Why Machines Cannot Learn Mathematics, Yet

André Greiner-Petter[1], Terry Ruas[2], Moritz Schubotz[1],
Akiko Aizawa[3], William Grosky[2], Bela Gipp[1]

[1] University of Wuppertal, Wuppertal, Germany
{last}@uni-wuppertal.de
[2] University of Michigan-Dearborn, Dearborn, USA
{truas,wgrosky}@umich.edu
[3] National Institute of Informatics, Tokyo, Japan
aizawa@nii.ac.jp

Abstract. Nowadays, Machine Learning (ML) is seen as the universal solution to improve the effectiveness of information retrieval (IR) methods. However, while mathematics is a precise and accurate science, it is usually expressed by less accurate and imprecise descriptions. Generally, mathematical documents communicate their knowledge with an ambiguous, context-dependent, and non-formal language. In this work, we apply text embedding techniques to the arXiv collection of STEM documents and explore how these are unable to properly understand mathematics from that corpus, while proposing alternative to mitigate such situation.

Keywords: Mathematical Information Retrieval, Machine Learning, Word Embeddings, Math Embeddings, Mathematical Objects of Interest

1 Introduction

Mathematics is capable of explaining complex concepts and relations in a compact, precise, and accurate way. The general applicability of mathematics allows a certain level of ambiguity in its expressions. This ambiguity is regularly mitigated by short explanations following or preceding these mathematical expressions, that serve as context to the reader. Along with context dependency, inherent issues of linguistics (e.g. non-formality) make it even more challenging for computers to understand mathematical expressions. Said that, a system capable of capturing the semantics of mathematical expressions automatically would be suitable for several applications, from improving search engines to recommender systems. Consider for example the the lower bound for Van der Waerden numbers

$$W(2,k) > 2^k/k^\varepsilon. \tag{1}$$

Learning connections, such as between W and the entity '*Van der Waerden's number*' from above, requires a large specifically labeled scientific database that contains mathematical objects. Word embedding techniques has received significant attention over the last years in the Natural Language Processing (NLP)

community, especially after the publication of word2vec [13]. Recently, more and more projects try to adapt this knowledge for solving Mathematical Information Retrieval (MIR) tasks [3, 9, 26, 24]. In this paper, we explore some of the main aspects that we believe are necessary to leverage the learning of mathematics by computer systems. We explain, with our evaluations of word embedding techniques on the arXMLiv 2018 [4] dataset, why current ML approaches are not applicable for MIR tasks, yet.

Current MIR approaches [8, 22, 20] try to extract textual descriptors of the parts that compose mathematical equations. This leads to two main issues: (i) how to determine the parts which have their own descriptors, and (ii) how to identify correct descriptors over others. Answers to (i) are more concerned in choosing the correct definitions for which parts of a mathematical expression should be considered as one mathematical object [7, 25, 21]. Current definitions, such as the content MathML 3.0[4] specification, are often imprecise. Consider α_i, where α is a vector and α_i its i-th element. In this case, α_i should be considered as a composition of three content identifiers, each one carrying its own individualized semantic information, namely the vector α, the element α_i of the vector, and the index i. However, with the current specification, the definition of these identifiers would not be canonical. Because of these problems in current standards, nowadays work focusing their efforts on (ii).

Schubotz et al. [22] presented an approach for scoring pairs of identifiers and descriptors by the number of words between them. They made the assumption that correct definiens appear close to the identifier and to the complex mathematical expression that contains this same identifier. Kristianto et al. [8] introduce a ML approach, in which they train a Support Vector Machine (SVM) to consider sentence patterns and other characteristics as features (e.g. part-of-speech (POS) tags, parse trees). Later, [20] combine the aforementioned approaches and use pattern recognition based on the POS tags of common identifier-definiens pairs, the distance measurements, and SVM, reporting results for precision and recall of 48.60% and 28.06%, respectively. More recently, some projects try to use embedding techniques to learn patterns of the correlations between context and mathematics. In [3], they embed single mathematical symbols, while Krstovski and Blei [9] represent complex mathematical expressions as single unit tokens for IR. Recently, M. Yasunaga et al. [24] explore an embedding technique based on recurrent neural networks to improve topic models by considering mathematical expressions.

2 Machine Learning on Embeddings

The *word2vec* [13] technique computes real-valued vectors for words in a document using two main approaches: skip-gram and continuous bag-of-words (CBOW). Both produce a fixed length n-dimensional vector representation for each word in a corpus. In the skip-gram training model, one tries to predict the context of a

[4] `https://www.w3.org/TR/MathML3/`

given word, while CBOW predicts a target word given its context. In word2vec, context is defined as the adjacent neighboring words in a defined range, called a sliding window. The main idea is that the numerical vectors representing similar words should have close values if the words have similar context.

The lack of solid references and applications that provide the semantic structure of natural language for mathematical identifiers make their disambiguation process challenging. In natural texts, one can try to infer the most suitable word sense for a word based on the lemma[5] itself, the adjacent words, dictionaries, thesaurus and so on. However, in the mathematical arena, the scarcity of resources and the flexibility of redefining their identifiers take this issue to a more delicate scenario. The text preceding or following the mathematical equation is essential for its understanding.

More recently, [9] propose a variation of word embeddings for mathematical expressions. Their main idea relies on the construction of a distributed representation of equations, considering the word context vector of an observed word and its word-equation context window. They treat equations as single-unit words (EqEmb), which eventually appears in the context of different words. They also try to explore the effects of considering the elements of mathematical expressions separately (EqEmb-U). While they present some interesting findings for retrieving entire equations, little is said about the vectors representing equation units and how they are described in their model.

Nowadays mathematics in science is mostly either given in LaTeX or MathML. The former is used by humans for writing scientific documents. The latter, on the other hand, is popular in web representations of mathematics due to its machine readability and XML structure. There has been a major effort to automatically convert LaTeX expressions to MathML [21] ones. However, neither LaTeX nor MathML are practical formats for embeddings. Considering the equation embedding techniques in [9], we devise three main types of mathematical embeddings.

Mathematical Expressions as Single Tokens: EqEmb [9] uses entire mathematical expressions as one token. In this type, the inner structure of the mathematical expression is not taken into account. For example, Equation (1) is represented as one single token t_1. Any other expression, such as $W(2, k)$, in the surrounding text of (1), is an entirely independent token t_2, i.e. no relation between $W(2, k)$ and (1) can be trained.

Stream of Tokens: This approach represents mathematical expressions as a sequence of its inner tokens. This approach has the advantage of learning all mathematical tokens. However, complex mathematical expressions may lead to long chains of elements, which increases noise[6]. There are several approaches to reduce the noise, e.g. by only considering identifiers and operands [3] or by implementing a long short-term memory (LSTM) architecture that is capable of handling longer chains [24]. Later in this paper, we present a model based on

[5] canonical form, dictionary form, or citation form of a set of words

[6] Noise means, the data consists of many uninteresting tokens that affect the trained model negatively.

the stream of tokens approach. We will show that this approach is valuable to capture relations between mathematical expressions but not between expressions and their descriptors.

Semantic Groups of Tokens: The third approach of embedding mathematics is only theoretical, and concerns the aforementioned problems related to the vague definitions of identifiers and functions in a standardized format (e.g. MathML). As previously discussed, current MIR and ML approaches would benefit from a basic structural knowledge of mathematical expressions, such that variations of function calls (e.g. $W(r,k)$ and $W(2,k)$) can be recognized as the same function. Instead of defining a unified standard, current techniques use their own ad-hoc interpretations of structural connections, e.g., α_i is one identifier rather than three [21, 20]. We assume that an embedding technique would benefit from a system that is able to detect the parts of interest in mathematical expressions prior any training processes. However, such system still does not exist.

2.1 Performance of Math Embeddings

The examples illustrated in [3, 9, 24] seem to be feasible as a new approach for distance calculations between complex mathematical expressions. While comparing mathematical expressions is essentially practical for search engines or automatic plagiarism detection systems, these approaches do not seem to capture the components of complex structure separately, which are necessary for other applications, such as automated reasoning. Another aspect to be considered is that in [9] they do not train mathematical identifiers, preventing their system from learning connections between identifiers and definiens. Additionally, the connection between entire equations and definiens is, at some level, questionable. Entire equations are rarely explicitly named. However, in the extension EqEmb-U [9], they use a Syntax Layout Tree (SLT) [27] representation to tokenize mathematical equations and to obtain specific unit-vectors, which is similar to our *identifiers as tokens* approach.

In order to investigate the discussed approaches, we apply variations of a word2vec implementation to extract mathematical relations from the arXMLiv 2018 [4] dataset, an HTML collection of the arXiv.org preprint archive[7], which is used as our training corpus. We used the *no_problem* and *warning* subsets for training. There are other approaches that also produce word embeddings given a training corpus as an input, such as GloVe [15], fastText [1], ELMo [16], and USE [2]. The choice for word2vec is justified because of its general applicability and robustness in several NLP tasks [6, 5, 10, 11, 17, 19].

We replace all mathematical expressions by the sequence of the identifiers it contains, i.e., $W(2,k)$ is replaced by '$W\ k$'. Further, we remove all common English stopwords from the training corpus. Finally, we train a word2vec model (skip-gram) using the following hyperparameters configuration[8]: vector size of

[7] `https://arxiv.org/`

[8] Non mentioned hyperparameters are used with their default values as described in the Gensim API [18]

300 dimensions, a window size of 15, minimum word count of 10, and a negative sampling of $1E-5$. The trained model is able to partially incorporate semantics of mathematical identifiers. For instance, the closest 27 vectors, considering cosine simularity, to the mathematical identifier f are mathematical identifiers themselves and the fourth closest noun vector to f is $\boldsymbol{v}_{\text{function}}$. Inspired by the classic *king-queen* example, we explore which tokens perform best to model a known relation. Consider an approximation $\boldsymbol{v}_{\text{variable}} - \boldsymbol{v}_{\text{a}} \approx \boldsymbol{v} - \boldsymbol{v}_{\text{f}}$, where $\boldsymbol{v}_{\text{variable}}$ represents the word *variable*, $\boldsymbol{v}_{\text{a}}$ the identifier a, and $\boldsymbol{v}_{\text{f}}$ represents f. We are looking for $\boldsymbol{v}$ that fits best for the approximation. We call this measure the *semantic distance* to f with respect to a given relation between two vectors. We perform an extensive evaluation on the first 100 entries of the *MathML-Ben* benchmark [21]. We evaluate the average of the *semantic distances* with respect to the relations between $\boldsymbol{v}_{\text{variable}}$ and $\boldsymbol{v}_{\text{x}}$, $\boldsymbol{v}_{\text{variable}}$ and $\boldsymbol{v}_{\text{a}}$, and $\boldsymbol{v}_{\text{function}}$ and $\boldsymbol{v}_{\text{f}}$. In addition, we consider only results with a cosine similarity of 0.7 or above to maintain a minimum quality in our results. The overall results were poor with a precision of $p = .0023$ and a recall of $r = .052$. For the identifier W (Equation (1)), the evaluation presents four semantically close results: *functions*, *variables*, *form*, and the mathematical identifier q. Even though expected, the scale of the presented results are astonishing.

Based on the presented results, one can still argue that more settings should be explored (e.g. different embedding techniques, parameters) and different preprocessing steps adopted. Nevertheless, the overall results would not be improved to a point of being comparable to [20] findings, which report a precision of $p = 0.48$. The main reason for this is that, mathematics as a language is highly customizable. Many of the defined relations between mathematical concepts and their descriptors are only valid in a local scope. Consider, for example, an author that notes his algorithm by π. This does not change the general meaning of π, even though it effects the meaning in the scope of the article. Current ML approaches only learn patterns of most frequently used combinations, e.g., between f and *function*. Furthermore, we assume this is a general problem that different embedding techniques and tweaks of settings.

3 Make Math Machine Learnable

A case study [23] has shown that only 70% of mathematical symbols are explicitly declared in the context. Four reasons are causing an explicit declaration in the context: (a) a new mathematical symbol is defined, (b) a known notation is changed, (c) used symbols are present in other contexts and require specifications to be properly interpreted, or (d) authors declarations were redundant (e.g. for improving readability). We assume (d) is a rare scenario compared to (a-c), unless in educational literature. On the other hand, (d) is most valuable for learning algorithms.

In cases (b-c), used notations are ambiguous. To overcome this issue, it requires an extensive database that collects all semantic meanings for mathematical expressions. In [25], they propose the use of tags, similarly to the POS tags in

linguistics, but for tagging mathematical TEX tokens. As a result, a lexicon containing several meanings for a large set of mathematical symbols is developed. Such lexicons might enable the disambiguation approaches in linguistics (e.g. via WordNet [14]) to be used in mathematical embeddings in the near future.

Usually, research documents represent state-of-the-art findings containing new and unusual notations and lack of extensive explanations (e.g. due to page limitations). In contrast, educational books carefully and extensively explain new concepts, thus they are rich of cases (a) and (d). Matsuzaki et al. [12] presented some promising results to automatically pass Japanese university entrance exams. The system required several manual adjustments. It illustrates the potential of a well-structured digital mathematical library that distinguishes the different levels of progress in articles (e.g. introductions vs. state-of-the-art publications) for ML algorithms.

Another problem in recent publications, is the lack of standards for properly evaluating MIR algorithms, leading to several publications that present promising results without an extensive evaluation [3, 9, 24]. While ML algorithms in NLP benefit from available extensive training and testing datasets, ongoing discussions about interpretations of mathematical expressions [21], and imprecise standards thwarts research progress in MIR. A common standard for interpreting semantic structures of mathematics would help to overcome the issues of different evaluation techniques. Therefore, we introduce *Mathematical Objects of Interest* (MOI). The goal of MOIs is to combine the advantages of concepts (1-3) and propose a unified solution for interpreting mathematical expressions. We suggest MOIs as a recursive tree structure in which each node is an MOI itself. The current workaround of the problematic example of α_i as an element of the vector α in content MathML is vague and inappropriate for content specific tasks. As an MOI, this expression would contain three nodes, with α_i as the parent node of two leaves α and i. While it first seems non-intuitive that α, as the vector, is a child node of its own element, this structure is able to incorporate all three components of semantic information of the expression. Hence, an MOI structure should not be misinterpreted as a logical network explaining semantic connections between its elements, but as a highly flexible and lightweight structure for incorporating semantic information of mathematical expressions.

4 Conclusion and future Work

In this paper, we explored how text embedding techniques are unable to represent mathematical expressions adequately. After experimenting with popular mathematical representations in MIR, we expose fundamental problems that prevent ML algorithms from learning mathematics. We further presented some concepts for enabling ML algorithms to learn mathematical expressions.

Acknowledgments This work was supported by the German Research Foundation (DFG grant GI-1259-1). We thank H. Cohl who provided insights and expertise.

References

[1] P. Bojanowski et al. "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5 (2017).

[2] D. Cer et al. "Universal Sentence Encoder for English". In: *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by E. Blanco and W. Lu. Association for Computational Linguistics, 2018.

[3] L. Gao et al. "Preliminary Exploration of Formula Embedding for Mathematical Information Retrieval: can mathematical formulae be embedded like a natural language?" In: *CoRR* abs/1707.05154 (2017). arXiv: 1707.05154.

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

[5] I. Iacobacci et al. "Embeddings for Word Sense Disambiguation: An Evaluation Study". In: *Proc. 54th Annual Meeting of the Association for Computational Linguistics (ACL) Vol. 1, Berlin, Germany*. ACL, 2016.

[6] I. Iacobacci et al. "SensEmbed: Learning Sense Embeddings for Word and Relational Similarity". In: *Proc. 53rd Annual Meeting of the Association for Computational Linguistics (ACL) Vol. 1, Beijing, China*. ACL, 2015.

[7] M. Kohlhase. "Math Object Identifiers - Towards Research Data in Mathematics". In: *Lernen, Wissen, Daten, Analysen (LWDA) Conference Proceedings, Rostock, Germany, September 11-13, 2017*. Ed. by M. Leyer. Vol. 1917. CEUR-WS.org, 2017.

[8] G. Y. Kristianto et al. "Extracting Textual Descriptions of Mathematical Expressions in Scientific Papers". In: *D-Lib Magazine* 20.11/12 (2014).

[9] K. Krstovski and D. M. Blei. "Equation Embeddings". In: *CoRR* abs/1803.09123 (2018). arXiv: 1803.09123.

[10] J. Li and D. Jurafsky. "Do Multi-Sense Embeddings Improve Natural Language Understanding?" In: *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal*. Lisbon, Portugal: Association for Computational Linguistics, 2015.

[11] M. Mancini et al. "Embedding Words and Senses Together via Joint Knowledge-Enhanced Training". In: *Proc. 21st Conference on Computational Natural Language Learning (CoNLL), Vancouver, Canada*. Association for Computational Linguistics, 2017.

[12] T. Matsuzaki et al. "The Most Uncreative Examinee: A First Step toward Wide Coverage Natural Language Math Problem Solving". In: *Proc. Twenty-Eighth AAAI Conference on Artificial Intelligence, Québec City, Québec, Canada*. Ed. by C. E. Brodley and P. Stone. AAAI Press, 2014.

[13] T. Mikolov et al. "Distributed Representations of Words and Phrases and Their Compositionality". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Lake Tahoe, Nevada: Curran Associates Inc., 2013.

[14] G. A. Miller. "WordNet: A Lexical Database for English". In: *Commun. ACM* 38.11 (1995).

[15] J. Pennington et al. "Glove: Global Vectors for Word Representation." In: *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar*. Vol. 14. Association for Computational Linguistics, 2014.

[16] M. Peters et al. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, 2018.

[17] M. T. Pilehvar and N. Collier. "De-Conflated Semantic Representations". In: *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*. The Association for Computational Linguistics, 2016.

[18] R. Řehůřek and P. Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. `http://is.muni.cz/publication/884893/en`. Valletta, Malta: ELRA, May 2010.

[19] T. Ruas et al. "Multi-sense embeddings through a word wense disambiguation process". In: *Expert Systems With Applications* (2019). Pre-print.

[20] M. Schubotz et al. "Evaluating and Improving the Extraction of Mathematical Identifier Definitions". In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings*. Ed. by G. J. F. Jones et al. Vol. 10456. Springer, 2017.

[21] M. Schubotz et al. "Improving the Representation and Conversion of Mathematical Formulae by Considering their Textual Context". In: *Proc. AMC/IEEE JCDL*. Ed. by J. Chen et al. ACM, 2018.

[22] M. Schubotz et al. "Semantification of Identifiers in Mathematics for Better Math Information Retrieval". In: *Proc. AMC SIGIR*. Pisa, Italy: ACM, 2016.

[23] M. Wolska and M. Grigore. "Symbol Declarations in Mathematical Writing". In: *Towards a Digital Mathematics Library*. Ed. by P. Sojka. Paris, France: Masaryk University Press, 2010.

[24] M. Yasunaga and J. Lafferty. "TopicEq: A Joint Topic and Mathematical Equation Model for Scientific Texts". In: *CoRR* abs/1902.06034 (2019). arXiv: `1902.06034`.

[25] A. Youssef. "Part-of-Math Tagging and Applications". In: *Proc. CICM*. Ed. by H. Geuvers et al. Cham: Springer International Publishing, 2017.

[26] A. Youssef and B. R. Miller. "Deep Learning for Math Knowledge Processing". In: *Proc. CICM*. Ed. by F. Rabe et al. Vol. 11006. Springer, 2018.

[27] R. Zanibbi et al. "Multi-Stage Math Formula Search: Using Appearance-Based Similarity Metrics at Scale". In: *Proc. 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy*. Ed. by R. Perego et al. ACM, 2016.