Modeling Mathematical Notation Semantics in Academic Papers

Presented at EMNLP2021(Findings)

Hwiyeol Jo, Dongyeop Kang, Andrew Head, & Marti A. Hearst

NAVER Clova, Univ. Minnesota, and UC Berkeley

hwiyeolj@gmail.com

Introduction

Introduction

- Despite the great advances in pretrained language models, they are still unable to analyze mathematical notation reliably
  - Our preliminary analysis shows the Pretrained models show very poor performance (9%) while N-gram based language model shows 19%
  - Fine-tuning on the task increases the performance (48%) but the accuracy is still inappropriate for an application

Introduction

Contribution

- We propose two notation prediction tasks to test models' mathematical semantics understanding
  - Notation auto-suggestions
  - Notation consistency checks
- We then present a fine-tuned model MathPredictor
  - Showing +12% and +16% for the tasks, respectively

Related Work

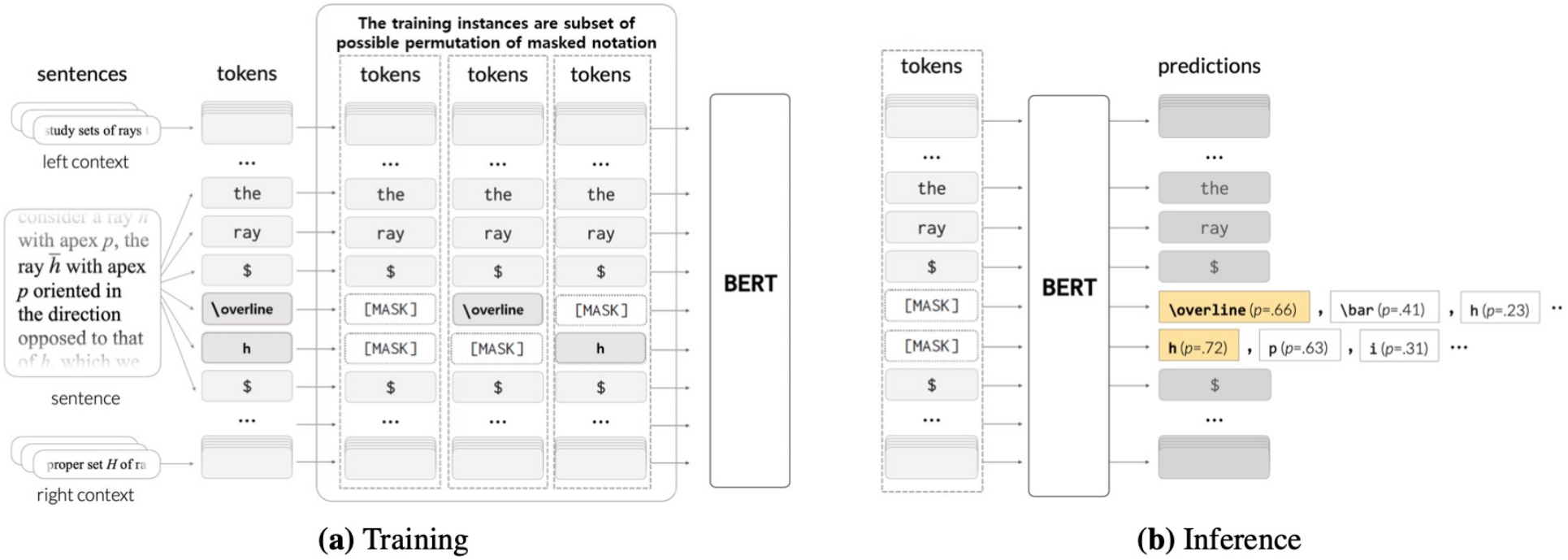
The use of mathematical notation in texts

Modeling mathematical notation

- Type inference in mathematical statements (Rabe et al., 2020)
- Topic-sensitive equation generation (Yasunaga and Lafferty, 2019)
- Superscript disambiguation (Youssef and Miller, 2018)
- Mathematical information retrieval (Greiner-Petter et al., 2020)
- Retrieve a relevant paragraph (Abekawa and Aizawa, 2016)
- Extract symbol description (Alexeeva et al., 2020)
- Symbol description detection (Madisetty et al., 2020)
- Contemporaneous work : Peng et al. (2021)

Proposed Method: MathPredictor

Model Architecture



(a) Training **(b)** Inference

Figure 2: The illustration of the proposed method. It encodes context (left and/or right sentences) and the target sentence where each token of notation in the target sentence is masked (`[MASK]`). At training, we permute the input sequences (dotted boxes) with random probability p in order to learn the structure of notation and then train BERT by using these representations of the sentences. As a result, the training instances are subset of the permutations. At inference time, the masked token is predicted with likelihood scores.

Proposed Method: MathPredictor

Notation Type Definition

Types	Examples	#-Uniq
Letter	n, m, SHA, model, loss, x ..	16K
Number	218, 00, 4k, 2K, 90., 2cm ..	234
OP&Symbol	α, θ, $\leq$, $\times$, arccos, %, $\exists$ ..	271
LaTeX Macros	\top, \text, \mathcal \quad, \bf, \rm, \underline, \em ..	562

Table 1: Examples of notation tokens. We report the unique number of notation tokens in the training data. #-Uniq means the unique number of notation tokens.

Details

- Mask Permutation
- BERT as a base model
- Length constraint of Notation is 10
- Larger context modeling
  - Global context / Local context

Experiment and Evaluation

Dataset

- S2ORC Dataset
  - 12.7 million full text in LaTeX format
- We randomly subsample 1,000 papers, which is tokenized by WordPiece tokenizer
  - Assign 80% to train, 10% to validation, and 10% to test

- Non-text entities are replaced
  - Author et al. to CITATION
  - Section N to SECTION
  - long equations to EQUATION
  - tables and figures to TABLE and FIGURE, respectively

Experiment and Evaluation

Research Question

- R1: How does our model compare to the baselines for the two prediction tasks?
- R2: Does the model simply memorize notations in context or does it learn domain conventional patterns from other papers?
- R3: Which types of notations is the model most able to predict?
- R4: How well does the model perform when evaluated at the sentence level?
- R5: How well does the model perform at the document-level (qualitatively)?

Experiment and Evaluation

Overall Performance

- Two tasks
  - Suggestion: Left-Only
  - Consistency: Left-and-Right

	Suggestion			Consistency		
	Top1	Top5	MRR	Top1	Top5	MRR
Random	3.3	14.1	-	3.6	15.1	-
4-gram	18.8	28.5	-	-	-	-
BERT	9.0	18.8	.146	13.8	28.3	.215
BERT(FT)	48.3	66.1	.568	57.8	75.4	.658
SciBERT	15.19	26.2	.207	16.6	26.6	.216
SciBERT(FT)	48.8	68.8	.579	58.6	76.7	.669
RoBERTa	0.5	1.5	.011	1.7	3.6	.029
RoBERTa(FT)	21.9	33.1	.277	32.8	45.8	.393
Ours	57.4	65.4	.613	71.7	77.7	.746
Ours(FT)	**60.5**	**71.3**	**.657**	**73.5**	**80.0**	**.767**
w/ FullContext	55.7	68.7	.620	72.2	79.8	.758

Table 2: Performance comparison on notation auto-suggestion and consistency checking tasks. FT means fine-tuning the model through masked language modeling on notations and words using our dataset. w/ FullContext means using full global context with ours (MATHPREDICTOR).

Experiment and Evaluation

Task-level Performance

- According to Difficulty
  - Easy set: The symbol(notation) is included in the context
  - Challenge set: not included in the context

Easy	··· We use a ring dimension $n = 8192$ with two plain text moduli $t^{(j)}$. Each coefficient modulus $n = 8192$ is decomposed into four 64-bit moduli for efficient use of FV-RNS.
Challenge	··· In scoring boardgames like Scrabble, swing, a state transition of advantage during the game progress is considered as successful shoot, and game length as attempt respectively. Let S and N be the average number of swings and the game length, respectively.

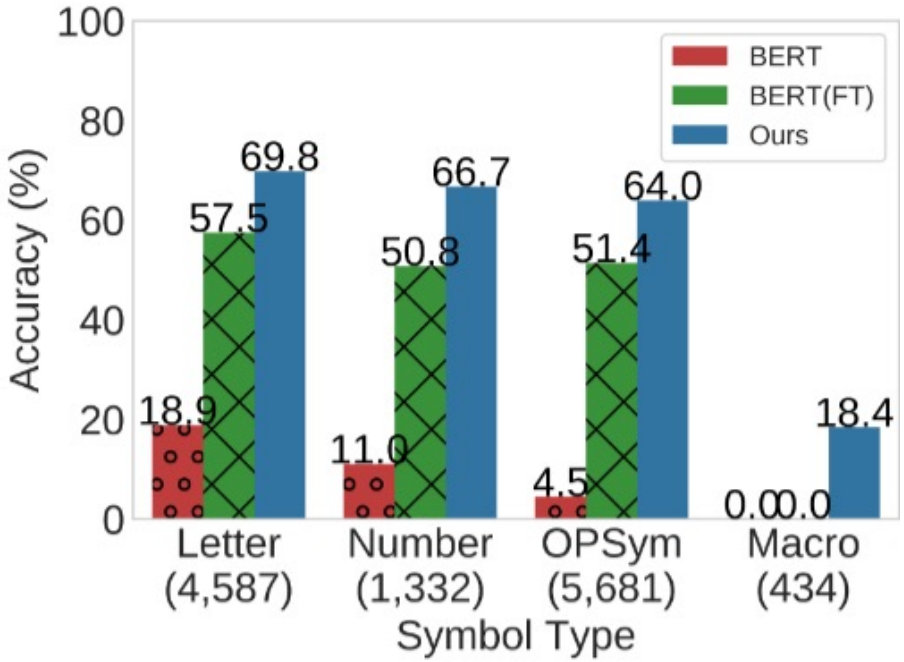
	Suggestion		Consistency	
	Easy	Challenge	Easy	Challenge
BERT	9.99	0.26	15.09	0.12
BERT(FT)	52.32	3.38	59.27	3.44
Ours	**66.97**	**7.62**	**77.72**	**6.38**
#-samples	12,364	1,511	12,382	826

Table 4: Top-1 accuracy of notation auto-suggestion and consistency checks on the `easy` set and `challenge` set. Note that the total sum of samples are different due to the different window sliding.

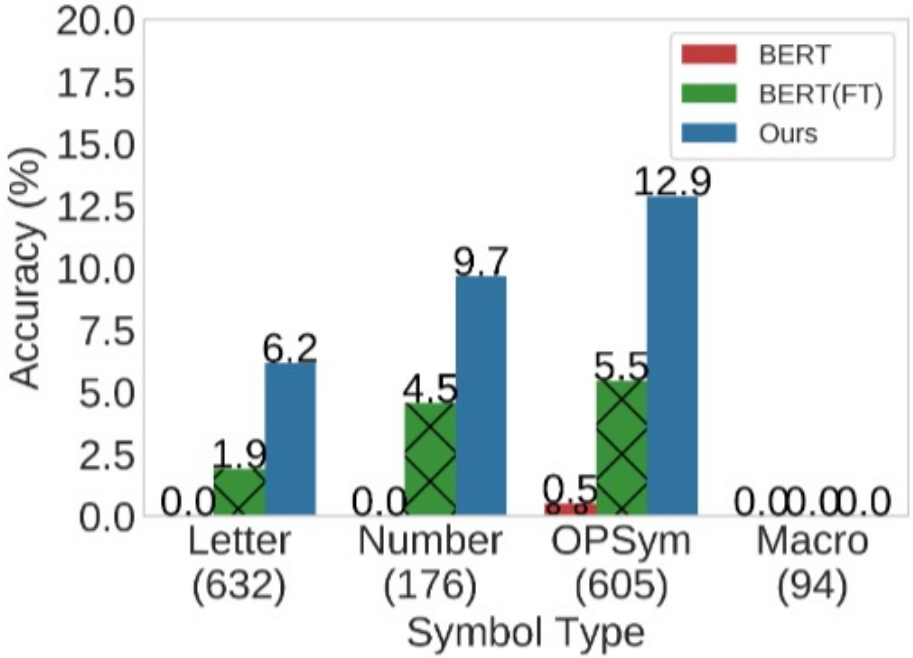
Experiment and Evaluation

Notation Type Performance

- Auto-suggestion



(a) Easy set

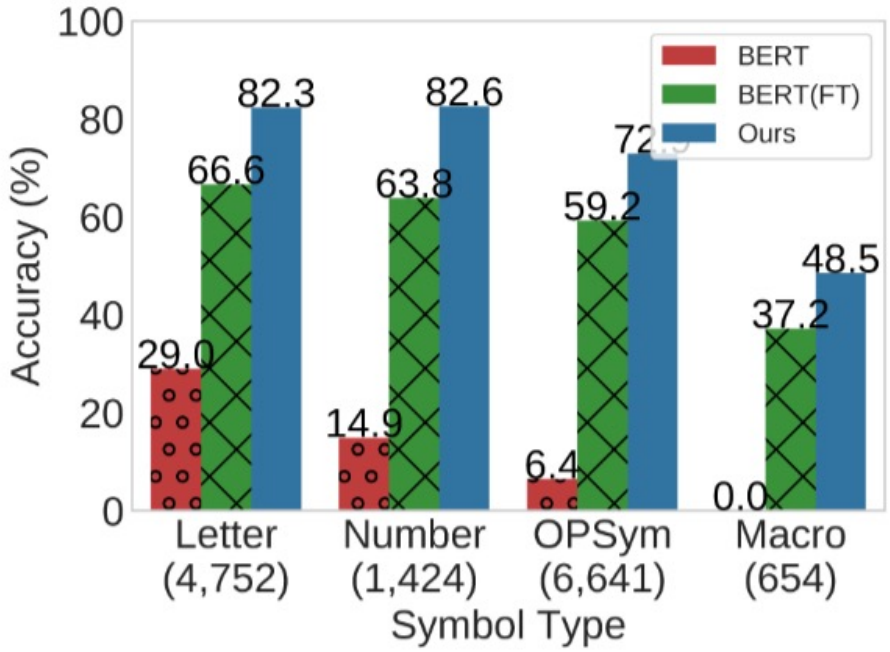


(b) Challenge set

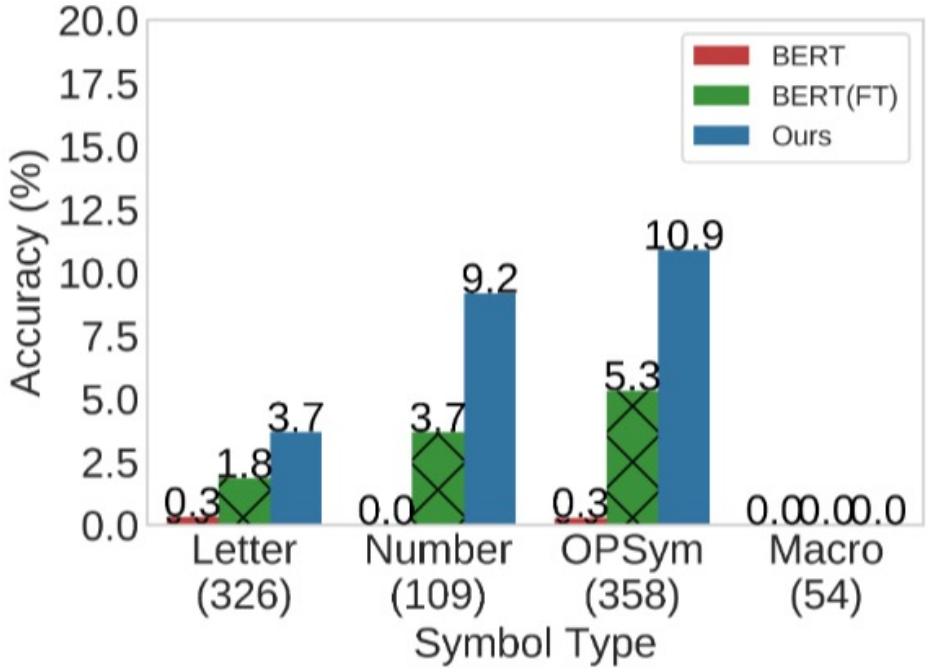
Experiment and Evaluation

Notation Type Performance

- Consistency-checks



(a) Easy set



(b) Challenge set

Experiment and Evaluation

Notation- and Sentence-level Performance

	Suggestion		Consistency	
	Notation	Sent.	Notation	Sent.
BERT	12.05	6.44	18.64	10.80
BERT(FT)	37.87	28.23	45.41	33.72
SciBERT(FT)	40.57	30.70	50.80	40.04
Ours	**45.11**	**37.11**	**57.20**	**48.56**
#-samples	5,672	2,888	4,711	2,769

Table 5: The comparison of notation-level and sentence-level top-1 accuracy in both tasks. The total number of tokens can be different because of predefined vocabularies in tokenizer.

Multi-tokens in notation	
..Thus, the procedure is worth trying in this range of [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK]..	
Gold:	\hat { p } _ { com ##m } ; $\hat{p}_{comm}$
BERT:	$ $ $ $ $ $ $ $
BERT(FT):	\hat { 1 } , , $ $ }
Ours:	**\hat { p } _ { com ##m }**
Multi-notations in sentence	
..that is they earn [MASK] [MASK] [MASK]/hour for class-1 VMs, [MASK] [MASK] [MASK]/hour for class-2VMs, and [MASK] [MASK] [MASK]/hour for class-3 VMs..	
Gold:	0 . 08 / 0 . 16 / 0 . 32
BERT:	$ $ $ / $ $ $ / $ $ =
BERT(FT):	**0 . 08 / 0 .** 10 / **0 .** 08
Ours:	**0 . 08 / 0 = 16 / 0 . 32**

Table 6: Example of notation-level and sentence-level predictions. Correctly predicted tokens are shown in bold blue, and incorrectly predicted tokens are in red. our method shows better performance than the baselines, but fails to predict the notation tokens perfectly.

Experiment and Evaluation

Full Paper (Paragraph) Result

PaperID in S2ORC dataset: 16122894, ArXivID: 1408.3083, Section: Computational Complexity of Binarization Scheme	
BERT:	.. Suppose, the length of input data is , (Gold: N), , (Gold: m) is the number of source symbols, and , (Gold: Y) is the source. For the first symbol, the length of the binary string would be M (Gold: N). The length of binary string for the second symbol would be the length of all the symbols, except the first symbol (see Table 1). Likewise, the length of n (Gold: N) binary string would be the length all symbols yet to be binarized. Mathematically, the length can be written as N, where m is the probability of m (Gold: Y) symbol. The total number of binary assignment would be N ...
BERT(FT):	... Suppose, the length of input data is m (Gold: N), m is the number of source symbols, and n (Gold: Y) is the source. For the first symbol, the length of the binary string would be N. The length of binary string for the second symbol would be the length of all the symbols, except the first symbol (see Table 1). Likewise, the length of N binary string would be the length all symbols yet to be binarized. Mathematically, the length can be written as N, where N (Gold: m) is the probability of m (Gold: Y) symbol. The total number of binary assignment would be N ...
MathPred(FT):	... Suppose, the length of input data is m (Gold: N), n (Gold: m) is the number of source symbols, and m (Gold: Y) is the source. For the first symbol, the length of the binary string would be N. The length of binary string for the second symbol would be the length of all the symbols, except the first symbol (see Table 1). Likewise, the length of N binary string would be the length all symbols yet to be binarized. Mathematically, the length can be written as N, where m is the probability of Y symbol. The total number of binary assignment would be N ...

Table 7: Example of paper-level predictions by MATHPREDICTOR and other baselines. We sequentially auto-suggest notations (left-only context) and concatenate the results. The notation tokens with gray background are the target. Blue colored notation tokens mean correct predictions and red colored notation tokens mean incorrect predictions. The gold labels (tokens) for the incorrect predictions are shown in parentheses.

Discussion

MathPredictor

- The performance is not likely good enough
  - Top-5 Accuracy is 71.3% and 80.0%
  - However, when we sub-sample 10x more
    The performance improved by +10% accuracy
    - Suggestion: 70.9% (Top-1) / 81.6% (Top-5)
    - Checking: 83.5% / 89.0%
- Current models memorize the meanings rather than generalize over them
  - Predicting notation is a challenging problem

Guidance for future work

- Utilize the structure of notation
  - Token permutation is not expressive enough
  - Direct modeling using tree structures
- Sophisticated model architecture to use global context

Conclusion

Conclusion

- In this paper,
  - We propose two notation prediction tasks
    - Auto-suggestion and Consistency checks
  - We present a fine-tuned BERT
    - particularly optimized on the tasks
    - outperforms other baselines
- We therefore foresee our method as aiding in helping authors of mathematical texts