Extracting Operator Trees from Model Embeddings

Anja Reusch and **Wolfgang Lehner**
Database Systems Group,
Technische Universität Dresden, Germany
firstname.lastname@tu-dresden.de

Abstract

Transformer-based language models are able to capture several linguistic properties such as hierarchical structures like dependency or constituency trees. Whether similar structures for mathematics are extractable from language models has not yet been explored. This work aims to probe current state-of-the-art models for the extractability of Operator Trees from their contextualized embeddings using the structure probe designed by (Hewitt and Manning, 2019). We release the code and our data set for future analyses[1].

1 Introduction

Transformer-based Language Models have not only a high impact on all domains in Natural Language Understanding but also on related fields that besides natural language try to model artificial languages such as programming code or mathematical notation written in LaTeX (Feng et al., 2020; Peng et al., 2021). The knowledge or linguistic properties that models like BERT or RoBERTa capture have been the subject of several studies: According to recent research, BERT encodes information about part-of-speech tags, roles, and syntactic features such as constituency and dependency trees (Rogers et al., 2020). Since transformer-encoder-based models were applied successfully for mathematical question answering or notation prediction (Reusch et al., 2022b; Jo et al., 2021), these models must have also acquired mathematical knowledge. However, the field of interpretability for mathematical information has not been a topic of research so far. Therefore, this work aims to analyze the prevalence of one type of mathematical knowledge: Operator Trees, a type of parse trees that can be generated from LaTeX formulas.

Generally, whether a model encodes a certain property is evaluated by applying a probe, i.e., a classifier that is trained on top of the contextualized embeddings. The performance of this classifier is used as an indicator whether the information about the property was encoded in the contextualized embeddings. To analyze whether it is possible to reconstruct an Operator Tree from the contextualized embeddings of a transformer-encoder model, we apply the structural probe introduced by (Hewitt and Manning, 2019). This probe approximates the distance between nodes in the trees using the distance of two embeddings.

In total, we train the structural probe on the embeddings of each layer of nine models for math and science and show that in most cases it is possible to reconstruct Operator Trees from the models' contextualized embeddings. The highest correlation between the learned tree distance and the gold standard is reached in the middle layers, e.g., around layer 6 for models based on bert-base and roberta-base. As Hewitt and Manning also found for dependency trees, most models follow a similar pattern of information spreading among layers.

2 Related Work

Within the last years, several transformer-encoder-based models for mathematics have been developed with different applications in mind. The recent ARQMath Lab 3 (Mansouri et al., 2022) included several teams that applied models pre-trained on math: MIRMU used mathBERTa, a model based on roberta-base (Novotnỳ and Štefánik, 2022; Geletka et al., 2022), (Reusch et al., 2022a) adapted albert-base-v2 and roberta-base for math, and (Zhong et al., 2022) further pre-trained a BERT model. In ARQMath Lab 1, the team PSU also released a further pre-trained model based on RoBERTa (Rohatgi et al., 2020). In addition, (Jo et al., 2021) fine-tuned a BERT model for notation prediction tasks based on scientific documents. MathBERT (Peng et al., 2021) leverages operator trees during pre-training for several tasks such as formula topic

[1] https://github.com/AnReu/extracting-opts

classification and information retrieval. Related to mathematics is also the domain of scientific documents for which SciBERT was trained (Beltagy et al., 2019).

However, little is known so far about what BERT-based models learn about mathematics. In contrast, their learning capacities on natural language received large attention in recent research (for a survey see (Rogers et al., 2020)). Several probes and classifiers were employed to analyze whether BERT captures grammatical structures like dependency or constituency trees (Tenney et al., 2019; Hewitt and Manning, 2019; Coenen et al., 2019) or which layer attends to which linguistic feature (Clark et al., 2019). Visual frameworks like bertviz by (Vig, 2019) support the analysis of BERT's inner working by visualizing the attention weights of trained models. Also ALBERT was shown to capture part-of-speech tags in different places as reported by (Chiang et al., 2020), but most studies were performed using BERT.

3 Probing for Mathematical Structures

We analyze whether it is possible to reconstruct mathematical parse trees from the models' contextualized embeddings. It was already shown that BERT is able to learn grammatical structures of natural languages which could be extracted in the form of constituency and dependency trees (Tenney et al., 2019; Hewitt and Manning, 2019). Therefore, we apply the same type of probe to test for Operator Trees.

3.1 Structural Probe

The goal of the structural probe as introduced by (Hewitt and Manning, 2019) is to learn a matrix B, such that the distance d_B defined by $d_B(U_i, U_j) := \sqrt{(U_i - U_j)^\intercal B^\intercal B(U_i - U_j)}$ approximates a given tree distance d_T, i.e., the length of the path between the node of word s_i and the one of word s_j in the tree of example s. U_i and U_j are the contextualized embeddings of the words s_i and s_j in s. B is learned by minimizing the loss function over each examples $s \in S$ in the training corpus:

$$\min_B \sum_{s \in S} \frac{1}{|s|^2} \sum_{i,j} |d_T(s_i, s_j) - d_B(U_i, U_j)|$$

Originally, Hewitt and Manning applied the structural probe to demonstrate that dependency structures of the English language are, to some extent, contained in BERT's contextualized embeddings. In this work, we will train a structure probe to evaluate whether the models' inner workings have learned about mathematical structures, i.e., operator trees.

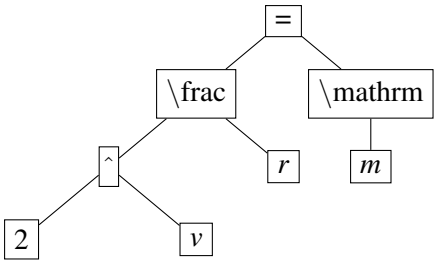

Figure 1: Operator Tree of the formula $\mathrm{m} = \frac{r}{v^2}$

3.2 Operator Trees

Formulas possess a hierarchical structure, which is encoded in Content Math ML [2], defining an operator tree (OPT). An example OPT for the equation $\mathrm{m} = \frac{r}{v^2}$ is shown in Fig. 1. Nodes of this tree representation can be individual or multiple symbols such as numbers, variables, text fragments indicating certain functions, fractions, radicals, LaTeX style expressions, or parentheses and brackets. This definition is similar to the one found in (Mansouri et al., 2019), but we added parentheses and brackets to investigate the way the models capture open and closed bracket relationships. OPT edges indicate an operator-argument relationship between parent and child nodes. Left and right brackets and parentheses have each an edge to the parent of the tree inside them. LaTeX style expressions like `\mathbb` can be simply seen as an operator applied on the argument inside. Hence, the original OPT stays intact.

4 Experimental Setup

We evaluated in total 13 models which are publicly available on the Huggingface Model Hub[3]. We chose the eight mathematical models by searching the Model Hub for transformer-encoder models that were (further) pre-trained on mathematics. We also added the popular model SciBERT as its domain, science, is close to mathematics. In addition, the four models which served as a base for pre-training were evaluated. A summary of the models can be found in Tab. 1. Of particular interest would have been an evaluation of MathBERT by (Peng et al.,

[2] https://w3c.github.io/mathml/#contm
[3] https://huggingface.co/models

Model Identifier	Base Model	Data Set
albert-base-v2	-	Books and Wikipedia
AnReu/math_albert	albert-base-v2	ARQMath
bert-base-cased	-	Books and Wikipedia
allenai/scibert-scivocab-cased	-	Scientific documents
AnReu/math_pretrained_bert	bert-base-cased	ARQMath
tbs17/MathBERT	-	Math text books, curricula, paper abstracts
tbs17/ MathBERT-custom	-	Math text books, curricula, paper abstracts
roberta-base	-	Books, Wikipedia, news, websites, stories
roberta-large	-	Books, Wikipedia, news, websites, stories
AnReu/math_pretrained_roberta	roberta-base	ARQMath
shauryr/arqmath-roberta-base	roberta-base	ARQMath
uf-aice-lab/math-roberta	roberta-large	Math discussion posts
witiko/mathberta	roberta-base	ARQMath, ArXiv documents

Table 1: Summary of the evaluated models, their base models and the data sets used for pre-training.

2021), a model that relied on Operator Trees during pre-training. However, neither the model nor the code are publicly available.

BERT, ALBERT and RoBERTa were trained on a general natural language corpus and serve as baselines. Six models were further pre-trained from a base model like BERT or RoBERTa, while three were developed from scratch. The data sources the models were trained on are rather diverse: Five models use ARQMath, others use math text books, school curricula, paper abstracts, or other discussion posts apart from ARQMath. SciBERT is the only model what was not specifically trained on mathematical content, but on scientific publications. All models can be found on Huggingface by using their model identifier.

4.1 Data

Our probe is trained on formulas, which were parsed to OPTs by a custom LaTeX parser written in Python adapted from the parser rules of the mathematical formula search engine Approach0 (Zhong and Zanibbi, 2019; Zhong et al., 2020). We could not use existing parsers because it is necessary to associate each LaTeX token with its node in the OPT and existing parsers only output the entire parse tree without annotation of a node's token in the formula. We selected 50k training examples by chance from the corpus of all formulas from ARQMath 2020 (Mansouri et al., 2020), which contains question and answer posts from the Q&A community Mathematics StackExchange[4]. From the remaining set we chose 10k for development, and an additional set of 10k as test set. The average number of nodes in all three sets is 16.5, while the average tree depth is 4.8. The most common node types are variables and numbers, followed by LaTeX braces and relation symbols. Among the relation symbols, the equal sign "=" occurs most often.

4.2 Metrics

We follow Hewitt and Manning and evaluate the performance using UUAS (Undirected Unlabeled Attachment Score), which denotes the percentage of correctly identified edges in the predicted tree and, distance Spearman (DSpr.), which is determined by first calculating the Spearman correlation between the predicted distances d_B and the gold-standard distances d_T. These correlations are then averaged among all formulas of a fixed length. Finally, the average across formulas of lengths 5–50 is reported as DSpr. We decided to include both metrics since it was shown that their scores can result in opposite trends (Hall Maudslay et al., 2020).

4.3 Setup

To train and evaluate the probing classifier, we used the original code provided by Hewitt and Manning[5] and adapted it to the transformers library[6]. We used

[4] https://math.stackexchange.com

[5] https://github.com/john-hewitt/structural-probes

[6] https://pypi.org/project/transformers/

the L1 loss and a maximum rank of the probe of 768, as reported by the authors. We trained the probes using one A100 GPU with 40 GB GPU memory. Depending on the base model, the training of a probe took between 15 min and 1.5h. Each model was trained on five different random seeds.

5 Results

Tab. 2 summarizes the highest values from all layers. We report our results using UUAS and DSpr. where higher values indicate a larger percentage of correctly reconstructed edges and a higher correlation between the predicted and gold-distances, respectively. Each value is the mean of the five runs. It is visible that almost all adapted models improve over their natural language baselines. The highest performance overall is demonstrated by AnReu/math_pretrained_bert. Only the performance of MathBERT-custom drops in comparison to bert-base-cased. The DSpr. scores of the best models in comparison to their baselines are visualized in Fig. 2.

In general, the models pre-trained on ARQMath demonstrated a better performance across both metrics compared to models pre-trained on other data sets. A possible reason could be that this data set contains a large variety of formulas written in LaTeX while this is unclear for the other data sets since they are not publicly available. We validated these results also using a second OPT data set based on the MATH data set (Hendrycks et al., 2021), which contains formulas written in LaTeXextracted from competition math problems. Since there was no drop in performance among the models pre-trained on ARQMath, we can conclude that models did not benefit from the overlap between the pre-training data and the probing formulas.

BERT and RoBERTa-based models show that the best extractability for Operator Trees lies in the middle layers, between layer 4 and 7 for base models and between layer 9 and 13 for large models. This pattern is consistent with the results reported by Hewitt and Manning for dependency structures. Notably, the same pattern does not emerge for ALBERT and AnReu/math_albert. Here, the highest scores are in layers 2 and 3. Overall, the scores for both ALBERT-based models are significantly lower, even after training on ARQMath. Interestingly, this model was among the best for the ARQMath Lab 3 on Mathematical Answer Retrieval and outperformed also AnReu/-

Model	DSpr.	UUAS
albert-base-v2	0.631 (3)	0.477 (3)
AnReu/math_albert	0.680 (2)	0.513 (2)
bert-base-cased	0.713 (7)	0.532 (6)
allenai/ scibert-scivocab-cased	0.727 (7)	0.545 (7)
AnReu/ math_pretrained_bert	**0.815** (7)	**0.700** (6)
tbs17/MathBERT	0.718 (6)	0.550 (5)
tbs17/ MathBERT-custom	0.686 (5)	0.530 (5)
roberta-base	0.703 (5)	0.526 (5)
roberta-large	0.706 (9)	0.536 (13)
AnReu/ math_pretrained_roberta	0.746 (5)	0.576 (5)
shauryr/ arqmath-roberta-base	0.715 (5)	0.541 (4)
uf-aice-lab/ math-roberta	0.711 (9)	0.547 (11)
witiko/mathberta	0.752 (5)	0.574 (5)

Table 2: Results of reconstruction of OPTs using UUAS and DSpr., displaying only the best results across all layers, best layer indicated by (layer number).

math_pretrained_roberta, which is the second best model for UUAS in this study. A similar mismatch between the performance in downstream natural language tasks and syntactic parsing was also found by (Glavaš and Vulić, 2021). Therefore, this finding casts a doubt on whether the models rely on their OPT knowledge when solving the downstream task of Mathematical Answer Retrieval. However, the limitations of probing classifiers as the one used in this work do not allow conclusions about the models usage of the knowledge. Hence, further research in this direction is required to investigate whether and how these models use structural knowledge during downstream tasks. In addition, Appendix A shows examples of reconstructed Operator Trees, while Appendix B contains the mean scores and standard deviation for each model in each layer.

6 Conclusion

This work aims to answer the question: Are Operator Trees extractable from the models' contextualized embeddings? We trained a structural probe that learns to approximate the distances between nodes in the trees. The results show that models

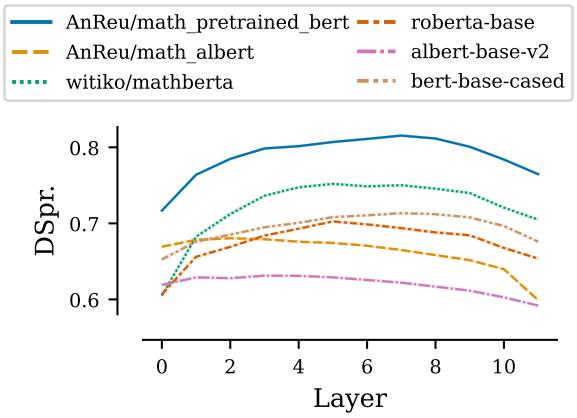

Figure 2: Results of reconstruction of OPTs across layers of the best models and all baselines, results for DSpr., the same pattern emerges for UUAS.

(further) pre-trained on mathematical data sets outperform their natural language baselines. The high correlation of the trained probe suggests that the models indeed encode useful information about Operator Trees in their contextualized embeddings. Given that the models have never been trained on Operator Trees, but only using masked-language modeling on string-based representation such as LaTeX, explicitly proving Operator Trees during a downstream task such as Mathematical Retrieval might not even be necessary.

Furthermore, we notice differences between model classes: While BERT and RoBERTa-based models demonstrate a higher extractability for the structural probe, both ALBERT-based models fall behind. In contrast, their performance on mathematical answer retrieval is on par with the other evaluated models. Further research is required to investigate this issue. We are open to offer other researchers the re-use of our work by making our source code and data set fully publicly available on GitHub[7].

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful feedback and comments. This work was supported by the DFG under Germany's Excellence Strategy, Grant No. EXC-2068-390729961, Cluster of Excellence "Physics of Life" of TU Dresden. Furthermore, the authors are grateful for the GWK support for funding this project by providing computing time through the Center for Information Services and HPC (ZIH) at TU Dresden.

[7] https://github.com/AnReu/extracting-opts

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 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.

Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. 2020. Pretrained language model embryology: The birth of albert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6813–6828.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 8594–8603.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, et al. 2020. Codebert: A pre-trained model for programming and natural languages. *arXiv preprint arXiv:2002.08155*.

Martin Geletka, Vojtěch Kalivoda, Michal Štefánik, Marek Toma, and Petr Sojka. 2022. Diverse semantics representation is king.

Goran Glavaš and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Hwiyeol Jo, Dongyeop Kang, Andrew Head, and Marti A Hearst. 2021. Modeling mathematical notation semantics in academic papers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3102–3115.

Behrooz Mansouri, Anurag Agarwal, Douglas Oard, and Richard Zanibbi. 2020. Finding old answers to new math questions: the arqmath lab at clef 2020. In *European Conference on Information Retrieval*, pages 564–571. Springer.

Behrooz Mansouri, Vít Novotnỳ, Anurag Agarwal, Douglas W Oard, and Richard Zanibbi. 2022. Overview of arqmath-3 (2022): Third clef lab on answer retrieval for questions on math (working notes version). *Working Notes of CLEF*.

Behrooz Mansouri, Shaurya Rohatgi, Douglas W Oard, Jian Wu, C Lee Giles, and Richard Zanibbi. 2019. Tangent-cft: An embedding model for mathematical formulas. In *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*, pages 11–18.

Vít Novotnỳ and Michal Štefánik. 2022. Combining sparse and dense information retrieval. *Proceedings of the Working Notes of CLEF*.

Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. Mathbert: A pre-trained model for mathematical formula understanding. *arXiv preprint arXiv:2105.00377*.

Anja Reusch, Maik Thiele, and Wolfgang Lehner. 2022a. Transformer-encoder and decoder models for questions on math. *Proceedings of the Working Notes of CLEF 2022*, pages 5–8.

Anja Reusch, Maik Thiele, and Wolfgang Lehner. 2022b. Transformer-encoder-based mathematical information retrieval. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 175–189. Springer.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Shaurya Rohatgi, Jian Wu, and C Lee Giles. 2020. Psu at clef-2020 arqmath track: Unsupervised re-ranking using pretraining. In *CLEF (Working Notes)*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Wei Zhong, Shaurya Rohatgi, Jian Wu, C Lee Giles, and Richard Zanibbi. 2020. Accelerating substructure similarity search for formula retrieval. *Advances in Information Retrieval*, 12035:714.

Wei Zhong, Yuqing Xie, and Jimmy Lin. 2022. Applying structural and dense semantic matching for the arqmath lab 2022, clef. *Proceedings of the Working Notes of CLEF 2022*, pages 5–8.

Wei Zhong and Richard Zanibbi. 2019. Structural similarity search for formulas using leaf-root paths in operator subtrees. In *European Conference on Information Retrieval*, pages 116–129. Springer.

A Examples of Reconstructed Operator Trees

Figure 3: Operator Trees calculated from the predicted squared distances between the tokens. The black edges above each formula are the gold edges from the OPT parser, while the red edges are the predicted ones by each model, taken from one seed of the best layer by DSpr. In a large majority of cases the models correctly identified the edges of the displayed formula. Most differences can be seen from the second part of the left hand side of the equation, where the models mostly struggle with the parent-child relationships of the equal sign.

B Results for all layers

	bert-base-cased		tbs17/MathBERT		tbs17/MathBERT-custom	
	mean	stdev	mean	stdev	mean	stdev
0	0.6525	0.00126	0.6455	0.00040	0.6468	0.00058
1	0.6759	0.00116	0.6481	0.00110	0.6492	0.00035
2	0.6851	0.00075	0.6540	0.00084	0.6530	0.00078
3	0.6949	0.00078	0.6834	0.00047	0.6785	0.00080
4	0.7008	0.00045	0.7047	0.00083	0.6824	0.00049
5	0.7082	0.00027	0.7145	0.00036	0.6863	0.00073
6	0.7106	0.00019	0.7175	0.00036	0.6832	0.00009
7	0.7134	0.00037	0.7092	0.00044	0.6772	0.00024
8	0.7121	0.00042	0.6987	0.00026	0.6703	0.00016
9	0.7079	0.00033	0.6842	0.00028	0.6609	0.00027
10	0.6965	0.00013	0.6646	0.00021	0.6531	0.00031
11	0.6759	0.00046	0.6495	0.00026	0.6425	0.00031
	allenai/scibert_scivocab_cased		AnReu/math_pretrained_bert			
0	0.6578	0.00094	0.7167	0.00020		
1	0.6835	0.00122	0.7639	0.00032		
2	0.6962	0.00110	0.7848	0.00041		
3	0.7061	0.00033	0.7985	0.00030		
4	0.7200	0.00040	0.8015	0.00011		
5	0.7263	0.00046	0.8070	0.00017		
6	0.7267	0.00009	0.8110	0.00005		
7	0.7261	0.00016	**0.8154**	0.00012		
8	0.7150	0.00063	0.8116	0.00006		
9	0.6938	0.00008	0.8007	0.00018		
10	0.6792	0.00019	0.7839	0.00008		
11	0.6764	0.00031	0.7647	0.00010		

Table 3: DSpr. Results of BERT, BERT-based and similar models.

	bert-base-cased		tbs17/MathBERT		tbs17/MathBERT-custom	
	mean	stdev	mean	stdev	mean	stdev
0	0.4585	0.00169	0.4832	0.00113	0.4891	0.00053
1	0.4947	0.00107	0.4845	0.00129	0.4927	0.00061
2	0.5109	0.00068	0.4845	0.00059	0.4929	0.00043
3	0.5178	0.00027	0.5171	0.00028	0.5162	0.00049
4	0.5216	0.00059	0.5393	0.00051	0.5255	0.00074
5	0.5315	0.00064	0.5496	0.00037	0.5300	0.00059
6	0.5323	0.00019	0.5491	0.00039	0.5248	0.00044
7	0.5321	0.00053	0.5363	0.00038	0.5169	0.00050
8	0.5283	0.00040	0.5202	0.00043	0.5074	0.00042
9	0.5221	0.00051	0.5017	0.00018	0.4973	0.00045
10	0.5032	0.00018	0.4756	0.00022	0.4854	0.00038
11	0.4779	0.00035	0.4560	0.00028	0.4725	0.00034
	allenai/scibert_scivocab_cased		AnReu/math_pretrained_bert			
0	0.4655	0.00036	0.5458	0.00069		
1	0.4984	0.00150	0.6336	0.00054		
2	0.5151	0.00103	0.6686	0.00034		
3	0.5244	0.00037	0.6825	0.00030		
4	0.5363	0.00038	0.6790	0.00043		
5	0.5450	0.00038	0.6920	0.00032		
6	0.5421	0.00066	**0.7000**	0.00043		
7	0.5453	0.00018	0.6952	0.00041		
8	0.5308	0.00087	0.6852	0.00043		
9	0.5101	0.00036	0.6694	0.00046		
10	0.4925	0.00033	0.6415	0.00030		
11	0.4865	0.00038	0.6028	0.00046		

Table 4: UUAS Results of BERT, BERT-based and similar models.

	albert-base-v2		AnReu/math_albert	
	mean	stdev	mean	stdev
0	0.6192	0.00072	0.6693	0.00017
1	0.6290	0.00133	0.6783	0.00022
2	0.6279	0.00039	**0.6805**	0.00043
3	0.6312	0.00030	0.6791	0.00024
4	0.6310	0.00049	0.6759	0.00015
5	0.6291	0.00029	0.6743	0.00031
6	0.6255	0.00067	0.6706	0.00014
7	0.6221	0.00082	0.6649	0.00017
8	0.6168	0.00053	0.6583	0.00017
9	0.6115	0.00041	0.6516	0.00041
10	0.6028	0.00037	0.6397	0.00033
11	0.5919	0.00040	0.5991	0.00025

(a) DSpr. Results

	albert-base-v2		AnReu/math_albert	
	mean	stdev	mean	stdev
0	0.4620	0.00128	0.5095	0.00039
1	0.4727	0.00132	**0.5130**	0.00040
2	0.4746	0.00054	0.5125	0.00025
3	0.4771	0.00055	0.5127	0.00038
4	0.4742	0.00063	0.5090	0.00032
5	0.4747	0.00059	0.5062	0.00023
6	0.4699	0.00088	0.5030	0.00062
7	0.4652	0.00091	0.4975	0.00048
8	0.4563	0.00070	0.4891	0.00032
9	0.4470	0.00049	0.4811	0.00024
10	0.4343	0.00088	0.4671	0.00056
11	0.4144	0.00075	0.4082	0.00046

(b) UUAS Results

Figure 4: Results of ALBERT and math albert.

	roberta-base		shauryr/ arqmath-roberta-base		witiko/mathberta		AnReu/ math_pretrained_roberta	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev
0	0.6066	0.00085	0.6219	0.00054	0.6051	0.00024	0.6179	0.00036
1	0.6561	0.00022	0.6682	0.00048	0.6824	0.00065	0.6917	0.00038
2	0.6691	0.00039	0.6841	0.00049	0.7123	0.00117	0.7195	0.00057
3	0.6840	0.00044	0.7014	0.00085	0.7363	0.00043	0.7345	0.00038
4	0.6930	0.00027	0.7114	0.00045	0.7475	0.00021	0.7422	0.00027
5	0.7025	0.00026	0.7146	0.00027	**0.7519**	0.00065	0.7464	0.00030
6	0.6986	0.00053	0.7102	0.00019	0.7487	0.00069	0.7409	0.00005
7	0.6937	0.00067	0.7038	0.00015	0.7501	0.00022	0.7366	0.00041
8	0.6881	0.00093	0.6997	0.00027	0.7456	0.00035	0.7331	0.00020
9	0.6843	0.00058	0.6961	0.00018	0.7399	0.00015	0.7274	0.00027
10	0.6677	0.00061	0.6798	0.00050	0.7207	0.00020	0.7094	0.00026
11	0.6538	0.00036	0.6616	0.00031	0.7049	0.00034	0.6942	0.00024

Table 5: DSpr. Results of RoBERTa-base and small RoBERTA-based models.

	roberta-base		shauryr/ arqmath-roberta-base		witiko/mathberta		AnReu/ math_pretrained_roberta	
	mean	stdev	mean	stdev	mean	stdev	mean	stdev
0	0.4456	0.00127	0.4619	0.00044	0.4268	0.00078	0.4543	0.00068
1	0.4825	0.00042	0.5010	0.00083	0.5086	0.00068	0.5296	0.00084
2	0.4988	0.00053	0.5184	0.00053	0.5388	0.00065	0.5477	0.00095
3	0.5187	0.00015	0.5352	0.00059	0.5618	0.00065	0.5690	0.00063
4	0.5224	0.00042	0.5414	0.00061	0.5732	0.00027	0.5731	0.00043
5	0.5255	0.00041	0.5378	0.00017	**0.5742**	0.00084	0.5759	0.00026
6	0.5172	0.00053	0.5358	0.00026	0.5687	0.00057	0.5672	0.00024
7	0.5088	0.00035	0.5247	0.00059	0.5692	0.00038	0.5591	0.00029
8	0.5106	0.00033	0.5311	0.00061	0.5704	0.00055	0.5599	0.00026
9	0.5091	0.00057	0.5324	0.00016	0.5659	0.00025	0.5590	0.00041
10	0.4899	0.00032	0.5167	0.00033	0.5471	0.00043	0.5411	0.00027
11	0.4713	0.00051	0.4902	0.00030	0.5294	0.00035	0.5238	0.00037

Table 6: UUAS Results of RoBERTa-base and small RoBERTa-based models.

	roberta-large		uf-aice-lab/ math-roberta	
	mean	stdev	mean	stdev
0	0.61374	0.00029	0.61387	0.00051
1	0.62816	0.00057	0.62524	0.00068
2	0.65153	0.00077	0.65213	0.00080
3	0.66078	0.00116	0.66054	0.00058
4	0.67341	0.00050	0.67043	0.00083
5	0.68048	0.00111	0.68125	0.00034
6	0.68313	0.00044	0.68719	0.00052
7	0.68996	0.00057	0.69613	0.00054
8	0.69757	0.00072	0.70321	0.00068
9	0.70602	0.00058	**0.71079**	0.00046
10	0.70484	0.00029	0.70922	0.00043
11	0.70425	0.00061	0.70943	0.00068
12	0.70255	0.00106	0.70796	0.00041
13	0.70144	0.00057	0.70940	0.00042
14	0.69807	0.00035	0.70646	0.00044
15	0.69522	0.00046	0.70268	0.00048
16	0.69463	0.00012	0.70220	0.00032
17	0.69444	0.00023	0.70017	0.00025
18	0.68966	0.00009	0.69823	0.00025
19	0.68492	0.00029	0.69437	0.00014
20	0.68087	0.00036	0.69277	0.00027
21	0.67582	0.00035	0.69129	0.00012
22	0.66197	0.00051	0.68796	0.00057
23	0.64475	0.00056	0.68602	0.00082
24	0.62023	0.00042	0.66011	0.00017

(a) DSpr. Results

	roberta-large		uf-aice-lab/ math-roberta	
	mean	stdev	mean	stdev
0	0.44171	0.000279	0.44210	0.000778
1	0.45419	0.001066	0.45024	0.001192
2	0.48857	0.000532	0.48968	0.001104
3	0.50158	0.000647	0.50030	0.000336
4	0.52109	0.000171	0.51772	0.000599
5	0.51957	0.001377	0.52384	0.000602
6	0.51196	0.000350	0.52088	0.000902
7	0.52005	0.000604	0.52933	0.000500
8	0.52343	0.001005	0.53698	0.000465
9	0.53249	0.000754	0.54388	0.000493
10	0.53287	0.000664	0.54576	0.000411
11	0.53620	0.000540	**0.54715**	0.000436
12	0.53255	0.001094	0.54254	0.000349
13	0.53622	0.000277	0.54502	0.000427
14	0.52932	0.000924	0.53911	0.000366
15	0.52427	0.000548	0.53527	0.000632
16	0.52411	0.000432	0.53579	0.000367
17	0.52099	0.000341	0.53276	0.000112
18	0.51146	0.000397	0.52748	0.000328
19	0.50598	0.000293	0.52150	0.000432
20	0.50340	0.000510	0.52375	0.000435
21	0.49970	0.000665	0.52158	0.000291
22	0.48542	0.000576	0.52027	0.000370
23	0.47364	0.000890	0.52066	0.000748
24	0.44753	0.000611	0.49027	0.000578

(b) UUAS Results

Figure 5: Results of RoBERTa-large and uf-aice-lab/math-roberta.