Classifying Mathematical Information with Natural Language Models

Anonymous ACL submission

Abstract

This study evaluates the performance of natural language models in understanding and categorizing mathematical statements, such as proofs and theorems, into their respective categories to see if they are able to deeply understand mathematical logic and context. Using an adapted version of an existing dataset, we trained Distilbert and Distilgpt2 on 9 overlapping categories and performed a binary text classification. Despite the model's high accuracy overall, variation in its scores by category lead us to believe the models are simply looking for text patterns to classify information and are not able to perform reasoning in a deeper mathematical context. These findings highlight the need for improved logical reasoning in language models to enhance AI/automated mathematical problem-solving.

1 Introduction

We are interested in assessing existing language models' abilities to understand abstract ideas and patterns for logical reasoning. More specifically, we hope to illuminate whether or not language models are capable of deeply understanding mathematics. Mathematics differs from natural language because of its unique structure and usage of single-character symbols that are often repeatedly used. Additionally, mathematical symbols and variables can have different meanings in different contexts more than words in other spoken natural languages. We thought that the task of evaluating a model's understanding of language in a mathematical context would be novel and interesting due to these nuances and would give us more insight into the current capabilities and limitations of existing language models.

In this study, we investigated whether or not language models can classify mathematical information such as proofs and theorems into categories such as Geometry and Algebra. For instance, given an equation with a series, a model will likely classify it into Calculus. However, will a model be able to also identify if it falls into Discrete Mathematics? While this question has intricacies, we do believe models will be able to quite accurately categorize these mathematical proofs and theorems. However, other literature has shown most models do a lackluster job of understanding mathematics in other contexts (Ferreira and Freitas, 2020). We hope our results inform researchers within both computer science and mathematics on how reliable mathematical output is from our current language models and where improvements in logical reasoning are still needed.

In our study, we conducted a modified method of text classification in order to be able to assess language models's pre-trained performance on multi-category classification. Much of the existing literature on this topic highlights both the lack of previous research on this topic as well as any promising or positive results, showing a clear demand for more investigation. In fact, quite a few existing studies primarily focused on building a dataset itself for further research (Welleck et al., 2021). We created our dataset using one from a previous study for our experiment, consisting of mathematical proofs and theorems with labeled categories. We trained our models Distilbert (HuggingFace, 2019) and Distilgpt2 (HuggingFace, 2020) on data from each of these categories and then evaluated their text classification performances on our test data. We calculated average accuracy scores and compared them across categories and models.

While our results showed that both models generally performed well at this task with pre-training, they had significantly lower accuracy in a few categories. This leads us to believe that despite the model's good performance on average, it does not in fact have a deeper understanding of mathematical logic. We suspect that the poor performance in categories such as Algebra was due to the fact that

said topics have large overlaps with other areas of mathematics. We suggest that the models are looking for recognizable patterns of terms, symbols, and notation that are characteristic to each mathematical category, instead of grasping the deeper logic behind the application of these proofs and theorems. From the results of our study and the lack of existing research on this topic, it is clear that much improvement is needed in the development of our language models in logical reasoning, especially with mathematical information. Classifying information is a key component of information retrieval, which is an essential component of AI model and natural language model capabilities. This study aims to contribute to existing research on this extremely limited field within language models that works towards improving the automation of mathematical problem solving, particularly with solving proofs.

2 Background

We looked at other previous research specifically targeted toward using language models on mathematical information and statements for a variety of tasks. One glaring theme across the sources we read was the general lack of research in this field. Two of the main relevant studies we looked at by Welleck et al. (2021) and Ferreira and Freitas (2020) mainly focused on the mere creation of an adequate, robust dataset that can be utilized for further related studies. These studies mention specifically that the scarcity of useful datasets is a clear result of the scarcity of existing research in this area of natural language processing and language models. This prompted us to use an adaptation of Welleck et al. (2021)'s dataset for our study. Aside from datasets, the performance of existing language models at similar tasks is not very good. Welleck et al. (2021)'s study uses its created dataset to compare the performance of the BERT model to the TF-IDF benchmark on two tasks: mathematical reference retrieval, which involves retrieving a set of references for a given theorem, and mathematical reference generation, which essentially involves generating an ordered proof. BERT's poor performance at the zero-shot parts of the experiment shows that the model is far from able to understand mathematical concepts or reasoning. Our study specifically lends itself to this process as well, as part of discerning what sorts of references are useful for a given proof likely involves categorizing information to eliminate irrelevant options. Ferreira and Freitas (2020)'s study similarly looked at BERT and SciBERT's performance on the task of natural proof selection, which is identifying mathematical statements that support proofs. This task works towards the process of automated theorem-proving. Again, while the models mildly outperformed their benchmark measures, they were largely unsuccessful at the given task. Because of the two shared themes in these two studies, the lack of adequate datasets, and the poor performance of existing models on tasks dealing with mathematical information, we propose that this is a relatively uncharted topic to explore in our study.

3 Methods

We adapted an existing dataset of mathematical statements– proofs and theorems– from previously conducted research on a similar topic. We used several Python scripts to re-format their data into a compatible format to be used with the NLP Scholar toolkit (Prasad and Davis, 2024).[1] From the existing dataset, each statement had a set of prelabeled categorical labels. The majority of entries had more than one label, meaning that a statement may appear in numerous categories. Using a single text classification model would involve giving a model a set of categories and having it output a singular category label prediction for each entry in the dataset. However, because the categories are not mutually exclusive, we wanted to see if the model would be able to correctly predict all of a given statement's categories. Therefore, instead of training and evaluating our data with a single model, we trained multiple models, one for each category, and evaluated our singular test dataset using all of the models. Each model would instead perform a binary 'Positive'/'Negative' classification, predicting whether or not the given entry fell under the given category. Using this method, we were able to account for the overlap between category labels for a single statement. After extracting and correctly formatting our data into train, validation, and test files, we trained each model on its respective train and validation datasets. Again, due to the way we are conducting our classification, we had to generate numerous copies of our test dataset with the correct 'Positive' and 'Negative' target labels for

[1] Datasets and code used to generate data available on our GitHub repository

the given category while keeping the content of the data the same. We then performed all evaluations with each model on the test data and calculated accuracy scores for comparison and analysis.

3.1 Models

Many of the studies in the existing research we looked at used the BERT model (Ferreira and Freitas, 2020). While most of these studies showed relatively poor performance of this model on their tasks, we thought it is worth using similar models in our experiment for comparison purposes. The two models used in our experiment were Distilgpt2 (HuggingFace, 2020) and Distilbert-base-cased (HuggingFace, 2019). We opted for the distilled versions of GPT2 and BERT models for computational efficiency and to contrast previous literature such as Welleck et al. (2021). We chose one unidirectional model and one bidirectional model respectively, suspecting that the unidirectional models would perform worse compared to bidirectional models due to the two-direction context nature of the latter. For Distilbert, we chose the cased version of this model because of regular distinctions between upper and lowercase letters in mathematical notation. As noted above, we trained one version of each model for each category. We had a total of 3 models for Distilgpt2 and 9 for Distilbert-base-cased.

3.2 Datasets

We extracted our data from the NATURAL-PROOFS dataset from another study (Welleck et al., 2021). This existing data set was in a large JSON file with mathematical statements organized with proofs and theorems separated. Each entry contained the content of the actual multi-line mathematical statement in an array and other metadata such as the mathematical category labels. We first extracted the mathematical statements and reformatted them into single-line entries. We extracted the set of proofs and theorems organized into 9 non-mutually exclusive categories, as a mathematical statement can have multiple category labels. These categories were as follows: Calculus, Discrete Mathematics, Abstract Algebra, Algebra, Logic, Topology, Geometry, Linear Algebra, and Number Theory. After sorting, we used an 80/10/10 split for each category's train, validation, and test data.

3.2.1 Train and Validate Data

For each category, we used an 80/10/10 split to obtain the 'Positive" labeled data, or in other words, the data that is part of the given category. To obtain 'Negative' data, we then sampled an equal number of negative examples between the other categories for training and validation. Because of overlapping entries in numerous categories, we ensured that there were no duplicate entries with different labels in datasets for each category. Our training and validation data was structured as two separate text files, with each entry on a new line of the file. Each line contained a mathematical statement followed by a tab character and then either a 1 or 0 indicating whether it was a positive or negative entry respectively.

3.2.2 Evaluation Data

After splitting our data to obtain our test data, we aggregated it into a single TSV file with 4 columns: "textid", "text", "condition", and "target". The "textid" column was simply the rows numbered in ascending order starting with 1, while "text" contained the actual data. The "condition" column noted the category that the given text fell under for organizational purposes. Each row in the "target" column contained either 'Positive' or 'Negative', representing the correct target prediction for the model. Again, because we are running binary classification with multiple models, we created one copy of the test data TSV file for each category with only different positive 'target' labels corresponding to the given category for the binary classification. All other elements of the file were kept the same.

3.3 Evaluation

We used a Python script to perform a simple accuracy score of successful predictions out of the total predictions performed for each category and then compared these scores across the different categories. For our available data on the Distilgpt2 model, we compared the accuracy scores to that of Distilbert.

4 Results

Table 1 gives our obtained results for the Distilbert model. The average accuracy score for Distilbert was 92%. Table 2 gives our results for Distilgpt2. The average accuracy score for Distilgpt2 was 90%.

Distilbert outperformed Distilgpt2 across the board on the three categories we were able to obtain

Category	Accuracy
Calculus	0.984
Discrete Mathematics	0.957
Abstract Algebra	0.826
Algebra	0.804
Logic	0.977
Topology	0.972
Geometry	0.952
Linear Algebra	0.949
Number Theory	0.841

Table 1: Accuracy scores per category for Distilbert-base-cased

Category	Accuracy
Discrete Mathematics	0.913
Logic	0.951
Linear Algebra	0.835

Table 2: Accuracy scores per category for Distilgpt2

data for. This is consistent with our prediction that the bidirectional model would fare better at this task than the unidirectional model because they are able to obtain two-directional context. In general, both models performed pretty well on this task, as every single category had an accuracy rate above 80%. This is somewhat surprising considering findings in our background research that mostly indicated that existing models tend to struggle with other similar but arguably more complicated tasks, but consistent with Welleck et al. (2021)'s findings that the pre-trained models performed well on test data within the same domain as their training data. What is most notable and interesting about the results in Table 1 is which categories performed the most poorly: Abstract Algebra, Algebra, and Number Theory for Distilbert. For Number Theory, we are not sure as to what caused this discrepancy. However, for the rest of our results, our proposed explanation for this occurrence has to do with the nature of overlapping with mathematical categories themselves within mathematics. Some categories in our data are quite distinct in nature such as Logic or Calculus and are likely identifiable by certain markers. For example, Logic uses a specific set of frequently recurring symbols and phrases such as "exclusive or" or "modus tollens" that are extremely unique to this topic alone. A model would likely be able to easily classify a mathematical statement as a Calculus statement by virtue of it including an integral or derivative. In contrast, Algebra itself is foundational to all mathematics and is used within many other categories. Algebra is essential to calculations such as finding volumes and areas of objects in Geometry, solving integrals or derivatives in Calculus, and recursion in Discrete Mathematics. Consequently, we suspect that this dynamic of one-way overlapping may be the explanation for why Algebra specifically significantly underperformed. We can now reason why certain subjects have high accuracy: If a subject has a high score, it is likely that it has distinct notation which implies a recognizable pattern. This leads us to the conclusion that the models are not able to perform a deeper level of reasoning above simple pattern recognition, at least within the context of mathematical statements. If the models were able to do so, they would have better recognized the nuances between the higher-level correct subject and core foundational mathematical concepts that appear in many different areas.

5 Discussion

This study sought to evaluate the ability of language models to classify mathematical information into its respective categories. Mathematics is a unique task due to its symbolic structure, contextual variability, and the overlapping nature of its categories. We used an adaptation of the NATURALPROOFS dataset for an adapted version of multi-label text classification (Welleck et al., 2021). We trained our models and evaluated our re-formatted data using the NLP Scholar toolkit (Prasad and Davis, 2024). Performance for both models was generally high but inconsistent among categories, leading us to believe that models are still unable to grasp mathematical concepts at a deeper level.

One limitation of our study was the platform used to train and evaluate these models. We were not able to obtain trained models for 6 out of the 9 categories for our unidirectional model due to unknown issues with the platform used to train our models. This is despite the fact that our data and configuration files were identical to those of the models that were able to successfully run train and evaluate. The results for our unidirectional model are less significant because we mainly ran them for brief comparison, and the results we were able to obtain seem to support our plausible assumption that it would perform worse than our bidirectional

model, but for further investigation seeing the full results would be a good confirmation.

For future directions, it would be interesting to run this experiment using a variation of models such as the non-distilled versions of GPT2 and BERT, or a model specialized in reading mathematics. If our results are consistent, this would allow us to potentially determine if our supposed explanation for the variation in our accuracy scores is consistent with a model that inherently understands math better than our normal language models. Additionally, our study focused on a smaller subset of categories that were included in the original dataset we looked at due to the sheer size of it. It would be worth replicating with more categories and the necessary computing power and time.

References

Deborah Ferreira and Andre Freitas. 2020. Natural language premise selection: Finding supporting statements for mathematical text. *arXiv preprint arXiv:2004.14959*.

HuggingFace. 2019. Distilbert base cased. https://huggingface.co/distilbert/distilbert-base-cased. Accessed: 2024-12-15.

HuggingFace. 2020. Distilgpt-2. https://huggingface.co/distilbert/distilgpt2. Accessed: 2024-12-15.

Grusha Prasad and Forrest Davis. 2024. Training an NLP scholar at a small liberal arts college: A backwards designed course proposal. In *Proceedings of the Sixth Workshop on Teaching NLP*, pages 105–118, Bangkok, Thailand. Association for Computational Linguistics.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalproofs: Mathematical theorem proving in natural language. *arXiv preprint arXiv:2104.01112*.