choosing the most suitable mathematical description as an identifier $\mathcal{D}$ was the most difficult for the document.

5.2 Analyses on the Annotation and Notable Phenomena in the Target Document

The dictionary we created for the target document consists of 104 items within 40 entries. We counted items for each entry (identifier) in the dictionary (Figure 3). Herein, 18 entries out of 40 have two or more items. This finding indicates that about half of the identifiers in the documents have ambiguities on their meanings and the readers must disambiguate to perform the grounding of formulae. The entry with the greatest number of items in the document was t in regular font. Concretely, it has 13 meanings in the single document (see Table 2).

Figure 4 portrays a plot of the positions of identifier appearances and annotated items in the dictionary. Biases are readily apparent in the plot. The trends of referred mathematical concepts, which are sort of *scopes*, differ from section to section. For

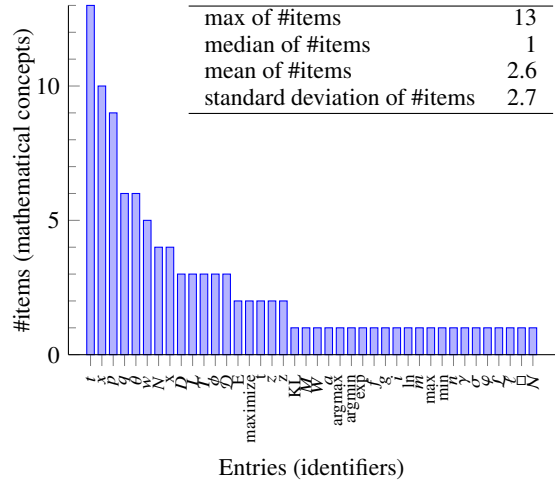

Figure 3: Number of items for each entry in the dictionary. In the paper, identifier t, x, p are used for 13, 10, 9 meanings respectively.

example, the scope of identifier x changed clearly at the beginnings of §3.2, §3.5, §3.6, and §5.1 in the target document. Moreover, some identifiers are used in the same meaning independent of such trends. The scope for t cannot be seen so clearly compared to x. The mathematical concepts referred by identifier $\mathcal{D}$, which is the most arguable one in the document, switch back and forth several times. Overall, as we mentioned, the usage of a notation is not constant in the paper but in fact so flexible that the meaning can change even in a single paragraph.

Incidentally, the target document includes several noteworthy sentences (Simeone, 2018). In the beginning of the article, the author of the paper states the following:

> Throughout, we use Roman font to denote random variables and the corresponding letter in regular font for realizations. (Simeone, 2018)

This is a meta-declaration about the font usage in formulae throughout the article. The annotators had to keep this declaration in mind to distinguish differences between variables in Roman font and in regular font.

6 Future Work

We made a long annotated paper and show that the flexibility of the mathematical notation is high in actual STEM literature. Moreover, we could check the feasibility of our task direction of the grounding of formulae. The number of annotated papers for the grounding needs to be increased because a single paper is naturally biased. However, the entirely manual annotation costs too much to enhance the size of the resource in the same way. Therefore, we will work on partial automation of the process first. With the combination of the partial automatic method of the grounding and manual annotation by humans, we will be able to efficiently enlarge the resource. Furthermore, we will develop an entirely automated grounding method, including the third step, i.e., the part of associating the groups with text descriptions or external knowledge, for various MLP applications.

Acknowledgments

This work was supported by the Japan Science and Technology Agency (JST CREST, Grant JPMJCR1513) and the German Research Foundation (DFG, Grant GI-1259-1).

References

Akiko Aizawa, Michael Kohlhase, and Iadh Ounis. 2013. NTCIR-10 math pilot task overview.

Akiko Aizawa, Michael Kohlhase, and Iadh Ounis. 2014. NTCIR-11 math-2 task overview. page 11.

Ron Ausbrooks, Stephen Buswell, David Carlisle, Giorgi Chavchanidze, Stéphane Dalmas, Stan Devitt, Angel Diaz, Sam Dooley, Roger Hunter, Patrick Ion, and Michael Kohlhase. 2014. *Mathematical Markup Language (MathML) 3.0 Specification*.

Christopher M Bishop. 2006. *Pattern Recognition and Machine Learning*.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*.

Kenny Davila and Richard Zanibbi. 2017. Layout and semantics: Combining representations for mathematical formula search. In *SIGIR 2017*.

Mohan Ganesalingam. 2013. *The Language of Mathematics: A Linguistic and Philosophical Investigation*.

Deyan Ginev. 2018. arxmliv:08.2018 dataset, an html5 conversion of arxiv.org. SIGMathLing.

Deyan Ginev, Heinrich Stamerjohanns, Bruce R. Miller, and Michael Kohlhase. 2011. The LaTeXML daemon: Editable math on the collaborative web. In *CICM 2011*.

André Greiner-Petter, Moritz Schubotz, Fabian Müller, Corinna Breitinger, Howard S. Cohl, Akiko Aizawa, and Bela Gipp. 2020a. Discovering mathematical objects of interest—a study of mathematical notations. In *WWW 2020*.

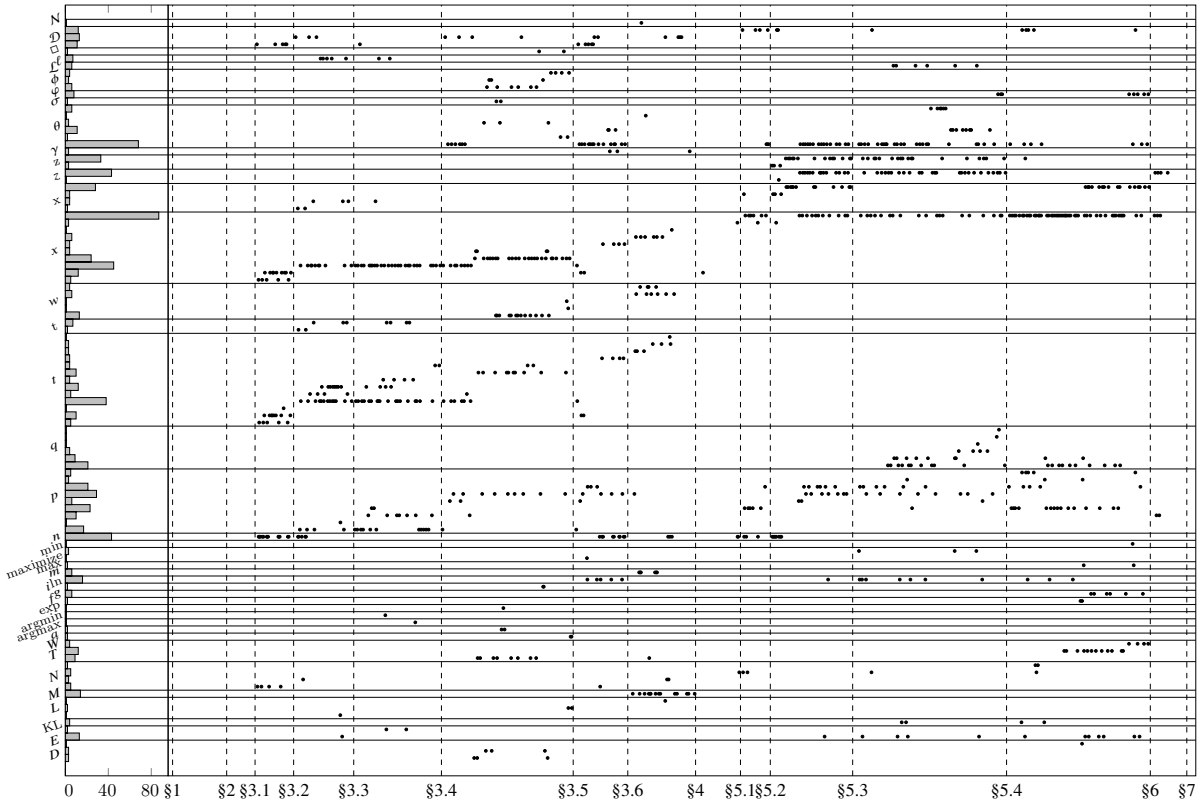

Figure 4: Math words appearances in the target document. The value of a coordinate on the horizontal axis is the position of the corresponding <mi> tag in the XHTML document. A notation can be used in several meanings in the document, and the concept that is referred to by each appearance is shown as the position in vertical axis. Between each separator, items are sorted by the positions of their first appearances.

André Greiner-Petter, Abdou Youssef, Terry Ruas, Bruce R. Miller, Moritz Schubotz, Akiko Aizawa, and Bela Gipp. 2020b. Math-word embeddings in math search and semantic extraction. *Scientometrics*.

Michael Kohlhase and Mihnea Iancu. 2014. Co-representing structure and meaning of mathematical documents.

Thomas Koprucki and Karsten Tabelow. 2016. Mathematical models: A research data category? In *Mathematical Software – ICMS 2016*, volume 9725, pages 423–428.

Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. 2014. Exploiting textual descriptions and dependency graph for searching mathematical expressions in scientific papers. In *Ninth International Conference on Digital Information Management (ICDIM 2014)*, pages 110–117.

Kriste Krstovski and David M. Blei. 2018. Equation embeddings. *arXiv*.

Daniel W. Lozier. 2003. Nist digital library of mathematical functions. *Annals of Mathematics and Artificial Intelligence*.

Behrooz Mansouri, Anurag Agarwal, Douglas Oard, and Richard Zanibbi. 2020. Finding old answers to new math questions: The arqmath lab at CLEF 2020. In *Advances in Information Retrieval*, volume 12036, pages 564–571.

Bruce Miller. 2018. *LaTeXML The Manual—A LaTeX to XML/HTML/MathML Converter, Version 0.8.3*.

Eugene A Nida. 1949. *Morphology: The descriptive analysis of words*.

Shunsuke Ohashi, Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. 2016. Efficient algorithm for math formula semantic search. *IEICE Transactions on Information and Systems*.

Robert Pagel and Moritz Schubotz. 2014. Mathematical language processing project. In *Joint Proceedings of the MathUI, OpenMath and ThEdu Workshops and Work in Progress track at CICM*.

Moritz Schubotz, Alexey Grigorev, Marcus Leich, Howard S. Cohl, Norman Meuschke, Bela Gipp, Abdou S. Youssef, and Volker Markl. 2016. 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*, pages 135–144.

Osvaldo Simeone. 2018. A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*.

Yiannos Stathopoulos, Simon Baker, Marek Rei, and Simone Teufel. 2018. Variable typing: Assigning meaning to variables in mathematical text. In *NAACL 2018*.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

Kai Wang, Xinfu Li, and Xuedong Tian. 2016. On ambiguity issues of converting latex mathematical formula to content mathml. In *Collaborative Computing: Networking, Applications, and Worksharing*.

Michihiro Yasunaga and John D. Lafferty. 2019. Topiceq: A joint topic and mathematical equation model for scientific texts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:7394–7401.

Abdou Youssef. 2017. Part-of-math tagging and applications. In *CICM 2017*.

Richard Zanibbi, Akiko Aizawa, and Michael Kohlhase. 2016. NTCIR-12 MathIR task overview. page 10.

Richard Zanibbi, Douglas W. Oard, Anurag Agarwal, and Behrooz Mansouri. 2020. Overview of arqmath 2020: Clef lab on answer retrieval for questions on math. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 169–193.