Premise Selection in Natural Language Mathematical Texts

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

Abstract

The discovery of supporting evidence for addressing complex mathematical problems is a semantically challenging task, which is still unexplored in the field of natural language processing for mathematical text. The natural language premise selection task consists in using conjectures written in both natural language and mathematical formulae to recommend premises that most likely will be useful to prove a particular statement. We propose an approach to solve this task as a link prediction problem, using Deep Convolutional Graph Neural Networks. This paper also analyses how different baselines perform in this task and shows that a graph structure can provide higher F1-score, especially when considering multi-hop premise selection.

1 Introduction

Mathematical proofs are used to establish the truth value of a mathematical claim. The act of creating a new proof contributes to the development of Mathematics, being one of its central components.

Premise selection is a well-defined task in the field of Automated Theorem Proving (ATP), where proofs are encoded using a formal logical representation. Given a set of premises P, and a new conjecture c, premise selection aims to predict those premises from P that will most likely lead to an automatically constructed proof of c, where P and c are both written using a formal language (Irving et al., 2016).

The issue with using formal mathematics is that only a small portion of the known mathematical statements is available in a formalised dataset, and formal statements are usually hard for humans to interpret and write.

In this paper, we focus on natural language mathematical text (mathematical statements as they are present in scientific papers and textbooks), since it is more accessible for mathematicians to write/read mathematical statements using natural language. The mathematical discourse is composed of a particular combination of words and mathematical terms, where terms follow a different set of syntactic rules and entail a specific lexicon. Nonetheless, words and mathematical terms 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.

The *natural language premise selection* (Ferreira and Freitas, 2020) task is defined as:

Definition (Natural language premise selection): Given a set of premises (or supporting facts) P in a mathematical corpus (containing both natural language and formulae) and a new conjecture c proposed by a user, predict those premises from P that will most likely be useful for generating a proof for c (i.e. partially entails c).

A premise is considered relevant if the knowledge it provides can be reused for generating a proof for a given conjecture.

We propose an approach to solve the natural premise selection task, representing all conjectures and premises as nodes and the dependencies as edges, formulating the problem as a *link prediction* problem. We hypothesise that graph-based embeddings are suitable structures for representing and detecting the dependencies between different mathematical statements. We then use Deep Convolutional Graph Neural Networks (Zhang et al., 2018) over a structural and content-based encoding of proofs in order to obtain the set of useful premises for proving a statement.

In order to evaluate this task, we use the dataset PS-ProofWiki. This dataset opens possibilities of applications not only for the premise selection task

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7365–7374

but also for evaluating different equational embeddings, textual entailment for mathematics and natural language inference in the context of mathematical texts. The performance of the proposed model is compared to a set of baselines.

The contributions of this paper can be summarised as follows: (i) Proposal of a novel representation for the natural language premise selection problem. (ii) Proposal of an approach for addressing the natural language premise selection task using link prediction under a Deep Convolutional Graph Neural Network representation. (iii) Quantitative and qualitative evaluation against existing baselines.

2 Related Work

Latent and explicit representation models have seen a substantial advance in the past years, with the introduction of neural embeddings such as BERT (Devlin et al., 2018), which are able to capture discourse-level relations and semantic abstractions. However, the development of representation models and their evaluation in the context of mathematical discourse is still an open problem.

In this section, we present some of the research in NLP applied to mathematics. We also describe existing works that apply premise selection in the domain of ATPs.

Mathematical Language Processing A relevant area that intersects both NLP and mathematical discourse is the research on how to automatically solve math word problems. Wang et al. (2018) test how different Seq2Seq models perform on mathematical word problems, where each question has a set of possible solution equations and the different equations are normalised to the same tree representation. Huang et al. (2016) analyse various approaches to solve mathematical word problems and concludes that it is still an unsolved challenge. Xie and Sun (2019) proposes a neural model to generate an expression tree following a reasoning similar to the way humans solve math word problems. Text2Math is an approach to solve arithmetic word problems and equation parsing tasks by proposing a joint representation to learn the correspondence between words and math expressions (Zou and Lu, 2019).

On the discourse analysis domain, Zinn (2003) introduces a proof representation structure for mathematical discourse using discourse representation theory and presents a prototype for automating the process of generating proofs. 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 syntactic and mathematical correctness. Ganesalingam and Gowers (2017) propose a program that solves elementary mathematical problems, with the focus on metric space theory, and presents solutions similar to the ones introduced by humans. The authors recognise 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.

Different works started exploring equational embeddings. EqEmbs (Krstovski and Blei, 2018) is built on exponential family embeddings, considering equations as single elements, modelling part of the equations, such as variables, symbols and operators. EqEmbs considers the context for the equations as a window of sixteen words. Tangent-CFT (Mansouri et al., 2019) uses fastText to produce formula embeddings for symbol layout trees (SLTs) and operator trees (OPTs). The embedding procedure converts the representation into a sequence of tuples, where the elements are tokenised as characters. The tuples are embedded using n-grams computed over the tuple and its neighbouring tuples. Greiner-Petter et al. (2019) developed a skip-gram-based model using as a reference corpus a collection of arXiv papers in HTML format using a term-level tokenisation granularity. The authors found that the induced vector space did not produce meaningful semantic clusters. Wallace et al. (2019) found that CNNs are useful for tasks involving understanding and working with numbers; however, it still struggles to extrapolate beyond the values seen during training.

Premise Selection Premise selection is an approach generally used for selecting useful premises to prove conjectures in Automated Theorem Proving (ATP) systems (Alama et al., 2014). Irving et al. (2016) propose a neural architecture for premise selection using formal statements written in Mizar. The authors were able to solve 67.9% of the conjectures present in the Mathematical Mizar Library. Other authors have used machine learning approaches such as Kernel-based Learning (Alama et al., 2014), k-NN algorithm (Gauthier and Kaliszyk, 2015) and Random Forests (Färber

and Kaliszyk, 2015).

Contrasted to related work, the model proposed on this paper targets capturing both content (local) and structural dependencies (global) across natural language mathematical statements and its evaluation on the natural language premise selection problem.

3 The Natural Language Premise Selection task

Figure 1 depicts an example of a theorem and its proof, where it can be observed that the proof is based upon two other supporting facts (premises): the theorem for *Factors of Composition Series for Prime Power Group* and the definition for *Solvable Group*.

In order to evaluate the premise selection, we used a corpus extracted from ProofWiki[1]. 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. Definitions and axioms are statements accepted without formal proof, while theorems, lemmas and corollaries require one (Solow, 2002). All entries are composed by a statement written in a combination of natural language and mathematical latex notation. The extracted corpus, which is named PS-ProofWiki, contains more than $18,000$ entries. 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 length of 289 symbols (characters and mathematical symbols). The specific number of tokens will depend on the type of tokenisation used for the mathematical symbols. A complete analysis of this corpus is made available in (Ferreira and Freitas, 2020).

In the next sections, we describe the proposed model for addressing the premise selection task. The proposed model uses a Deep Graph Convolutional Neural Network (DGCNN) for solving the premise selection task as a link prediction task (Zhang and Chen, 2018). The proposed model aims to encode the natural language and the formulae terms as well as the dependencies and graph-structural patterns of the mathematical text.

[1] http://proofwiki.org/

4 Encoding mathematical propositions and supporting facts

4.1 Graph construction

In Mathematics, theorems are always built on top of previous mathematical knowledge, such as lemmas, corollaries, definitions and other theorems. Thus, Mathematics as a discourse intrinsically entails a network structure. With this hierarchy and interlinking of concepts in mind, we developed a graph representation to represent all mathematical statements present in the corpus and their associated dependencies.

The extracted dependency graph is a directed graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices, composed by mathematical statements and $\mathcal{E}$ is a set of ordered pairs of vertices (edges), in this case the relationship between mathematical statements. If $m_1, m_2 \in \mathcal{V}$ and $(m_1, m_2) \in \mathcal{E}$ that means the statement m_1 is a premise to the statement m_2.

4.2 Subgraph extraction

From the set of graphs containing all asserted dependency relations, an enclosing sub-graph (with a fixed hop h size of $1 \leq h \leq 2$) is extracted by selecting a pair of nodes as the target. These pair of nodes will be used to define the link prediction classification context, in which a binary class is assigned, P when $(m_1, m_2) \in \mathcal{E}$ and NP (not a premise) otherwise (Figure 2).

As we predict the link between different statements, we are also predicting the dependencies between different statements, therefore, addressing the natural premise selection problem.

4.3 Node features

Every node $m_i \in \mathcal{V}$ is composed of two parts: (1) a label based on a function which encodes its neighbourhood, (2) an embedding of its textual content.

The framework generates labels for the nodes using the *Double-Radius Node Labelling* (DRNL) (Zhang and Chen, 2018) mapping, assuming that the graph is undirected. The labelling technique was altered so it could also work for a directed graph setting. Considering two different statements $m_1, m_2 \in \mathcal{V}$, where we want to predict if m_2 is a premise for m_1; all nodes are labelled as follows: (i) m_1 is labelled as 1, (ii) m_2 is labelled

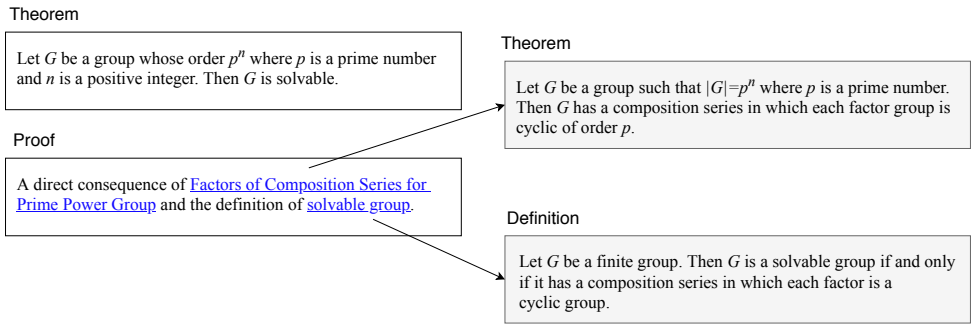


Figure 1: Theorem and premises for the theorem "Prime Power Group is Solvable".

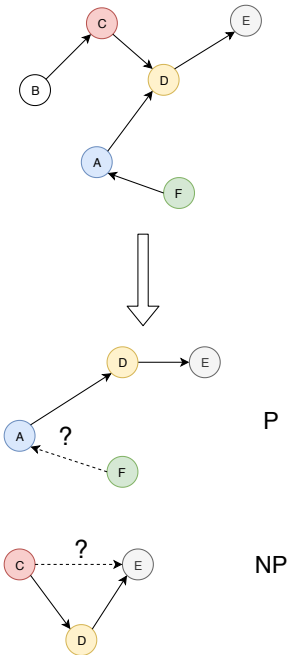


Figure 2: Sub-graph extraction for link prediction.

as 2, (iii) for every x in S reachable from m_1, label x as the distance between m_1 and x, (iv) for every y in S unreachable from m_1, label y as 0.

The embedding of the textual content is an embedding of the mathematical statements. A mathematical statement is composed of a hybrid setting of mathematical notation and natural language statements. Paragraph Vector Distributed Memory (PV-DM/Doc2Vec) (Le and Mikolov, 2014) was used to encode a statement-level representation of the constituent statements of the proof (where each statement is a 'paragraph'). The expressions and equations are encoded as a tree, by representing every sub-expression as a token. For example, the expression '$(x + y) * c$' is represented as the sequence of tokens ['x','y','$(x + y)$','$(x + y) * c$'], capturing the syntactic structure of the mathematical expression. The same model captures both the natural language and the formulae tokens. Figure 3 depicts how the structural and content aspects are represented.

5 Proposed Model: Premise Selection based on DGCNNs

5.1 Design Principles

A Deep Graph Convolutional Neural Network (DGCNN) architecture (Zhang et al., 2018) was used as the default GNN engine of the premise selection. The architecture was selected due to its ability to encode network features with a consistent performance across different graph network (GN) evaluation scenarios. Moreover, we use the graph encoding proposed in (Zhang and Chen, 2018), which aims for learning subgraph structural patterns using DGCNNs. This approach embeds the learning of a problem-specific graph heuristic function (which is formalised as the γ-decaying heuristic theory). This can be contrasted with the use of pre-defined methods from a single heuristic framework (such as Katz index, PageRank and SimRank (Zhang and Chen, 2018)), by using a graph-specific approximation instead.

The underlying assumption behind the selection of the base architecture is that the premise selection problem requires the encoding of both the statement content and of the graph-dependency patterns.

The final problem of premise selection is rephrased as a problem of link prediction, and the final classification layer has a binary classifier. Figure 4 depicts the main components of an end-to-end architecture.

5.2 Detailed Model

A denotes the adjacency matrix of a graph, n the number of vertices where each vertex has a c-dimensional feature vector, denoted as $X \in R^{n \times c}$. For a vertex v, we use $\Gamma(v)$ to denote the set of

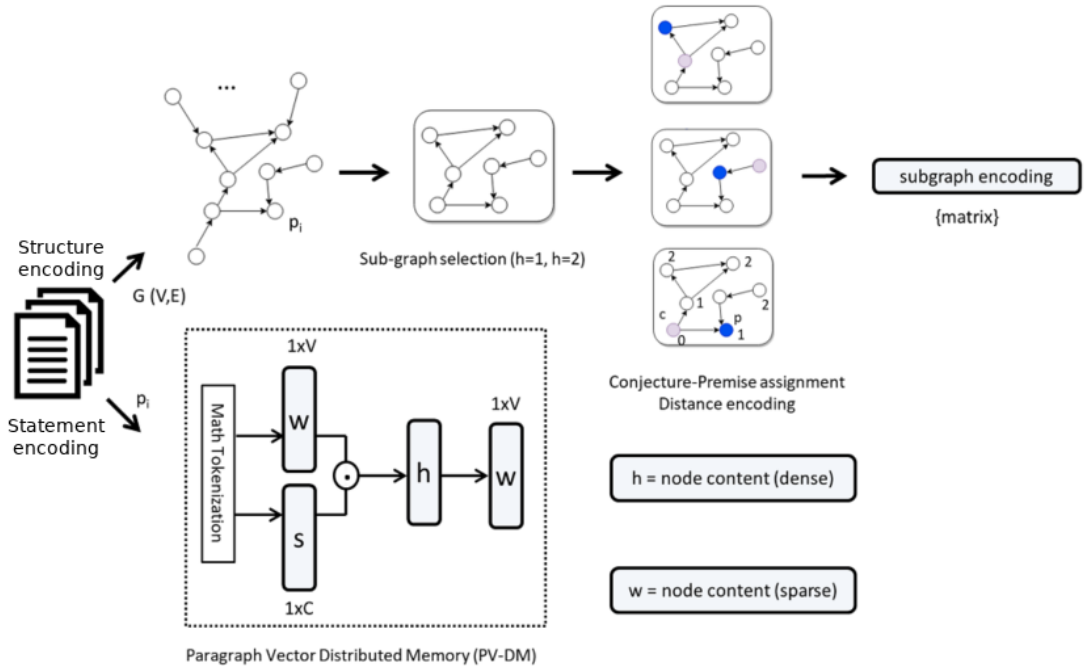


Figure 3: Pre-processing workflow of the proof corpus.

v's neighbouring nodes. DGCNN uses the graph convolution function:

$$Z = f(\tilde{D}^{-1}\tilde{A}XW) \quad (1)$$

where $W \in R^{c \times c}$ is a weight matrix of graph convolution parameters, $\tilde{A} = A + I$ based on the adjacency matrix A, $\tilde{D}$ is a diagonal degree matrix (Zhang and Chen, 2018) and f is a non-linear activation function. $\tilde{D}^{-1}\tilde{A}$ is a propagation matrix.

The graph aggregation layer builds for each node a graph-level feature vector based the individual node states, which is defined by:

$$Z_i = f(\frac{1}{|\Gamma(i)| + 1}[X_i W + \sum_{j \in \Gamma(i)} X_j W]) \quad (2)$$

The graph convolution aggregates node patterns, extracting local subgraph patterns. The last graph convolution layer output can be used to sort the graph vertices in an order which reflects the vertices structural roles (Zhang and Chen, 2018).

After the aggregation, the DGCNN uses a *sort pooling layer*, which sorts the final node states based on to the last graph convolution layer's output (Zhang and Chen, 2018). The sorting criteria are based on a topological-based ordering. For example (Niepert et al., 2016) provide a labelling scheme for vertexes based on topological patterns. This topological ordering is consistent across graphs: vertices in two different graphs will be assigned similar relative positions if they have similar structural roles (Zhang et al., 2018).

The ordering operation is followed by a *max-k pooling operation* which creates a representation for the different graphs with uniform dimensions (truncating or extending into k dimensions). This allows the application of a *1-D CNN layer* on the node sequence. A *final dense layer* connected to a *softmax layer* performs the binary classification of the target vertices into the *premise/non-premise case*.

A standard DGCNN configuration is used (Zhang et al., 2018), containing four graph convolution layers, a sort pooling layer with a k assignment 0.60 (graph coverage), two 1-D convolution layers and a dense layer with 128 neurons.

5.3 Assumptions & Critique

The proposed model has a locality assumption expressed at the statement encoding level, which limits the proof neighbourhood to two hops. This follows the intuition that the premise selection model aims to reflect the mentioned structure of proofs (expanding, however an additional hop) privileging the classification of closer and more specific conjecture-premise relations. More exploratory types of proofs may require the expansion of the hops to cope with longer distance relations.

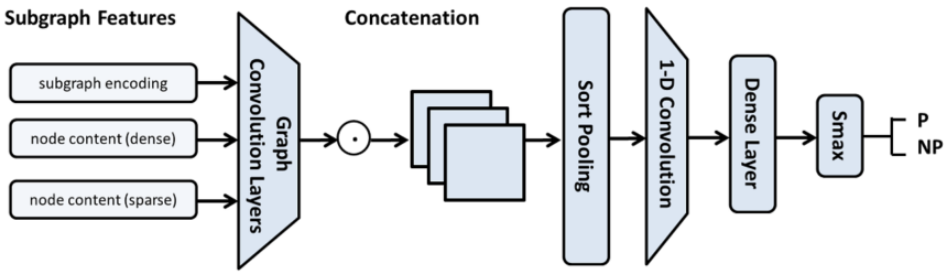


Figure 4: Depiction of the DGCNN architecture used in the premise selection task.

6 Evaluation

This section evaluates the performance of the proposed model using PS-ProofWiki. We introduce initial baselines using two basic approaches, TF-IDF and PV-DBOW. These are further expanded using a transformer-based architecture (BERT), due to its state-of-art results for the encoding of sentence-level embeddings and their use in tasks such as natural language inference.

For the experiments using BERT and the proposed approach, we split the dataset using a 50/20/30 (train/dev/test) split. We run all experiments ten times, evaluating on the test set, and report the average Precision, Recall and F1-score. All evaluation data, as well as the experimental pipeline, can be found online[2] for reproducibility purposes.

6.1 Bag-of-Words Baselines

In order to identify the challenges of the task of natural language premise selection using PS-ProofWiki, we performed initial experiments using two Bag-of-words (BoW) baselines: TF-IDF and PV-DBOW (Le and Mikolov, 2014). We use both weighting schemes to define the vector representations for all mathematical statements. Then we compute the cosine similarity between each entry and rank the results by their distance. The Mean Average Precision (MAP) is computed for each baseline:

$$MAP = \frac{\sum_{i=1}^{N} AvegP(s_i)}{N} \quad (3)$$

where N is the total number of statements, s_i is the i-th mathematical statement and AvegP is the average precision. MAP has been used in similar ranking tasks, such as supporting facts (explanations) retrieval (Valentino et al., 2020).

Table 1 presents the results for the BoW baselines. Three different types of tokenisations are compared for encoding the mathematical expressions. In the first instance, we treat the expressions and equations as single tokens; for example, the expression "$x + y + z$" would be considered a single token. We also considered tokenised expressions, tokenising variables and operators, the example would be tokenised as ['x','+','y','+','z']. In both examples, the natural language part of the text is tokenised as a sequence of words. 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 the MAPs obtained by the BoW, we can conclude that the task is semantically non-trivial and cannot be addressed with retrieval-based strategies which are based on lexical overlap. We can also notice that better results are obtained when the expressions are tokenised as a sequence of operations and variables, suggesting 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 for the encoding of the expressions.

In Table 2 we compare the results for different sizes of the dataset. We consider the full dataset and three different subsets with different categories of mathematical statements. 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 have a higher lexical

[2] https://github.com/ai-systems/premise_selection_graph

Table 1: MAP results for TF-IDF and PV-DBOW comparing different tokenisation strategies for the mathematical expressions.

	TF-IDF	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

overlap.

Table 2: Comparing results for different categories (the number in parenthesis indicates the number of entries for that category).

	TF-IDF	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

We can also consider the fact that premises are transitive, i.e., if one a mathematical text t_i has a premise x and a mathematical text t_j has t_i as a premise, then x should also be a premise of t_j. In this case, the task becomes semantically more challenging, as it can be observed in Table 3, where we consider the transitivity within 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.

Table 3: Comparing number of hops needed for obtaining premises.

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

6.2 Baseline: BERT

In order to use BERT, we reformulate this problem as a pairwise relevance classification problem, as done previously in the context of ATP systems. We have a set of mathematical statements S, a set of conjectures C and a set of premises P, where $C \subseteq P$, $C \subseteq S$ and $P \subseteq S$. Considering a conjecture $c \in C$ and a premise $p \in P$, a function $f(c, p)$ is defined, where $f(c, p) = 1$ if p is a part of the proof of c and $f(c, p) = 0$ otherwise.

For this experiment, we used the pre-trained BERT model *bert-base-uncased*, fine-tuning it for the target task with a sequence classifier, adding a linear layer on top of the transformer embeddings.

6.3 Quantitative analysis

The dataset is imbalanced by the nature of the natural premise selection problem. In order to solve the natural premise selection task, any approach would have to be able to handle a large number of negative examples. There are 10k different possible premises, and some conjectures are only connected to one premise, creating a large number of negative pairs in our dataset, requiring the definition of a cap for the number of negative samples. In order to provide a more constrained setting, we define a subset of the PS-ProofWiki, named PS-ProofWiki$_{TRIG}$ targeting trigonometric functions.

The proposed approach outperforms the BERT-based model by 41% in terms of F1-score, as shown in Table 4. We hypothesise that the encoding of the structural patterns of the dependency relations in addition to the content-based similarity better captures the semantic nature of the proof (fundamental to interpret a proof by its neighbourhood).

6.4 Scalability & Imbalance Robustness analysis

In order to evaluate the robustness of the proposed approach and the baseline with regard to an increase in imbalance (reflecting a notion of *scalability* of the quality of the inference within the KB), we compare how the F1-score changes as we add more (random) negative examples to the dataset.

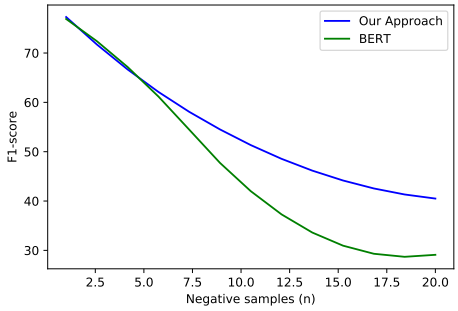
Figure 5a and Figure 5b presents a comparison between BERT and our approach for the PS-ProofWiki$_{TRIG}$ and the PS-ProofWiki datasets, respectively.

The results indicate that the BERT-based classifier performance degrades faster as we increase the number of negative samples in the dataset. For $n = 30$, the F1-score reaches a value of almost zero. In contrast, the proposed model presents a significantly slower decline (25%), showing better scalability properties in the context of the premise selection problem.

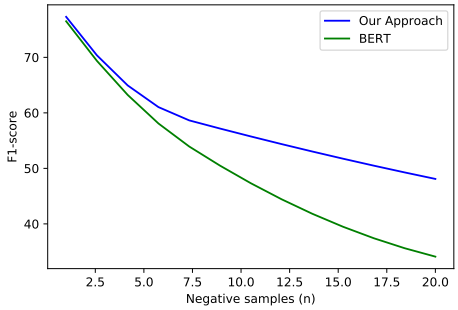
Finally we experiment on how BERT and the proposed model compares when we consider transitivity between premises (n-hop relations), using PS-ProofWiki$_{TRIG}$ and 10 negative examples for each positive example. We report the results in Table 5, where we can see that the proposed model obtains better overall performance as the number of

Table 4: Precision (P), recall (R), and F1-score (F1) for the BERT baseline and the proposed approach, with 30 negative examples for each positive case (values are multiplied by 100).

	BERT			Proposed Model		
	P	R	F1	P	R	F1
PS-ProofWiki$_{TRIG}$	**39.9**	22.9	29.1	34.0	**50.0**	40.5 (+ 39%)
PS-ProofWiki	**47.1**	26.7	34.1	**48.5**	47.7	48.1 (+ 41%)



(a) Evaluating on PS-ProofWiki$_{TRIG}$



(b) Evaluating on PS-ProofWiki

Figure 5: Comparison of the proposed model and BERT, showing how both models perform (in terms of F1-score) when adding more negative examples to the training and test set.

hops is increased. These results reinforce the architectural design supported by graph-based models.

6.5 Qualitative analysis

From the results obtained from our model we observed that the model struggles to encode statements which are centered around pure equational (formulae) content. Embeddings for mathematical symbols should take into consideration more specific semantics of operators: such semantics is not obtained using PV-DM (Doc2Vec) or BERT. This provides evidence on the need for more principled structural embeddings for mathematical formulas, which could most certainly improve the prediction of future work in the natural premise selection task.

Table 5: Comparison of BERT and the proposed model for different levels of transitivity between premises (values are multiplied by 100).

	BERT			Proposed Model		
	P	R	F1	P	R	F1
2-hop	47.5	**78.9**	59.3	54.8	**68.7**	61.0 (+ 3%)
3-hop	41.0	45.1	**49.2**	58.8	**63.3**	61.2 (+ 24%)

Even though BERT is not trained in a mathematical corpus, it still obtains relevant results, hinting that training BERT on a mathematical corpus could achieve better results. However, this task is outside the scope of this work and will be left for future work.

The proposed DGCNN-based model is capable of finding structural patterns between the statements and to reinforce content-based semantic evidence. We observed that statements that are similar in content, commonly have a significant intersection of premises, as a result of the graph embedding, the DGCNN-model is able to better discriminate more fine-grained semantic cues better.

7 Conclusion & Future Work

In this work, we introduced an approach for natural language premise selection (finding relevant theorems, axioms and definitions) in large natural language mathematical texts. The proposed approach, which uses Deep Graph Convolutional Neural Networks (DGCNNs) combines both structural and content elements of mathematical statements for addressing the premise selection problem as a link prediction classification problem. Results show that the approach outperforms a BERT-based baseline by 41% in F1-score. Moreover, the proposed model shows significantly lower F1-score degradation concerning class imbalance, a fundamental desirable scalability property for the problem of premise selection.

Our approach is also able to obtain better performance when we consider the transitivity of premises. The qualitative analysis indicates that

there is the demand to design principled embeddings for better capturing the semantics of proofs which are denser in mathematical formulae. As future work, we will explore different heuristics for navigating in the premises graph, as researched before for textual entailment (Silva et al., 2019, 2018) and selective reasoning (Freitas et al., 2014).

Acknowledgments

The authors would like to thank the anonymous reviewers for the constructive feedback, we also would like to thank Mokanarangan Thayaparan and Marco Valentino for the helpful discussions.

References

Jesse Alama, Tom Heskes, Daniel Kühlwein, Evgeni Tsivtsivadze, and Josef Urban. 2014. Premise selection for mathematics by corpus analysis and kernel methods. *Journal of Automated Reasoning*, 52(2):191–213.

Marcos Cramer, Bernhard Fisseni, Peter Koepke, Daniel Kühlwein, Bernhard Schröder, and Jip Veldman. 2009. The naproche project controlled natural language proof checking of mathematical texts. In *International Workshop on Controlled Natural Language*, pages 170–186. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Michael Färber and Cezary Kaliszyk. 2015. Random forests for premise selection. In *Frontiers of Combining Systems*, pages 325–340, Cham. Springer International Publishing.

Deborah Ferreira and Andre Freitas. 2020. Natural language premise selection: Finding supporting statements for mathematical texts. *12th Language Resources and Evaluation Conference (LREC), Marseille, France.*

André Freitas, Joao Carlos Pereira da Silva, Edward Curry, and Paul Buitelaar. 2014. A distributional semantics approach for selective reasoning on commonsense graph knowledge bases. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 21–32. Springer.

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

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

Thibault Gauthier and Cezary Kaliszyk. 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.

Andr Greiner-Petter, Terry Ruas, Moritz Schubotz, Akiko Aizawa, William Grosky, and Bela Gipp. 2019. Why Machines Cannot Learn Mathematics, Yet. (July).

Danqing Huang, Shuming Shi, Chin-Yew Lin, Jian Yin, and Wei-Ying Ma. 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.

Geoffrey Irving, Christian Szegedy, Alexander A Alemi, Niklas Eén, François Chollet, and Josef Urban. 2016. Deepmath-deep sequence models for premise selection. In *Advances in Neural Information Processing Systems*, pages 2235–2243.

Kriste Krstovski and David M Blei. 2018. Equation embeddings. *arXiv preprint arXiv:1803.09123*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Behrooz Mansouri, Shaurya Rohatgi, Douglas W Oard, Jian Wu, C Lee Giles, and Richard Zanibbi. 2019. Tangent-cft: An embedding model for mathematical formulas.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 2014–2023. JMLR.org.

Vivian Dos Santos Silva, Siegfried Handschuh, and André Freitas. 2018. Recognizing and justifying text entailment through distributional navigation on definition graphs. In *AAAI*.

Vivian S Silva, André Freitas, and Siegfried Handschuh. 2019. Exploring knowledge graphs in an interpretable composite approach for text entailment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7023–7030.

Daniel Solow. 2002. How to read and do proofs an introduction to mathematical thought processes.

Marco Valentino, Mokanarangan Thayaparan, and Andr Freitas. 2020. Unification-based reconstruction of explanations for science questions.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. *arXiv preprint arXiv:1909.07940*.

Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a math word problem to an expression tree. *arXiv preprint arXiv:1811.05632*.

Zhipeng Xie and Shichao Sun. 2019. A goal-driven tree-structured neural model for math word problems. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5299–5305. AAAI Press.

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, pages 5165–5175.

Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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

Yanyan Zou and Wei Lu. 2019. Text2math: End-to-end parsing text into math expressions. *arXiv preprint arXiv:1910.06571*.