Tokenization on the Number Line is All You Need

Anonymous ACL submission

Abstract

Despite the recent breakthroughs in language modeling, their ability to represent numbers is insufficient. Subword tokenization, the standard choice for number representation, breaks down a number into arbitrary chunks thereby failing to explicitly capture the relationship between two numbers on on the number-line. To alleviate this shortcoming, alternate approaches have been proposed that modify numbers at various stages of the language modeling pipeline. These methods can be broadly classified into three categories that make changes to a) the notation (e.g. scientific vs decimal) b) vocabulary (e.g. introduce a new token for numbers in range $10-100$) and c) architectural changes to directly regress to a desired number. The contributions of this work are three fold – firstly, we propose vocabulary level changes in the decoding stage and study its behavior. Next, we study the performance of both the proposed approach and existing number representation schemes in the context of masked number presentation. We find that a carefully designed tokenization scheme is both the simplest to implement and sufficient i.e. with similar performance to the state-of-the-art approach that requires making significant architectural changes. Finally, we evaluate the various number representation schemes on the downstream task of numerical fact estimation (for fermi problems) in a zero-shot setting and find similar trends i.e. changes at the tokenization level achieve near state-of-the-art results while requiring minimal resources compared to other number representation schemes.

1 Introduction

The standard practice in the language modeling community is to process numbers in exactly the same manner as words. This second class treatment of numbers leads to their inaccurate representation and therefore, limited numerical understanding of large-scale language models (LMs). To illustrate, a number like \$799 is *subword* tokenized (Sennrich et al., 2016; Schuster and Nakajima, 2012) as 79 and ##9. Such a tokenization method, by construction, prevents accurately modeling the relationship of this number with other numbers on the number line say, \$800, as the surface forms share no common tokens. Many alternatives have been proposed to capture the scalar magnitude of numbers; see survey by Thawani et al. (2021b) for further details.

Degree of Change	**Expected Predictions for:** iPhone `[MASK]` costs \$ `[MASK]`.
(default)	iPhone 13 costs \$ 79 ##9 .
Notation	iPhone 13 costs \$ 7 . 99 e 2 .
Vocabulary	iPhone 10-100 costs \$ 100-1000 .
Model	iPhone 13.0000 costs \$ 799.0000 .

Table 1: Multiple approaches to masked number prediction or number decoding. Color Coding: Tokens in the vocabulary of BERT (Devlin et al., 2019). New tokens. Continuous-valued predictions.

All the approaches proposed to capture the magnitude of numbers fall into one of the following categories, corresponding to modifications to a) notation (e.g. scientific vs decimal) b) vocabulary (e.g. introducing new tokens that denote all numbers within a specified range) and c) architectural changes (e.g. directly regressing to a number). Table 1 shows the various approaches on a example sentence. While all these approaches overcome the limitations of using subword tokenization, they present their unique challenges and trade-offs. In this work, we study the utility of these number representations in the *decoding* stage and therefore, focus on the task of masked number prediction.

The contributions of this work are as follows:

1. We propose using a modification to the tokenization scheme for numbers with a particular focus on decoding (outputting) of numbers.

2. We study the utility of this approach and other approaches to represent numbers in language modeling in the context of masked number prediction. We find that applying our tokenization scheme leads to near state-of-the-part performance requiring no additional pre-training or architectural changes.

3. Finally, we evaluate the number representation schemes on their ability to generalize to downstream tasks – in this case, numerical fact estimation in the context of solving fermi problems (Kalyan et al., 2021). We find trends similar to the task of masked number prediction demonstrating the utility of the simple yet effective tokenization scheme in the decoding setting.

2 Methods

In this section, we dive deeper into each of the three number representation categories and discuss the trade-offs involved in using them.

Change of Notation. We first discuss the most straightforward approach towards number representation. Here, the numbers are represented in an alternate notation – e.g. scientific notation as opposed to decimal notation. Note that this approach does not require changing any of the other components of language modeling. In this work, we consider the following variations:
Scientific. Using scientific notation in lieu of the usual decimal notation was first proposed by Zhang et al. (2020). In this work, we closely follow their version with minor implementation level changes [1] Importantly, note that following the notation change, the tokenizer nevertheless splits it into subwords as before.
Digits: Here, the number is split into its constituent digits or characters, e.g., 329 becomes 3 2 9. This approach offers a consistent decomposition of numbers into digits, as opposed to the arbitrary tokens from subword segmentation and has been proven effective on simple numeric probes as well as arithmetic word problems Geva et al. (2020).
Change of Vocabulary. Unlike words, the notion of distance or similarity is more obviously defined for numbers in terms of their separation on the number line, a cognitive tool that human beings are known to intuitively used to process numeracy (Dehaene, 2011). This forms the basis of our approach i.e. numbers within a specified range are collapsed into a single token – at the cost of precise representation of numbers. This approach to tokenizing the number space is analogous to stemming of words. Stemming is a simple technique to collapse low frequency words to their lemma in order to curtail the vocabulary size, e.g., *play*ing, *play*er and *play*ed all collapse into the token for *play*. Similarly, exponent embeddings collapse multiple numbers into a single token covering a range of numbers.

While this approach has already been used in the context of encoding numbers (Berg-Kirkpatrick and Spokoyny, 2020; Thawani et al., 2021a), our work is the first to use and study this approach when outputting or decoding numbers.

Change in Architecture. Several recent methods have modified the language model to emit continuous values when predicting numbers. At their core, they operate by regressing to the desired number conditioned on the language context. See Berg-Kirkpatrick and Spokoyny (2020) for a thorough comparison within this class of methods. We directly compare against their best variant: Discrete Latent Exponents, which first models the exponent part of a number as a multinomial, and then uses it to parameterize a truncated log normal distribution to sample the mantissa as a continuous value.

3 Experiments

We evaluate different number decoders and evaluate them on the task of masked number prediction (MNP). Before analyzing their performance, we first describe the datasets, models and metrics used.

Dataset and Metrics. We follow (Berg-Kirkpatrick and Spokoyny, 2020) to finetune and evaluate our models on three datasets – Financial News Articles (FinNews), its subset containing mostly price-based numbers (FinNews-$), and Scientific Articles (Sci); all numbers in these datasets lie between 1-10^{16}. We evaluate using two metrics – a) Exponent Accuracy (E-Acc) that checks whether the predicted answer is of the same order of magnitude as the ground truth and b) Log Mean Absolute Error (LMAE). For more details on both the datasets and metrics, refer (Berg-Kirkpatrick and Spokoyny, 2020).

[1] 329 is written as 329 `[EXP]` 2. However, we find that representing the same instead as $3x29$ where ‘x’ is the common English alphabet, works better in practice.

Metrics	FinNews E-Acc ↑	FinNews LogMAE↓	FinNews-$ E-Acc↑	FinNews-$ LogMAE↓	Sci E-Acc↑	Sci LogMAE↓
Baselines						
Train-Mean	1.02	7.69	6.02	4.68	0.01	8.81
Train-Median	5.52	1.88	10.58	2.66	49.52	0.83
Train-Mode	24.23	2.02	8.13	6.30	49.52	1.00
Subword-Pad8	63.56	0.68	29.05	1.36	68.02	0.68
Notation-change						
Digit-Pad17	52.23	0.93	33.04	1.37	55.12	0.91
Scientific-Pad8	52.53	0.84	NA	NA	71.14	0.66
Vocabulary-change						
DExp-fixed	74.40	0.65	57.14	0.93	81.16	0.51
Exp	73.70	0.60	56.99	0.92	**81.32**	0.44
Model-change	Berg-Kirkpatrick and Spokoyny (2020)					
DExp	**74.56**	**0.50**	**57.50**	**0.89**	81.17	**0.39**

Table 2: Order of magnitude accuracy (E-Acc) and Log Mean Absolute Error (LMAE) over the test set of three datasets, contrasting the three degrees of freedom for improving numeracy of language models. NA denotes subword models which were unable to emit valid numbers for at least 50% of the examples.

Baselines. Our primary baseline is the standard approach of subword tokenization. We require each number prediction to be 8 tokens long, with appropriate padding. Additionally, we evaluate on three trivial baselines that make a constant prediction corresponding to the mean, median, and mode of all numbers present in the training set.

Models. First, we compare against both the approaches discussed in Sec. 2 that employ change of notation i.e. scientific and digit, with a padding of 8 and 17 respectively. Next, among the approaches the introduce architectural changes, we compare against the state-of-the-art discrete exponent model (DExp) proposed by (Berg-Kirkpatrick and Spokoyny, 2020). Finally, we compare against two variations that introduce vocabulary level changes – both, discretize the number line with logarithmic-ally sized bins (with base 10). The two variants differ in how the mantissa is chosen – either a constant of 5 (DExp-fixed) or the log-scale mean of the extremes of a bin (DExp), e.g. the token 10-100 is replaced by the number 31.62. We extend the code provided by Berg-Kirkpatrick and Spokoyny (2020) for most of our experiments[2].

Further, note that we only compare number decoders and not the encoders – therefore, when numbers are present in the input, standard encoding schemes are used. For approaches with changes to vocabulary and architecture, we follow (Berg-Kirkpatrick and Spokoyny, 2020) and use exponent embeddings to encode numbers (with no shared parameters with the decoder's tokens) and for approaches with notation changes, we use subword tokenization.

3.1 Results

We find that the straightforward, change of notation approaches are inferior to the subword baseline. This is in contrast to prior work on extrapolating the arithmetic abilities of language models by simple notation changes (Nogueira et al., 2021; Geva et al., 2020). This result suggests that simple preprocessing changes like changes of notation are not sufficient for contextual understanding of numbers for language modelling

Next, we find that while DExp model is the best performing method, approaches that instead make changes to the vocabulary are a close second – notably, over 90% of the gain in E-Acc from subword to DExp models for FinNews corpus, is achievable without modelling the mantissa at all!

3.2 Downstream zero-shot transfer

Given the trends observed in masked number prediction, we are interested in analyzing the utility of these models on a downstream number prediction task. For this purpose, we evaluate on numerical fact estimation. We pick the Fermi

[2] https://github.com/dspoka/mnm

Fermi-Real 510 egs.	FinNews E-Acc ↑	FinNews LogMAE↓	FinNews-$ E-Acc↑	FinNews-$ LogMAE↓	Sci E-Acc↑	Sci LogMAE↓
Sub-Pad8	26.11	2.38	16.07	3.17	25.89	2.84
Dig-Pad17	18.79	2.58	NA	NA	23.27	2.87
Sci-Pad8	24.78	2.93	NA	NA	20.09	2.75
DExp-fixed	**32.21**	2.19	24.38	**2.42**	27.29	2.42
DExp	**32.21**	**2.13**	**25.06**	2.51	**28.19**	**2.40**
Fermi-Syn 3437 egs.	FinNews E-Acc ↑	FinNews LogMAE↓	FinNews-$ E-Acc↑	FinNews-$ LogMAE↓	Sci E-Acc↑	Sci LogMAE↓
Sub-Pad8	28.72	2.89	19.12	3.25	38.93	2.83
Dig-Pad17	21.66	2.93	NA	NA	40.73	2.87
Sci-Pad8	25.75	3.06	NA	NA	27.05	2.76
DExp-fixed	39.08	2.61	40.85	**2.42**	46.86	2.52
DExp	**39.22**	**2.44**	**41.36**	2.44	**47.60**	**2.48**

Table 3: Downstream performance of our main methods over fact estimation for solving Fermi Problems. NA denotes subword models which were unable to emit valid numbers for at least 50% of the examples.

Problems dataset (Kalyan et al., 2021), which consists of challenging estimation problems such as “How many tennis balls fit in a school bus?”. Solving such questions require sestimating numeric facts such as ‘the volume of a tennis bus’ or ‘the length of a bus.’

We evaluate each of our models on such annotated facts provided as part of both the real and synthetic datasets part of the fermi problem dataset. The task setup is of masked number prediction as before, e.g., “the size of a tennis ball is `[MASK]` cubic centimeters." We report E-Acc and Log MAE as before, in Table 3. We find similar trends as in 3.1 i.e. change of notation is sufficient while vocabulary-change approaches are closely behind approaches that make architectural changes – highlighting that most of the gains could be retained by simply tokenizing in number space.

4 Related Work

The NLP community has recently proposed several ways of improving the numeracy of language models, including architectural and notation interventions. Several such methods are aimed at helping LMs extrapolate easily to larger numbers (Kim et al., 2021) or for improving their arithmetic skills (Nogueira et al., 2021). We restrict our analysis to the task of *approximately* decoding numbers in MNP setting, which requires different methods and metrics compared to the tasks which require *exact* arithmetic skills (Thawani et al., 2021b).

The method we highlight in this paper i.e. tokenization in number space, has been previously used in different settings. Zhang et al. (2020) probe word embeddings from BERT with similar exponent embeddings on the task of measurement estimation (Elazar et al., 2019). Others have shown the benefits of using such exponent embeddings as *number encoders* for language models, whether it be for the task of masked number prediction (Berg-Kirkpatrick and Spokoyny, 2020) or masked word prediction (Thawani et al., 2021a). Our work extends these results with further evidence of the representational power gained by simply tokenizing numbers on the number line.

5 Conclusion

Subword tokenization, the standard approach to representing numbers leads to inaccurate numerical understanding. In this work, we propose a simple yet effective tokenization based approach that alleviates this shortcoming. In addition, we analyze number representation approaches that make notational (e.g. scientific vs. decimal) and architectural changes. We find that the proposed tokenization scheme has near state-of-the-art order-of-magnitude accuracy (74.40% vs SotA 74.56%) while requiring minimal resources as opposed to making architectural changes. Finally, we evaluate these methods in a zero-short setting on the numerical fact estimation task in the context of fermi problems. We find that in this challenging setting, the same trends hold – indicating that *tokenization is all you need* to represent numbers effectively and with minimal effort.

References

Taylor Berg-Kirkpatrick and Daniel Spokoyny. 2020. An empirical investigation of contextualized number prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4754–4764, Online. Association for Computational Linguistics.

Stanislas Dehaene. 2011. *The number sense: How the mind creates mathematics*. OUP USA.

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.

Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3973–3983, Florence, Italy. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Ashwin Kalyan, Abhinav Kumar, Arjun Chandrasekaran, Ashish Sabharwal, and Peter Clark. 2021. How much coffee was consumed during EMNLP 2019? fermi problems: A new reasoning challenge for AI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7318–7328, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jeonghwan Kim, Giwon Hong, Kyung-min Kim, Junmo Kang, and Sung-Hyon Myaeng. 2021. Have you seen that number? investigating extrapolation in question answering models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7031–7037, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2021. Investigating the limitations of transformers with simple arithmetic tasks.

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.

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.

Avijit Thawani, Jay Pujara, and Filip Ilievski. 2021a. Numeracy enhances the literacy of language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6960–6967, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Avijit Thawani, Jay Pujara, Pedro A. Szekely, and Filip Ilievski. 2021b. Representing numbers in NLP: a survey and a vision. *CoRR*, abs/2103.13136.

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.