Improving Numeracy by Input Reframing and Quantitative Pre-Finetuning Task

Chung-Chi Chen,[1] Hiroya Takamura,[2] Ichiro Kobayashi,[3] Yusuke Miyao[2]
[1] Artificial Intelligence Research Center, AIST, Japan
[2] Ochanomizu University, Japan
[3] University of Tokyo, Japan
c.c.chen@acm.com, takamura.hiroya@aist.go.jp,
koba@is.ocha.ac.jp, yusuke@is.s.u-tokyo.ac.jp

Abstract

Numbers have unique characteristics to words. Teaching models to understand numbers in text is an open-ended research question. Instead of discussing the required calculation skills, this paper focuses on a more fundamental topic: understanding numerals. We point out that innumeracy—the inability to handle basic numeral concepts—exists in most pretrained language models (LMs), and we propose a method to solve this issue by exploring the notation of numbers. Further, we discuss whether changing notation and pre-finetuning along with the comparing-number task can improve performance in three benchmark datasets containing quantitative-related tasks. The results of this study indicate that input reframing and the proposed pre-finetuning task is useful for RoBERTa.

1 Introduction

Numerals are an indispensable part of narratives and provide much fine-grained information.[1] How models learn the number system has intrigued many researchers (Spithourakis and Riedel, 2018; Naik et al., 2019; Chen et al., 2019; Wallace et al., 2019; Zhang et al., 2020). Researchers have long discussed some numeracy-related properties of pretrained language models (LMs). In this study, we propose a new concept — *innumeracy*. The problem of innumeracy becomes most evident when models are faced with numerals that do not appear in training data, e.g., when the range of numerals in training data is different from that in the test data. Moreover, LMs often face difficulties understanding numbers even though the numbers are present in the training data. One possible cause of this problem is that numerals can have various notations, some of which are difficult to understand from their subwords. Another possible cause is that LMs are not pretrained to deal with numbers. Therefore, in this study, we address the problem of innumeracy via input reframing and quantitative pre-finetuning tasks.

Model	Notation	Tokenized Example
BERT	*Org.*	"147", "##70", "##2"
	Digit	"1", "4", "7", "7", "0", "2"
	SN	"1", ".", "47", "##70", "##200", "##00", "##0", "##e", "+", "05"
RoBERTa	*Org.*	"147", "702"
	Digit	"1", "4", "7", "7", "0", "2"
	SN	"1", ".", "47", "70", "200000", "E", "+", "05"

Table 1: Tokenized example. Org. and SN denote original and scientific notation, respectively.

Input reframing refers to changing the notations of numbers, which can be one of the crucial clues for understanding numerals (Zhang et al., 2020; Chen et al., 2021). In addition to the original notation, we consider the digit-based and scientific notations. Table 1 lists examples of using different representations for numerals. Our experiments indicate that RoBERTa (Liu et al., 2019) performs poorly than BERT-based models (Devlin et al., 2019; Yasunaga et al., 2022) in understanding numerals. However, its performance is at par with vanilla BERT-based models with a proper input reframing method. Furthermore, in previous studies, pretraining with the self-supervised learning approach been determined to be a compelling method (Devlin et al., 2019; Yasunaga et al., 2022). However, it is costly to pretrain a new LM from scratch. Thus, an alternative way is to design pre-finetuning tasks to enhance the ability of LMs (Aghajanyan et al., 2021). Inspired by this idea, we propose a novel pre-finetuning task to enhance the ability of the models to deal with quantitative questions and improve the numeracy of the models. Specifically, the proposed method automatically generates a simple dataset for the comparing-numbers task (ComNum), and uses it to pre-finetune LMs. This study experiments with representative pretrained LMs, includ-

[1] In this paper, we focus on the numerals represented by digits (0 to 9 and decimal point) and do not discuss those written in words such as "one" and "two".

Findings of the Association for Computational Linguistics: EACL 2023, pages 69–77
May 2-6, 2023 ©2023 Association for Computational Linguistics

ing BERT, RoBERTa, and LinkBERT (Yasunaga et al., 2022), and the experimental results show that pre-finetuning with the proposed ComNum improves the performance in the Quantitative Natural Language Inference (QNLI) task regardless of the LMs used.

To evaluate the influence of the input reframing and the quantitative pre-finetuning task, we constructed the Quantitative 101 dataset, which is a combination of three benchmark datasets: Numeracy-600K (Chen et al., 2019), EQUATE (Ravichander et al., 2019), and NumGLUE Task 3 (Mishra et al., 2022). The tasks in Quantitative 101 include Quantitative Prediction (QP), QNLI, and Quantitative Question Answering (QQA). In the future, Quantitative 101 can be used as a new collection by researchers studying the quantitative skills of LMs. [2]

2 Related Work

Numeracy, one of the recent hot topics in NLP, incorporates many skills such as calculation, algebra, and geometry. Some previous studies (Spithourakis and Riedel, 2018; Chen et al., 2019) have discussed the prediction of the masked number tasks, while others (Wallace et al., 2019; Naik et al., 2019; Zhang et al., 2020) have explored numeracy from the perspective of embedding properties. The math word problem (Chen et al., 2021; Mishra et al., 2022) is a high-level task requiring several numeracy skills. The textual representation of numerals, such as digit-based or scientific notations-based, is one of the possible directions for improving numeracy. Chen et al. (2021) suggested to use a digit-based encoder to encode numerals. Meanwhile, Zhang et al. (2020) used scientific notation to represent numerals and explored scale understanding tasks. In this paper, we explore the role of these notations of numbers in quantitative skill tasks.

A recent trend is to design pretraining tasks to enhance the capability of models to understand natural language. Devlin et al. (2019) proposed two pretraining tasks: masked language model (MLM) and next sentence prediction (NSP), and broadened the horizons of the transformer-based natural language processing research direction. Yasunaga et al. (2022) designed a new cross-document pretraining task, called document relation prediction (DRP), to improve the performance of LMs in several benchmark datasets, especially those requiring multi-hop reasoning and multi-document understanding skills. To the best of our knowledge, this is one of the earliest works proposing a tailor-made pre-finetuning task to understanding numerals. Our experimental results also support the usefulness of the proposed task, specifically in the QNLI task.

Task	Question	Answer
ComNum	[Num 1] is equal to [Num 2]. [Num 1] is smaller than [Num 2]. [Num 1] is larger than [Num 2].	TRUE/FALSE
QP	FED'S DUDLEY REPEATS EXPECTS GDP GROWTH TO PICK UP IN 2014, FROM [Masked] PCT POST-RECESSION AVERAGE	1
QNLI	S1: Nifty traded above 7500, Trading Calls Today S2: Nifty above 7400	Entailment
QQA	Elliot weighs 180 pounds whereas Leon weighs 120 pounds. Who has a bigger gravity pull? Option1: Elliot Option2: Leon	Option 1

Table 2: Example for each task.

3 Datasets and Tasks

This section introduces two datasets: the Comparing Numbers Dataset (CND) and Quantitative 101, with the corresponding quantitative tasks, including ComNum, QP, QNLI, and QQA.

3.1 Comparing Numbers Dataset (CND)

Comparing numbers (ComNum) is one of the basic quantitative skills. We propose the Comparing Numbers dataset (CND) to test the ability of different pretrained LMs to perform the ComNum task. CND is an automatically created dataset, and the ComNum task is designed as a binary classification task. In essence, the models need to determine whether a given statement of comparing numbers is true or false. In the CND, there are only three templates as shown in Table 2. There is one training set and two test sets in CND. Specifically, we randomly select two numbers from 0 to 199,999 and insert them into the template. The selected numbers are deleted from the pool of numbers to avoid duplication. Finally, 100,000 instances are obtained, and the numbers in all instances are unique. Note that the distributions of each template and answers are balanced. 80% of the dataset is considered as the training set and the remaining 20% is taken as the CND-T1 test set. Next, two numbers from 4,000,000 to 5,000,000 are randomly selected for 10,000 times to construct the CND-T2 test set. Thus, the order of magnitude of the training set and the first test set (CND-T1) is from 0 to 5, and that of the other test set (CND-T2) is 6. In this study, we focused on natural numbers, and fu-

[2] We release this dataset for academic use and follow the license of the sources (Appendix C).

	BERT		RoBERTa		LinkBERT		FinBERT	
	CND-T1	CND-T2	CND-T1	CND-T2	CND-T1	CND-T2	CND-T1	CND-T2
Original	99.86	95.59 (↓ 4.27)	99.44	86.75 (↓ 12.69)	99.92	97.58 (↓ 2.34)	99.55	78.37 (↓ 21.18)
Digit-based	99.96	99.03 (↓ 0.93)	99.92	98.46 (↓ 1.46)	99.99	96.54 (↓ 3.45)	99.96	97.03 (↓ 2.93)
Scientific Notation	99.92	99.68 (↓ **0.24**)	99.82	99.13 (↓ **0.69**)	99.95	99.81 (↓ **0.14**)	99.72	98.78 (↓ **0.94**)

Table 3: Experimental results of ComNum task. The evaluation metric is Micro-average of F1 score (%).

Model	Notation	QP		QNLI					QQA	Score
		Comment	Headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
BERT	*Original*	70.44%	57.46%	64.40%	59.20%	72.29%	60.42%	99.91%	53.20%	67.17
	Digit-based	65.38%	54.74%	57.86%	56.46%	71.36%	60.11%	99.11%	**53.75%**	64.85
	Scientific Notation	65.31%	55.99%	**64.42%**	**60.73%**	72.23%	59.66%	99.56%	**53.24%**	66.39
CN-BERT	*Digit-based*	69.93%	54.84%	61.07%	**60.27%**	**75.54%**	**65.39%**	99.42%	52.53%	**67.37**
	Scientific Notation	64.87%	56.40%	**66.39%**	54.70%	**75.41%**	**63.94%**	99.42%	51.90%	66.63
LinkBERT	*Original*	68.81%	55.70%	59.94%	56.85%	73.43%	59.01%	99.91%	54.14%	65.97
	Digit-based	63.76%	55.41%	59.54%	**57.42%**	**73.63%**	**60.17%**	99.73%	53.44%	65.39
	Scientific Notation	65.81%	**56.05%**	57.00%	56.78%	**75.51%**	58.51%	99.82%	**54.33%**	65.48
CN-LinkBERT	*Digit-based*	68.61%	54.44%	**63.59%**	55.08%	71.21%	58.99%	**100.00%**	50.44%	65.30
	Scientific Notation	63.48%	53.15%	**62.02%**	**59.39%**	**75.70%**	**62.61%**	99.73%	52.11%	**66.02**

Table 4: Experimental results of the BERT-based models. The results in bold are the ones that are better than the *Original*. The score indicates Quantitative-101 Score.

ture studies can extend our results to decimals and fractions. Since natural numbers are in the infinite set, and it is impossible to let models learn with a dataset containing all magnitudes and numbers, we designed the task in the way following the human learning process because human beings do not need to learn to count from zero to trillion to get the ability to compare all numbers.

3.2 Quantitative 101

Quantitative 101 collects recent benchmark datasets and focuses on quantitative tasks. There are three tasks in Quantitative 101, including Quantitative Prediction (QP), Quantitative Natural Language Inference (QNLI), and Quantitative Question Answering (QQA). This section briefly introduces the tasks, and we further provide details in Appendix C.

QP is the task of predicting the correct magnitude of the masked numeral. Although a possible choice would be to predict the exact number given a context, doing so is often very difficult, even for a human. For example, the QP listed in Table 2, in which the correct answer is 2.2. However, making an accurate rough estimate for the magnitude would often be feasible only for seasoned experts. We attempt to test whether models can also learn such a numeracy skill after being trained with a large amount of data. Thus, we adopt Numeracy-600K (Chen et al., 2019) as the dataset for this task. Chen et al. (2019) designed this task as an eight-class classification task, which includes the magnitude from 1 to 6, decimal, and a magnitude larger than 6. Numeracy-600K contains two subsets: market comments and blog headlines.

QNLI is the task of making natural language inferences based on quantitative clues. It is a complex version of ComNum, because the given sentences could be varied. The example of QNLI presented in Table 2 shows that models need to compare numbers based on more complex semantics. We selected EQUATE (Ravichander et al., 2019) to experiment on real-world scenarios for QNLI. EQUATE has five subsets, including RTE-QUANT, AWP-NLI, NEWSNLI, REDDITNLI, and Stress Test.

QQA is the other format for testing whether models can understand numerals and semantics. We selected the Task 3 subset of NumGLUE (Mishra et al., 2022) for the QQA experiments. Table 2 provides an example of this dataset. It is under a binary-classification setting, and each instance has two options.

We chose these three datasets to test the basic quantitative skills of models. We noticed that several instances in these datasets can be solved using only the basic ability to understand numbers. However, the other subtasks in NumGLUE required reasoning skills including the generation of equations. These tasks are not the target of this paper.

4 Methods

4.1 Notation of Numbers

The findings of previous studies (Chen et al., 2021; Zhang et al., 2020) suggest two methods that are worth trying: digit-based notation and scientific no-

Model	Notation	QP		QNLI					QQA	Score
		Comment	Headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
RoBERTa	*Original*	60.46%	58.03%	60.15%	57.64%	79.58%	58.77%	98.93%	51.96%	65.69
	Digit-based	**69.25%**	57.65%	59.40%	56.69%	78.90%	**62.38%**	**99.91%**	**54.34%**	**67.31**
	Scientific Notation	**64.32%**	55.49%	60.08%	57.41%	78.68%	**60.81%**	**100.00%**	**53.67%**	**66.31**
CN-RoBERTa	*Digit-based*	**64.25%**	55.92%	**68.96%**	**58.80%**	77.99%	**60.99%**	**99.73%**	50.88%	**67.19**
	Scientific Notation	60.28%	54.85%	**62.15%**	**58.74%**	65.92%	**59.59%**	**99.47%**	**52.27%**	64.16

Table 5: Experimental results of the RoBERTa-based models.

tation. Table 1 shows an example for each method. *Original* signifies that we did not perform any preprocessing on the input data, and the results are tokenized based on WordPiece (Schuster and Nakajima, 2012; Wu et al., 2016) and Byte-Pair Encoding (BPE) (Sennrich et al., 2016). In the *Digit-based* method, we separated a numeral into digits. In the *Scientific Notation* method, we we converted numerals into scientific notation according to the method described in Zhang et al. (2020), and Table 1 provides examples to show that tokenizers provide different results in this case. Note that we pad the mantissa to 10 significant figures to retain the information of most numerals.

4.2 Pre-Finetuning Task

We pre-finetune LMs with the CND for learning the numeracy of comparing numbers. We believe that this learning process can make models aware of the numerals and may help answer the questions listed in Table 2. We further test whether the proposed pre-finetuned method is helpful in the QP, QNLI, and QQA tasks. We primarily use BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and LinkBERT (Yasunaga et al., 2022) for the experiments. Since the market comment subset for the QP task is in the financial domain, we also experiment with FinBERT (Araci, 2019) in this subset. The pre-finetuned LMs using BERT, RoBERTa, LinkBERT, and FinBERT as initial models are named CN-BERT, CN-RoBERTa, CN-LinkBERT, and CN-FinBERT, respectively. During the pre-finetuning process, we use the *Digit-based* or *Scientific Notation* reframing methods to transform the numerals in the input data. Thus, each proposed pre-finetuned LM has two versions depending on the notation of numbers.

5 Experiment

5.1 Innumeracy

Innumeracy can be tested via various experiments. In this section, we observe the innumeracy phenomenon with the empirical results of the ComNum task. We aim to answer whether LMs have different performances between CND-T1 and CND-T2. We use the micro-average of the F1 score to evaluate the results of the ComNum task. Table 3 shows the results. It is not surprising that models perform well in CND-T1. However, model performances drop when we test using CND-T2. In CND-T2, the order of magnitude of the numerals is different from that in the training set. We call this phenomenon "innumeracy", and find that both *Digit-based* and *Scientific Notation* perform well for most pretrained LMs. In particular, using *Scientific Notation* method leads to the least performance drops with all LMs.[3]

Model	Reframing	QP-Comment
FinBERT	*Original*	65.26%
	Digit-based	**69.89%**
	Scientific Notation	**70.03%**
CN-FinBERT	*Digit-based*	**68.84%**
	Scientific Notation	**69.76%**

Table 6: Results of the FinBERT-based models.

5.2 Experimental Results

We follow the setting of previous studies to use the macro-average of F1 score for the QP task and the micro-average of F1 score for the QNLI and QQA tasks. Table 4 presents the results of the BERT-based models, and Table 5 presents the results of the RoBERTa-based models.[4] To evaluate the aggregate performance, we average all results as in previous studies (Dua et al., 2019; Mishra et al., 2022), and named this score the Quantitative-101 Score. First, it can be observed that all notation methods and the pre-finetuning task improved the overall performance of RoBERTa, and lead RoBERTa to perform at par with the BERT-based LMs. Second, we observed that the proposed pre-finetuning task helped improve the QNLI task performance. Third, using a proper reframing method improved the QQA task performance. Fourth, the

[3] We provide more analysis on this point in Appendix B.

[4] We provide a fine-grained analysis in Appendix A for the QNLI-Stress Test.

Model	Preprocessing	QP		QNLI					QQA	Score
		Comment	Headline	RTE-QUANT	AWP-NLI	NEWSNLI	REDDITNLI	Stress Test		
RoBERTa	*Original*	60.46%	58.03%	60.15%	57.64%	79.58%	58.77%	98.93%	51.96%	65.69
CN-RoBERTa		**86.86%**	**77.29%**	**62.52%**	56.70%	78.82%	**64.29%**	**99.94%**	50.71%	**72.14**

Table 7: Results of CN-RoBERTa without input reframing.

reframing methods and the pre-finetuning task were not helpful for the BERT-based LMs in the QP task as well as the overall performance.

Table 6 shows the results of the FinBERT-based models in QP-comment. The results indicate that the performances of FinBERT can be improved with a proper reframing method. Additionally, the proposed CN-FinBERT performs better than the *Original* FinBERT.

To sum up our findings, the input reframing methods can improve the performance of RoBERTa and FinBERT. However, it does not work for BERT-based models. The proposed pre-finetuning task can improve the performance in the QNLI task regardless of the LM used.

5.3 Ablation Analysis

In this section, we train CN-RoBERTa without input reframing for ablation analysis. Table 7 shows the results. The results indicate that the performances of QP tasks were improved significantly, and the performance of QNLI tasks was also improved. These results indicate the proposed pre-finetuning task is important for the QP tasks, but input reframing is not. However, the performance of the QQA did not improve without input reframing. This result implies that, for QQA, input reframing provides some hints to the models to make predictions. Overall, this study does not find a silver bullet for solving quantitative problems, but shows that input reframing and basic quantitative pre-finetuning design are promising directions.

6 Conclusion

This study deals with the innumeracy of LMs and shows that the notation of numbers matters, especially for RoBERTa. We also propose a novel pre-finetuning task for improving the quantitative skills, and find that the performance in the QNLI task can be improved after pre-finetuning. We hope our results in Quantitative 101 lead to a more in-depth discussion on the ability of LMs to understand numerals.

Limitations

The first limitation of the paper is that we focus on the numerals represented by digits (0 to 9 and decimal point) and do not discuss those written in words such as "one" and "two". Future work can extend the findings of this work and transfer the numeral words to digits. The second limitation of this paper is that we do not discuss long text scenarios because the length of the instances in the datasets is within 512. Future work can design quantitative-related tasks with longer documents and examine whether the proposed methods still work. The third limitation of this paper is that we do not train the model from scratch with the proposed input reframing methods. We leave it as one of the open questions for future studies. The fourth limitation of this work is that we do not experiment with all cases, including using data in several ranges and experimenting with all kinds of pretrained LMs, to prove that the innumeracy phenomenon is a general phenomenon. Instead, we present a pilot exploration of the phenomenon and further pay attention to improving the performances of other quantitative-related tasks.

Ethical Note

All datasets used in our experiment are available online, and we provide the details and the license information in Appendix C. We release the pre-finetuned LMs (CN-BERT, CN-RoBERTa, CN-LinkBERT, and CN-FinBERT) on the Hugging Face models platform.[5] Future work can reproduce our results easily and use our pre-finetuned LMs for further research issues. Please refer to Appendix B for details.

Acknowledgements

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

[5] https://huggingface.co/models

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811.

Dogu Araci. 2019. FinBERT: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. NQuAD: 70,000+ questions for machine comprehension of the numerals in text. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2925–2929.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600K: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. 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 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. NumGLUE: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523, Dublin, Ireland. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Carolyn Rose, and Eduard Hovy. 2019. Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380, Florence, Italy. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2104–2115, Melbourne, Australia. Association for Computational Linguistics.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland. Association for Computational Linguistics.

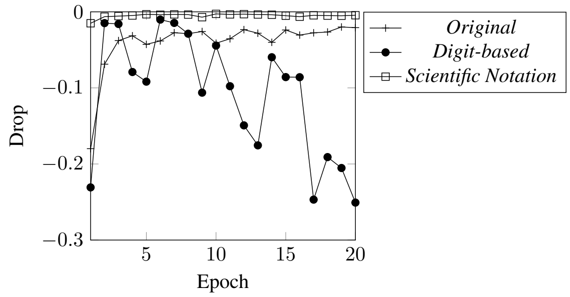

Figure 1: BERT's innumeracy phenomenon. (Performance Drop between CND-T1 and CND-T2.)

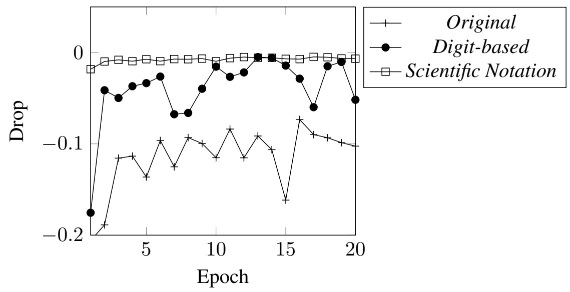

Figure 2: RoBERTa's innumeracy phenomenon. (Performance Drop between CND-T1 and CND-T2.)

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do language embeddings capture scales? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4889–4896, Online. Association for Computational Linguistics.

A Analysis of QNLI-Stress Test

QNLI-Stress Test uses the data collected from AQuA-RAT, and was annotated by an automatic method (Ravichander et al., 2019). We follow the splitting method in NumGLUE Task 7 (Mishra et al., 2022) to separate it into training, development, and test sets. First, we find 316 repeated instances in both training and evaluation sets (development and test sets). We already removed these repeated instances from the training set in our experiment. Second, we check the instances by removing all numerals in each instance and find that 2,229 instances appear in both training and evaluation sets, with 1,639 appearing in the same training and test sets, and 80.17% have the same answer. That could be the reason that the models perform well in this dataset, since most instances do not need to understand numerals.

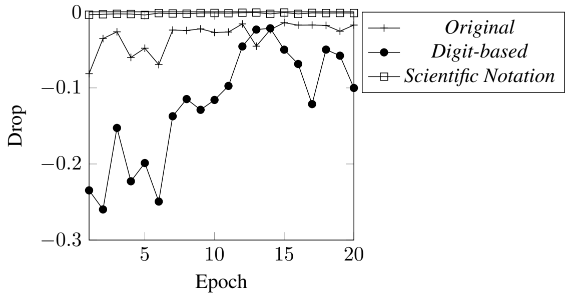

Figure 3: LinkBERT's innumeracy phenomenon. (Performance Drop between CND-T1 and CND-T2.)

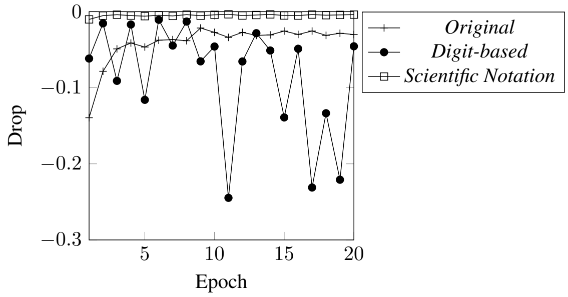

Figure 4: FinBERT's innumeracy phenomenon. (Performance Drop between CND-T1 and CND-T2.)

B Implementation Detail

We used the Hugging Face transformers package (Wolf et al., 2019) for the experiment. [6] Intel Xeon Gold CPU and Nvidia Tesla V100 w/32GB are the CPU and GPU used in our experiment. Table 8 provides the links to the LMs used in our experiment. All pre-finetuned LMs (CN-BERT, CN-RoBERTa, CN-LinkBERT, and CN-FinBERT) are released on the Hugging Face platform.

Figure 1 to 4 present the tracing results of the drop between CND-T1 and CND-T2 during the training process. It can be observed that when using Scientific Notation, the performances of LMs stabilizes more quickly. In contrast, the change of the performances with the *Digit-based* method varies, and we did not obtain stable results in some cases.

C Dataset

CND is our own generated dataset; therefore, we did not have to obtain license permissions to use it. There are three subsets in the proposed Quantitative 101. Numeracy-600K (Chen et al., 2019) for

[6] `https://huggingface.co/docs/transformers/index`

	URL
BERT (Devlin et al., 2019)	https://huggingface.co/bert-base-uncased
RoBERTa (Liu et al., 2019)	https://huggingface.co/roberta-base
LinkBERT (Yasunaga et al., 2022)	https://huggingface.co/michiyasunaga/LinkBERT-base
FInBERT (Araci, 2019)	https://huggingface.co/ProsusAI/finbert

Table 8: Reference for the models in our experiments.

Model	Reframing Method	URL
CN-BERT	*Digit-based*	https://huggingface.co/NLPFin/CN-BERT-Digit
	Scientific Notation	https://huggingface.co/NLPFin/CN-BERT-Sci
CN-RoBERTa	*Original*	https://huggingface.co/NLPFin/CN-RoBERTa
	Digit-based	https://huggingface.co/NLPFin/CN-RoBERTa-Digit
	Scientific Notation	https://huggingface.co/NLPFin/CN-RoBERTa-Sci
CN-LinkBERT	*Digit-based*	https://huggingface.co/NLPFin/CN-LinkBERT-Digit
	Scientific Notation	https://huggingface.co/NLPFin/CN-LinkBERT-Sci
CN-FinBERT	*Digit-based*	https://huggingface.co/NLPFin/CN-FinBERT-Digit
	Scientific Notation	https://huggingface.co/NLPFin/CN-FinBERT-Sci

Table 9: Reference for the proposed models.

QP task could be downloaded from GitHub[7], and it is under the Creative Commons Attribution-Non-Commercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. EQUATE (Ravichander et al., 2019) for QNLI task can also be downloaded from GitHub[8], and it is under the MIT License. NumGLUE (Mishra et al., 2022) for QQA task can be downloaded from the page of Allen Institute for AI (AI2)[9], and it is under the ODC Attribution License (ODC-By).[10] In the following subsections, we provide details of each subset. The README document of the dataset provides all details about the separation. Please download the dataset from `https://huggingface.co/datasets/NLPFin/Quantitative101`.

C.1 Quantitative Prediction

Quantitative prediction (QP) is a task to predict the correct magnitude of the masked numeral. For example, even for a human, it is difficult to predict the exact numeral (2.2) of the QP's instance in Table 2; however, some seasoned experts can make a correct rough estimate of the magnitude. We attempt to test whether models also learn to make such predictions after being trained with a large amount of data. Thus, we adopt Numeracy-600K (Chen et al., 2019) as the dataset for this task. Chen et al. (2019) designed this task as an eight-class classification task, which includes the magnitude from 1 to 6, decimal, and the magnitude larger than 6. We follow their setting in this paper. There are two subsets, including 600K market comments and 600K news headlines. We use 80%, 10%, and 10% of instances as training, development, and test sets in each subset, respectively.

C.2 Quantitative Natural Language Inference

Quantitative Natural Language Inference (QNLI) is a complex version of ComNum because the given sentences can be varied. The example of QNLI provided in Table 2 shows that models need to compare numbers based on more complex semantics. We select EQUATE (Ravichander et al., 2019) to experiment on real-world scenarios for QNLI. EQUATE has five subsets collected from different sources, including RTE-QUANT, AWP-NLI, NEWSNLI, REDDITNLI, and Stress Test. Since four of these subsets are less than 1,000 instances, we perform the 10-fold cross-validation in the experiments. For the Stress Test, which contains 7,500 instances, we follow the splitting method in NumGLUE Task 7 (Mishra et al., 2022) to separate it into training, development, and test sets. Ravichander et al. (2019) designed the QNLI task as a two or three-class classification task depending on the subset. We follow their settings for each subset.

C.3 Quantitative Question Answering

Quantitative Question Answering (QQA) is the other format for testing whether models can un-

[7] `https://github.com/aistairc/Numeracy-600K`

[8] `https://github.com/AbhilashaRavichander/EQUATE/blob/master/LICENSE`

[9] `https://allenai.org/data/numglue`

[10] `https://github.com/allenai/numglue/blob/main/license.txt`

derstand numerals and semantics. We selected the Task 3 subset of NumGLUE (Mishra et al., 2022) for the QQA experiments. Table 2 provides an example of this dataset. It is under a binary-classification setting, and each instance has two options. We follow Mishra et al. (2022) to separate the dataset into training, development, and test sets.