Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 2175–2182
Marseille, 11–16 May 2020

Natural Language Premise Selection: Finding Supporting Statements for Mathematical Text

Deborah Ferreira and Andre Freitas
Department of Computer Science
University of Manchester
{deborah.ferreira, andre.freitas}@manchester.ac.uk

Abstract

Mathematical text is written using a combination of words and mathematical expressions. This combination, along with a specific way of structuring sentences makes it challenging for state-of-art NLP tools to understand and reason on top of mathematical discourse. In this work, we propose a new NLP task, the natural premise selection, which is used to retrieve supporting definitions and supporting propositions that are useful for generating an informal mathematical proof for a particular statement. We also make available a dataset, NL-PS, which can be used to evaluate different approaches for the natural premise selection task. Using different baselines, we demonstrate the underlying interpretation challenges associated with the task.

Keywords: mathematical text, mathematical language processing, mathematical text analysis

1. Introduction

Comprehending mathematical text requires evaluating the semantics of its mathematical structures (such as expressions) and connecting its internal components with the respective definitions or premises (Greiner-Petter et al., 2019).

State-of-the-art models for natural language processing, such as BERT (Devlin et al., 2019), have high scores for several tasks, such as entity recognition, textual entailment and machine translation, but they do not encode the intricate mathematical background knowledge needed to reason over mathematical discourse.

The language of mathematics is composed of a combination of words and symbols, where symbols follow a different set of rules and have a specific alphabet. Nonetheless, word and symbols are interdependent in the context of mathematical discourse. This phenomenon is exclusive to mathematical language, not found in any other natural, or artificial, language (Ganesalingam, 2013), providing a unique and challenging application for semantic evaluation and natural language processing.

Understanding mathematical discourse has been explored before as a Mathematical Knowledge Extraction task (Aizawa et al., 2014); however, several aspects of the mathematical discourse related to deeper and more granular reasoning over mathematical discourse has not yet been investigated. There is a lack of datasets in the literature needed for exploring and studying mathematical discourse and its associated interpretation and reasoning.

We propose the task of natural premise selection, inspired by the field of automated theorem processing. Premise selection appeared initially as a task of selecting a (useful) part of an extensive formal library in order to limit the search space for an *Automated Theorem Proving* (ATP) system, increasing the chance of finding a proof for a given conjecture (Blanchette et al., 2016). Premises considered relevant are the ones that ATPs use for the automatic deduction process of finding a proof for a conjecture. The premise selection task is defined as: Given a collection of premises P, an ATP system A with given resource limits, and a new conjecture c, predict those premises from P that will most likely lead to an automatically constructed proof of c by A (Irving et al., 2016).

Natural premise selection is based not on formally structured mathematics, but on human-generated mathematical text. It takes as input mathematical text, written in natural language and outputs relevant mathematical statements that could support a human in finding a proof for that mathematical text. The premises are composed by a set of supporting definitions and supporting propositions, that act as explanations for the proof process.

For example, the famous *Fermat's Little Theorem* (Warner, 1990) has different possible proofs, one of them using the *Euclid's Lemma*. In this example, Euclid's Lemma would be considered useful for a human trying to prove Fermat's Little Theorem; therefore, it is a premise for the conjecture that Fermat's Little Theorem presents.

In order to evaluate this task, we propose a new dataset: NL-PS (Natural Language - Premise Selection), using as a basis the human-curated website ProofWiki[1]. This dataset opens possibilities of applications not only for the premise selection task but also for evaluating semantic representations for mathematical discourse (including embeddings), textual entailment for mathematics and natural language inference in the context of mathematical texts.

The contributions of this paper can be summarised as follows:

- Proposal of a new NLP task: natural language premise selection.
- A novel dataset, NL-PS, to support the evaluation of premise selection methods using natural language corpora.
- Comparison of different baselines for the natural premise selection task.

2. Related Work

NLP has been applied before in the context of general Mathematics. Chaganty and Liang (2016) proposes a new

[1] https://proofwiki.org/wiki/Main_Page

task for semantic analysis, the task of perspective generation, i.e., generating description to numerical values using other values as reference. Huang et al. (2016) analyze different approaches to solve mathematical word problems and concludes that it is still an unsolved challenge.

Ganesalingam and Gowers (2017) propose a program that solves elementary mathematical problems, mainly in metric space theory, and presents solutions similar to the ones proposed by humans. The authors recognize that their system is operating at a disadvantage because human language involves several constraints that rule out many sound and effective tactics for generating proofs.

Wang et al. (2018) propose an approach to automatically formalize informal mathematics using statistical parsing methods and large-theory automated reasoning. The idea is to convert from an informal statement to a formal one, using Mizar as the output language. After the statement has been correctly translated, it can be checked using an automatic tool.

Naproche (Natural language Proof Checking) (Cramer et al., 2009) is a project focused on the development of a controlled natural language (CNL) for mathematical texts and adapting proof checking software to work with this language in order to check syntactical and mathematical correctness.

Zinn (2003) proposes proof representation structures to represent mathematical discourse using discourse representation theory and also proposes a prototype that could be used to automate the process of generating proofs.

Approaches for creating embeddings of mathematical text have applied variations of the Skip-gram model (Mikolov et al., 2013), extending it with a specific tokenization strategy for equations and mathematical terms. Most tokenization strategies will use the tree structure of an equation to define the target tokens and can range from considering the full equation (Krstovski and Blei, 2018) as a single token or decomposing its component expressions or at the individual symbol-level (Gao et al., 2017). Greiner-Petter et al. (2019) developed a skip-gram-based model using as a reference corpus a set of arXiv papers in HTML format using a term-level tokenization granularity. The authors found that the induced vector space did not produce meaningful semantic clusters.

Premise selection is an approach generally used for selecting useful premises to prove conjectures in Automated Theorem Proving (ATP) systems (Alama et al., 2014a). Irving et al. (2016) propose a neural architecture for premise selection using formal statements written in Mizar. Other authors have used machine learning approaches such as Kernel-based Learning (Alama et al., 2014b), k-NN algorithm (Gauthier and Kaliszyk, 2015) and Random Forests (Färber and Kaliszyk, 2015). However, the neural approaches previously presented (Irving et al., 2016) have obtained higher scores at the premise selection task.

3. Linguistic Considerations

In this section, we describe some of the linguistics features present in a mathematical corpus. Our aim is to examine its discourse in combination with natural language. The following definitions are not of mathematical objects since those already have established mathematical definitions; in this work, we are interested in how the different mathematical objects are presented inside the mathematical text.

Let $\boxed{a \in \mathbb{R}_{>0}}$ be a strictly positive real number.
Let $\boxed{f : \mathbb{R} \to \mathbb{R}}$ be the real function defined as: $f(x) = a^x$ where a^x denotes a to the power of x.
Then $\boxed{f}$ is convex.

Figure 1: Theorem with three definiendums and six definiens, where the content inside the boxes are definiendums and the underlined content are definiens.

Definition 1. A **mathematical expression** $\mathcal{M}$, in a mathematical text, is defined by a set $\Sigma = \{s_1, s_2, s_3, ..., s_n\}$ where $s_i \in S$, and S is the set of symbols present in a certain mathematical domain of discourse, such as variables, constants and functions. A variable, for example, is considered an expression.

Definition 2. An **equation** $\mathcal{E}$ is defined as a combination of $m_i, m_j \in \mathcal{M}$ and an (in)equality predicate $p \in \{>, <, \leq, \geq, \neq, =\}$.

Definition 3. A **mathematical statement** μ can be:

- A sequence of words (from the mathematical domain) or;
- A sequence of words and expressions and/or equations or;
- A sequence of only equations.

Definition 4. A **mathematical text** τ is a sequence $\{\mu_1, \mu_2, \mu_3, ..., \mu_n\}$ of mathematical statements.

In the mathematical text, words, expressions and equations, can be directly related through a relationship of *definiendum* and *definiens*, where an expression, the definiendum, is defined by a mathematical statement or part of a mathematical statement, the definiens is also used to determine the *set of values* and properties associated with an expression.

Definition 5. A **mathematical definiens** σ_{τ_i} is the set of tuples composed by e and the set of (part of) mathematical statements that declares and/or quantifies e in the mathematical text τ_i. Figure 1 presents an example where the *definiens* and the *definiendum* are highlighted. A definiendum can have more than one *definiens*, for example, the expression "$f : \mathbb{R} \to \mathbb{R}$" is declared by the equation "$f(x) = a^x$", and has the property "*real function*". Therefore:

$$\sigma_{\tau_{example}} = \{(\text{“}f : \mathbb{R} \to \mathbb{R}\text{”}, \text{“}f(x) = a^x\text{”}), (\text{“}f : \mathbb{R} \to \mathbb{R}\text{”}, \text{“real function”}), ...\}$$

Different mathematical texts can also be related, since mathematical knowledge is often incremental, where one element depends on others. For example, in Figure 1, in

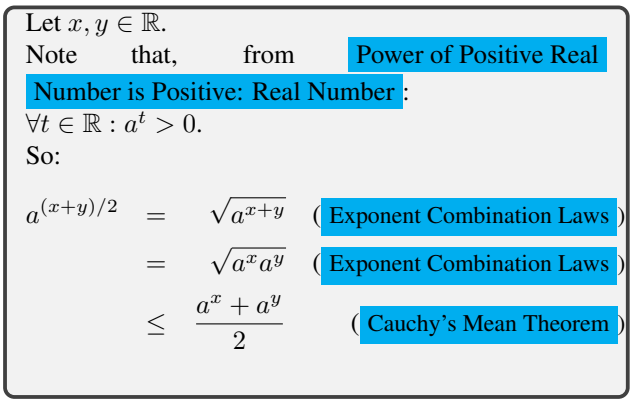

Figure 2: Example of part of a proof, where four mathematical supporting facts are present.

order to understand the meaning of the presented text, we need to understand the definition of a *real function*, which is defined in another mathematical text.

Definition 6. A **mathematical supporting definition** δ_{τ_i} is the set of mathematical texts $\{\tau_j, \tau_k, \tau_l, ...\}$, where all elements in δ_{τ_i} contains a definition of a concept presented in τ_i. For example, the theorem in Figure 1 is connected to the mathematical text that defines what is a real function.

Definition 7. A **definition** $\mathcal{D}$ is composed by a 4-tuple $(\tau, c, \sigma_\tau, \delta_\tau)$, where τ is the definition text, c is the set of categories that the definition belongs to, σ_τ is the set of definiens in the text and δ_τ is the set definitions that is referenced in $\mathcal{D}$. If δ_τ is empty, we call it an **atomic definition**.

A mathematical proof is a particular mathematical text that tries to convince the reader that a specific hypothesis can lead to a conclusion (Solow, 2002). Proofs often contain mathematical bindings. They can also be connected to other propositions, such as lemmas, theorems and corollaries, as we will define next.

Definition 8. A **mathematical supporting proposition** ω_{τ_i} is the set of propositions that helps support the argument proposed in the mathematical text τ_i of a proof. It is often used as an explanation for certain statements used for the construction of the proof. Figure 2 presents part of the proof, where the name of the supporting facts is highlighted. For example, the mathematical statement of Cauchy's Mean Theorem is a supporting fact for the proof shown.

Definition 9. The set of **premises** ϕ_{τ_i} of a mathematical text τ_i of a proof is the set of supporting facts ω_{τ_i} and the set of supporting definitions δ_{τ_i}, i.e., $\phi_{\tau_i} = \omega_{\tau_i} \cup \delta_{\tau_i}$.

Definition 10. A **mathematical proof** $\mathcal{P}$ is defined is composed by a tuple (τ, ϕ_τ), where τ is the proof text and ϕ_τ is the set of premises for $\mathcal{P}$.

Definition 11. A **theorem** $\mathcal{T}$ is composed by a tuple $(\tau, c, \sigma_\tau, \mathcal{P})$, where τ is the theorem's text, c is the set of categories that the theorem belongs to, σ_τ is the set of definiens in the text, P is the set of proofs for the theorem (one theorem can have more than one possible proof).

Definition 12. Similarly, we can define a **lemma** $\mathcal{L}$. $\mathcal{L}$ is composed by a 5-tuple $(\tau, c, \sigma_\tau, \mathcal{P}, t)$. With the addition of t, theorem where the lemma occurs.

Definition 13. A **corollary** $\mathcal{C}$ is composed by a 5-tuple $(\tau, c, \sigma_\tau, \mathcal{P}, t)$, where t is the theorem that derives $\mathcal{C}$.

4. Dataset Construction: NL-PS

In this section, we present our dataset, NL-PS, and detail the steps we took in order to construct it. Our dataset is available as a set of JSON files in http://github.com/debymf/nl-ps. A summary of the process is presented in Figure 3.

Parsing the corpus
The proposed dataset was extracted from the source code of ProofWiki. ProofWiki is an online compendium of mathematical proofs, with a goal to collect and classify mathematical proofs. ProofWiki contains links between theorems, definitions and axioms in the context of a mathematical proof, determining which dependencies are present. ProofWiki is manually curated by different collaborators; therefore, there are different styles of mathematical text and many elements cannot be extracted automatically.

Cleaning wiki tags
ProofWiki has wikimedia tags; however, ProofWiki has also specific tags related to the mathematical domain. Therefore, we cannot use default wiki extraction tools. A bespoke tool was developed to comply with ProofWiki's tagging scheme. For example, there is a particular tag for referring to another mathematical text, using passages from other texts in order to support a claim (Figure 4).

Proof curation
Several pages in ProofWiki are not directly related to mathematical propositions or definitions, such as users pages, help pages, and pages about specific talks. We manually analysed the pages and removed the ones that are not definitions, lemmas, theorems or corollaries. Some pages also contained tags to indicate that

Extraction of categories
ProofWiki has associated categories for each page. However, the categories are not harmonised across definitions and propositions. We merged different categories that belonged to the same mathematical branch and selected the categories that contained at least 100 different entries. The categories selected are: Analysis, Set Theory, Number Theory, Abstract Algebra, Topology, Algebra, Relation Theory, Mapping Theory, Real Analysis, Geometry, Metric Spaces, Linear Algebra, Complex Analysis, Applied Mathematics, Order Theory, Numbers, Physics, Group Theory, Ring Theory, Euclidean Geometry, Class Theory, Discrete Mathematics, Plane Geometry and Units of Measurement

Extracting supporting facts
The pages in ProofWiki are connected using hyperlinks. We leverage this structure to extract supporting propositions and supporting definitions. From the definition mathematical text, we extract the hyperlinks connecting to other definitions and these links are the supporting definitions.

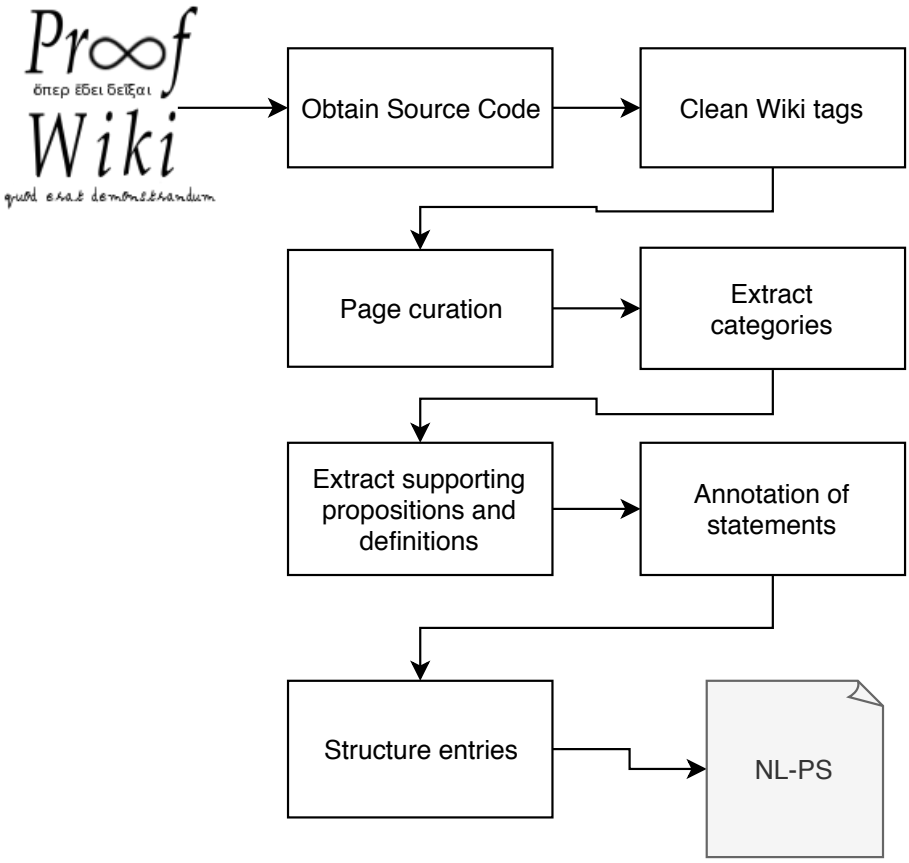

Figure 3: Pipeline used to build the NL-PS dataset.

Mathematical text 1

```
...
{{:Existence of Radius of Convergence of Complex
Power Series/Absolute Convergence}}
...
```

reference

Mathematical text 2

```
Let $\xi \in \C$.
Let $\displaystyle S \paren z = \sum_{n \mathop = 0}^\infty a_n \paren
{z - \xi}^n $ be a complex power series about $\xi$.
Let $R$ be the radius of convergence of $S \paren z$.

<onlyinclude>
Let $B_R \paren \xi$ denote the open $R$-ball of $\xi$.
Let $z \in B_R \paren \xi$.
Then $S \paren z$ converges absolutely.
If $R = +\infty$, we define $B_R \paren \xi = \C$.
</onlyinclude>
```

Figure 4: An example where Mathematical text 1 references a passage in Mathematical text 2 using the name of the passage to be referenced between curly brackets. Only the part highlighted is being referenced.

From the mathematical text of proofs, we extract the hyperlinks to other propositions and we consider these as supporting propositions. For example, Figure 5 presents a theorem and its respective proof. The proof contain links (highlighted) to other propositions, these are supporting propositions needed in order to support the proof.

Theorem

For every integer n such that $n > 1$, n can be expressed as the product of one or more primes, uniquely up to the order in which they appear.

Proof

In Integer is Expressible as Product of Primes it is proved that every integer n such that $n > 1$, n can be expressed as the product of one or more primes.

In Prime Decomposition of Integer is Unique, it is proved that this prime decomposition is unique up to the order of the factors.

Figure 5: Example of supporting propositions for a theorem.

Annotating mathematical text

The entries in ProofWiki are often divided in sections, for NL-PS, we are only interested in the sections that present a definition, a proposition or a proof. Proofs were curated (combining manual and automatic annotation) to contain only mathematic discourse, removing satellite discourse such as *Historical Notes*. Because some propositions can be proved in different ways, we also annotated the different proofs which can be found inside one single page.

Structuring the entries

Finally, the dataset is structured as follows:

- Definitions entries are composed by a mathematical

text and a set of supporting definitions.

- Lemmas and Theorems have a mathematical text, a proof and a set of premises.
- Corollaries are composed by a mathematical text, a proof, a set of premises and the theorem that derives the corollary.

5. Dataset Analysis

NL-PS has a total of 20,401 different entries, composed of definitions, lemmas, corollaries and theorems, as shown in Table 5..

Type	Number of entries
Definitions	5,633
Lemmas	327
Corollaries	292
Theorems	14,149
Total	20,401

Table 1: Types of mathematical documents in NL-PS

Figure 6 presents the distribution of different categories in the dataset.

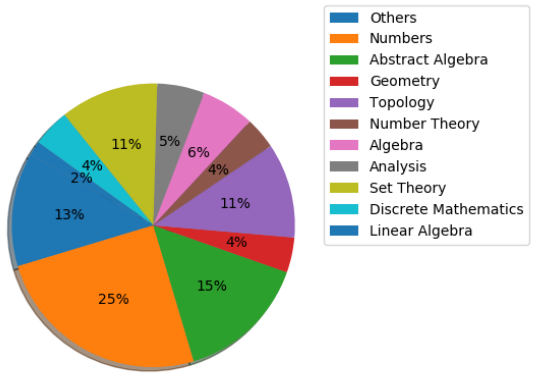

Figure 6: Distribution of documents per category in the dataset.

Figure 7 presents a histogram with the frequency of the different number of premises. We can observe that the statements usually have a small number of premises, with $8,046$ statements containing between one and five premises. The highest number of premises for one theorem is 32 (text for the theorem "The Sorgenfrey line is Lindelöf.").
Similarly, the histogram in Figure 8 shows the frequency of the different number of dependencies.
We also computed how many times each statement is used as a premise, and we observed that most of the statements are used as dependencies for only a small subset of premises. A total of $6,866$ statements has between one and three dependants. On average, statements contain a total of 289 symbols (characters and mathematical symbols). The

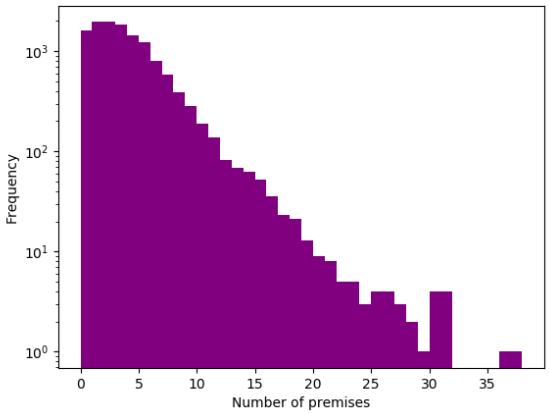

Figure 7: Distribution of the number of premises in the ProofWiki corpus.

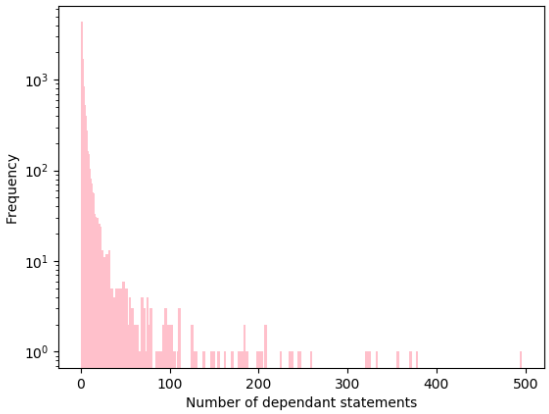

Figure 8: Number of times a statement is referred as a premise.

specific number of tokens will depend on the type of tokenisation used for the mathematical symbols.

We can also represent the connections (premises) between different mathematical texts as a graph. This graph has a total of 14,393 nodes (the number of nodes is smaller than the number entries, since some of the entries are disconnected, and we do not consider those for the graph) and 34,874 edges.

The dataset provides a specific semantic modelling challenge for natural language processing as it requires specific tokenization, co-reference resolution and the modelling of specific discourse structures tailored towards mathematical text. One crucial challenge is how to resolve the semantics of variables in mathematical expressions, which requires a particular binding method. As shown in Figure 9, variables that refer to the same set can often have different names. For example, in the definition of sine, the variable being used is x, but a and b refers to the same set. Basically, variables serve as a mathematical alternative to anaphora (Ganesalingam, 2013).

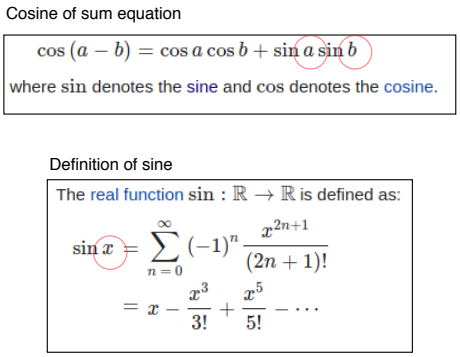

Figure 9: Variables with different symbols, but referring to the same set.

6. Experiments

In order to identify the challenges of the task of natural premise selection using NL-PS, we performed initial experiments using two baselines: TF-IDF and PV-DBOW (Le and Mikolov, 2014). We use both techniques to create vector representations for all the mathematical texts. Then compute the cosine similarity between each entry and rank the results by proximity. We then compute the Mean Average Precision (MAP) for each baseline, ranking all possible premises, computed as:

$$\text{MAP} = \frac{\sum_{i=1}^{N} \text{AveP}(\tau_i)}{N}$$

where N is the total number of documents, τ_i is the i-th mathematical text and AveP is the average precision.

Table 6. presents the initial results. We compare three different types of tokenisations for the mathematical elements. Initially, we treat the expressions and equations as single tokens; for example, the expression "$x + y + z$" would be considered a single word. We also considered tokenised expressions, tokenising operations and operators, the expression "$x+y+z$" would be tokenised as ['x','$+$','y','$+$','z']. Finally, we tokenise the whole text as a sequence of characters. We run PV-DBOW with the default parameters, comparing different sizes of embeddings, with the best results obtained with an embedding size of 100.

From these initial results, we can conclude that the task is semantically non-trivial and cannot be solved with retrieval strategies such as lexical overlap. We can also notice that we obtain better results when we tokenise the expressions, hinting that the elements inside the expressions have semantic properties that are relevant for determining the relevant premises. For the following experiments, we are using the tokenised expressions and PV-DBOW with an embedding size of 100.

In Table 6. we compare the results for different sizes of the dataset. We consider the full dataset and three different subsets with different categories. We can notice that for smaller datasets, both baselines perform better. This result was expected since with smaller datasets there are less possible premises, and elements from the same categories tend to be more uniform between themselves.

We can also consider the fact that the premises are transitive, i.e., if one a mathematical text τ_i has a premise x and a mathematical text τ_j has τ_i as a premise, then x should also be a premise of τ_j. In this case, the task becomes even more challenging, as we present in Table 6., where we consider the transitivity with two and three hops of distance. From the results, we notice that the more hops needed to obtain the premise, the worse our baselines perform.

	TFIDF	PV-DBOW		
		50	100	200
Expression as words	0.073	0.048	0.051	0.046
Tokenised expressions	**0.089**	**0.069**	**0.073**	**0.072**
Char level	0.051	0.059	0.065	0.061

Table 2: MAP results for TF-IDF and PV-DBOW comparing tokenisation of expressions. We compare the results for PV-DBOW for different dimension values.

	TFIDF	PV-DBOW
All Categories	0.089	0.076
Algebra (1,241)	0.183	0.177
Analysis (1,102)	0.191	0.212
Number Theory (741)	0.242	0.188

Table 3: Comparing results for different categories (the number between parenthesis indicates the number of entries for that category).

	TFIDF	PV-DBOW
1-hop premises	0.089	0.073
2-hop premises	0.052	0.047
3-hop premises	0.038	0.031

Table 4: Comparing number of hops needed for obtaining premises.

We also verify on how state-of-the-art embedding models perform with such specific dataset. BERT (Devlin et al., 2019) is reported to have performed in different NLP tasks, including understanding numeracy (Wallace et al., 2019).

In order to use BERT, we formulate the problem as a pairwise relevance classification problem, where we aim to classify if one mathematical text is connected to another. We do not perform any pre-processing for the expressions.

For this experiment, we used the pre-trained BERT model *bert-base-uncased* and SciBERT (Beltagy et al., 2019) model *scibert-scivocab-uncased*, fine-tuning for our task with a sequence classifier, adding a linear layer on top of the transformer vectors. The results are presented in Table 6.. Even though BERT is not pre-trained using a mathematical corpus, it performs better than TF-IDF and PV-DBOW. SciBERT perform slightly better than BERT, since it was trained in a scientific corpus, but not in a mathematical corpus. This hints that BERT trained from scratch in a mathematical corpus could have even better results, however, this is outside the scope of this work.

Model	MAP
SciBERT	0.383
BERT	0.377

Table 5: Results for BERT and SciBERT.

7. Conclusion

In this paper we proposed a new task for mathematical language processing: natural language premise selection. We also made a new dataset available for the evaluation of the task and we analysed how the dataset works with the task on different baselines.

From our experiments we identified that handling mathematical symbols are crucial for solving the task, taking into consideration more specific semantics of operators and variables: such semantics are not captured using PV-DM or BERT. This provides evidence on the need for specific embeddings and representation for mathematical formulas and discourse, which could most certainly improve the prediction of future work in the natural language premise selection task.

We also identify that the task becomes more challenging when we consider that the premises are transitive, suggesting that the task could benefit from graph-based representations.

Our dataset can be used in a different set of natural mathematical reasoning tasks, aiding researchers on the creation of mechanisms for improving the way machines understand mathematical text.

8. Acknowledgements

The authors would like to thank the anonymous reviewers for the constructive feedback.

9. Bibliographical References

Aizawa, A., Kohlhase, M., Ounis, I., and Schubotz, M. (2014). Ntcir-11 math-2 task overview. In *NTCIR*, volume 11, pages 88–98. Citeseer.

Alama, J., Heskes, T., Kühlwein, D., Tsivtsivadze, E., and Urban, J. (2014a). Premise selection for mathematics by corpus analysis and kernel methods. *Journal of Automated Reasoning*, 52(2):191–213.

Alama, J., Heskes, T., Kühlwein, D., Tsivtsivadze, E., and Urban, J. (2014b). Premise selection for mathematics by corpus analysis and kernel methods. *Journal of Automated Reasoning*, 52(2):191–213, Feb.

Beltagy, I., Lo, K., and Cohan, A. (2019). Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3606–3611.

Blanchette, J. C., Kaliszyk, C., Paulson, L. C., and Urban, J. (2016). Hammering towards qed. *Journal of Formalized Reasoning*, 9(1):101–148.

Chaganty, A. and Liang, P. (2016). How much is 131 million dollars? putting numbers in perspective with compositional descriptions. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 578–587.

Cramer, M., Fisseni, B., Koepke, P., Kühlwein, D., Schröder, B., and Veldman, J. (2009). The naproche project controlled natural language proof checking of mathematical texts. In *International Workshop on Controlled Natural Language*, pages 170–186. Springer.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Färber, M. and Kaliszyk, C. (2015). Random forests for premise selection. In Carsten Lutz et al., editors, *Frontiers of Combining Systems*, pages 325–340, Cham. Springer International Publishing.

Ganesalingam, M. and Gowers, W. T. (2017). A fully automatic theorem prover with human-style output. *Journal of Automated Reasoning*, 58(2):253–291.

Ganesalingam, M. (2013). The language of mathematics. In *The Language of Mathematics*, pages 17–38. Springer.

Gao, L., Jiang, Z., Yin, Y., Yuan, K., Yan, Z., and Tang, Z. (2017). Preliminary Exploration of Formula Embedding for Mathematical Information Retrieval: can mathematical formulae be embedded like a natural language? *CIKM 2017 Workshop on Interpretable Data Mining (IDM)*.

Gauthier, T. and Kaliszyk, C. (2015). Premise selection and external provers for hol4. In *Proceedings of the 2015 Conference on Certified Programs and Proofs*, CPP '15, pages 49–57, New York, NY, USA. ACM.

Greiner-Petter, A., Ruas, T., Schubotz, M., Aizawa, A., Grosky, W., and Gipp, B. (2019). Why Machines Cannot Learn Mathematics, Yet. *4th BIRNDL workshop at 42nd SIGIR*.

Huang, D., Shi, S., Lin, C.-Y., Yin, J., and Ma, W.-Y. (2016). How well do computers solve math word problems? large-scale dataset construction and evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 887–896.

Irving, G., Szegedy, C., Alemi, A. A., Een, N., Chollet, F., and Urban, J. (2016). Deepmath-deep sequence models for premise selection. In *Advances in Neural Information Processing Systems*, pages 2235–2243.

Krstovski, K. and Blei, D. M. (2018). Equation Embeddings.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Solow, D. (2002). How to read and do proofs an introduction to mathematical thought processes.

Wallace, E., Wang, Y., Li, S., Singh, S., and Gardner, M. (2019). Do nlp models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5310–5318.

Wang, Q., Kaliszyk, C., and Urban, J. (2018). First experiments with neural translation of informal to formal mathematics. In *International Conference on Intelligent Computer Mathematics*, pages 255–270. Springer.

Warner, S. (1990). *Modern algebra*. Courier Corporation.

Zinn, C. (2003). A computational framework for understanding mathematical discourse. *Logic Journal of IGPL*, 11(4):457–484.