# Predicting Numerals in Text Using Nearest Neighbor Language Models

**Taku Sakamoto**
The University of Tokyo
t_sakamoto@nii.ac.jp

**Akiko Aizawa**
National Institute of Informatics
The University of Tokyo
aizawa@nii.ac.jp

## Abstract

Commonsense about quantitative properties is essential for a deep understanding of texts containing numerals. However, naive language models (LMs) treat numerals as string tokens; therefore, they lack an understanding of the magnitudes of numerals, resulting in a difficulty in acquiring the commonsense. In this study, we apply the $k$-nearest neighbor LM ($k$NN-LM) to the masked numeral prediction (MNP) task, which measures the quantitative commonsense of LMs. $k$NN-LM extends pretrained neural LMs with the $k$-nearest neighbor ($k$NN) search. Since it can utilize patterns that appear in the datastore for prediction, we expect an improvement in numeral prediction accuracy, which is associated with a high rate of occurrence of out-of-vocabulary (OOV) words. Through experiments, we verified that the retrieval-based method is effective for fine-grained predictions of numerals from context, especially for the OOV numerals. We also compared two different context spans for context representations to improve the accuracy of $k$NN search by using only the words that are closely related to the masked numeral: the mask and its surrounding words, and the mask and its subsequent words. Our results reveal that using only the embeddings of mask tokens for numerals in $k$NN search is the most effective approach for realizing MNP tasks.

## 1 Introduction

Real-world objects and events have various quantitative properties, such as size, weight, length, and price. Commonsense about these quantitative properties is essential for a deep understanding of texts containing numerals and for reasoning on a similar or better level than humans. Figure 1 shows examples of a masked numeral prediction (MNP) task requiring quantitative commonsense. The first example requires deriving a numeral referring to height that is considered tall based on commonsense about the distribution of human heights.

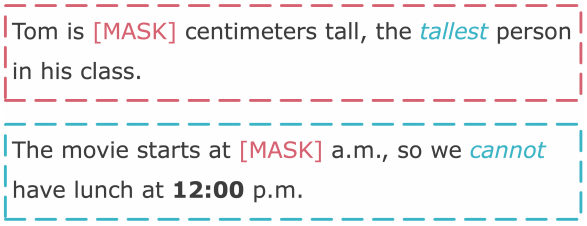


Figure 1: Examples of the masked numeral prediction task requiring quantitative commonsense.

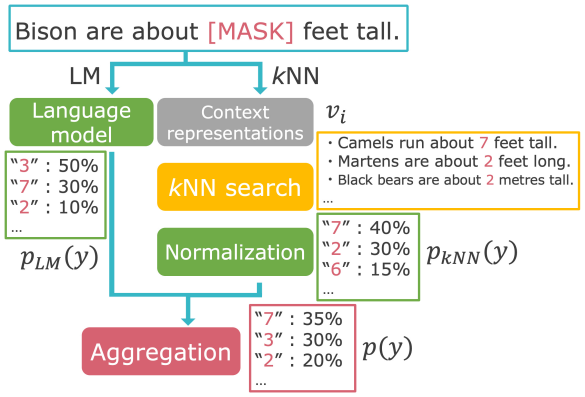


Figure 2: Overview of $k$NN-LM for the MNP task.

The second example requires deriving the value of a typical movie length using commonsense and subtracting it from 12 p.m. Humans can easily choose numerals that approximately correspond to the ground-truth answers to these questions with considerable confidence. However, for models that lack such commonsense and computational skills, such inferences pose a challenge.

In recent years, large-scale neural language models (LMs), such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), GPT-3 (Brown et al., 2020), and GPT-4 (OpenAI, 2023), have achieved comparable or better performance than humans on various natural language processing (NLP) tasks. However, previous studies have reported that these models still perform poorly on tasks that require quantitative commonsense (Elazar et al., 2019), such as MNP (Spithourakis and Riedel, 2018; Lin



*Findings of the Association for Computational Linguistics: ACL 2023*, pages 4795–4809
July 9-14, 2023 

et al., 2020), numerical error detection or correction (Chen et al., 2019), and numerical question answering (Dua et al., 2019; Zhou et al., 2022).

One of the main reasons why LMs fail to perform well on tasks that require quantitative commonsense is that they do not learn the mapping between strings of numerals and their magnitudes accurately. Naive LMs treat numerals in text only as string tokens, that is, other words. While humans can associate the magnitude of a numeral with the string of the numeral, LMs that treat numerals only as strings are unable to accurately make such associations for arbitrary numerals (Wallace et al., 2019). This makes it difficult for naive LMs to understand the magnitude of numerals, resulting in difficulty in acquiring quantitative commonsense.

To address this problem, previous studies have attempted to employ methods such as using word embeddings of numerals that reflect the magnitudes of the numerals (Wallace et al., 2019; Thawani et al., 2021), adding texts of arithmetic formulas to the training data (Geva et al., 2020), training LMs with a loss function that depends on the magnitudes of numerals (Sakamoto and Aizawa, 2021), and tokenizing numerals in a text into single digits to allow LMs to understand the concept of digits (Spithourakis and Riedel, 2018). However, these methods require fine-tuning or additional pre-training specific to the understanding of numerals. Therefore, in this study, we aim to improve the performance of LMs in a task that requires quantitative commonsense (specifically, the MNP task (Spithourakis and Riedel, 2018; Lin et al., 2020; Sundararaman et al., 2022)) without such additional training that is specific to numerals by using the $k$-nearest neighbor LM ($k$NN-LM) (Khandelwal et al., 2020b), which is an LM extended by a retrieval-based method.

In addition, numerals have a higher rate of occurrence of out-of-vocabulary (OOV) words than regular words (Spithourakis et al., 2016a), making it difficult even for recent large neural LMs to accurately predict numerals in sentences from the context. In this study, based on the hypothesis that numerals that appear in similar contexts tend to be of the same type (e.g., date, amount of money, and number of people) and similar sizes, we expect that $k$NN-LM model will improve the accuracy of the MNP task by reflecting numerals that appear in similar contexts in the prediction results. We also believe that an advantage of using $k$NN search is not only the improvement in top-k accuracy but also the improvement in interpretability by providing contexts with similar use of the predicted numeral.

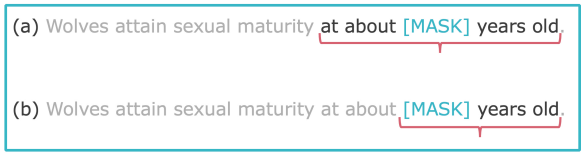


Figure 3: Two context ranges of a masked numeral for $k$NN search.

In our experiments, we used the pre-trained BERT of HuggingFace (Wolf et al., 2020) as the base LM for $k$NN-LM. Two types of context ranges were used to compute the representation of the context of the masked numeral: the numeral mask and its surrounding words (Figure 3 (a)), and the numeral mask and its subsequent words (Figure 3 (b)). Contextual range Figure 3 (b) is expected to improve search accuracy by focusing on words that follow the numerals, such as units, which strongly represent the type of the preceding numerals.

For both ranges, nearest neighbors were searched based on context representation, which is computed as the average vector of the embedding vectors of all the words in the range. $k$NN-LM outperformed the base LM in MNP on most of the datasets we used. In addition, it was confirmed that the range of only the mask token for a numeral was the most effective context range for $k$-nearest neighbor ($k$NN) search in MNP.

To summarize, our contributions are as follows:

- We apply $k$NN-LM to MNP and show that the retrieval-based method can improve the performance of pre-trained LMs without additional training specific to numerals.
- We experiment with several different types of context ranges and find the optimal range for $k$NN search for the MNP task.
- We analyze the prediction accuracy for numerals included in the model vocabulary and those not included and confirm that $k$NN search significantly improves the performance for OOV numerals, which are difficult to predict with naive LMs.

## 2 Related Work

### 2.1 Masked Numeral Prediction

The MNP task can be used as a probing task to evaluate the quantitative commonsense acquired by LMs.

Spithourakis and Riedel (2018) evaluated the numeracy of a long short-term memory model by using the MNP task and concluded that current LMs have a problem with learning the mappings between strings of numerals and their magnitudes. Therefore, to help the models understand the magnitude of numerals, they proposed a method to predict numerals as continuous Gaussian distributions and a method using character-level recurrent neural networks (Graves, 2013; Sutskever et al., 2011) for prediction, which led to an improvement in their prediction accuracy.

Lin et al. (2020) used the MNP task with uniquely determinable masked numerals, such as "A bird usually has [MASK] legs" or "A car usually has [MASK] wheels," and evaluated quantitative commonsense acquired by BERT and RoBERTa. They showed that even pre-trained LMs that achieve comparable performance to humans on many NLP tasks perform significantly worse than humans on this task. In addition, although the pre-trained LMs seemed to make correct predictions, they often failed to maintain these predictions even for small sentence changes that did not change the answer, such as when the target sentence was changed to "A car usually has [MASK] *round* wheels." This finding implies that achieving suitable robustness of model predictions is also a challenge.

In this study, considering these problems of the current LMs, we do not revise the base model itself but reinforce the model predictions by using a retrieval-based approach, specifically, a $k$NN search computed on the similarity of the contexts.

### 2.2 Retrieval Augmented Methods in NLP

Retrieval-based approaches, which refer to the datastores as external knowledge, have been successful in many NLP tasks (Meng et al., 2021), such as named entity recognition (Wang et al., 2022), machine translation (Khandelwal et al., 2020a), and question answering (Guu et al., 2020).

$k$NN-LM is an LM whose predictions are augmented with the results of a $k$NN search for similar texts (Khandelwal et al., 2020b). The detailed design of the model is described in Section 4. When predicting a masked word in a sentence, $k$NN-LM searches the dataset for sentences similar to the context around the masked word. It aims to improve prediction accuracy by reflecting the searched nearest neighbors in the prediction score of the base LM. Since $k$NN search is based on the distance in the embedding space of the base LM, it has the advantages of not requiring additional training for the search and of being able to use any dataset as the datastore for the search. Improvements in perplexity from 18.65 to 15.79 on the WikiText-103 dataset (Merity et al., 2016) are reported. Khandelwal et al. (2020b) also found that $k$NN-LM is particularly useful for predicting rare patterns due to the augmentation provided by the retrieval-based approach. Based on the hypothesis that it may also be effective in predicting numerals, where rare patterns occur frequently (Spithourakis et al., 2016b), we applied $k$NN-LM to MNP in this study.

## 3 Task

### 3.1 Task Description

In this study, we used the MNP task to evaluate the numeracy of LMs. This task is defined as follows:

**Input:** A passage containing exactly one target numeral masked with a special token "[MASK]"

**Output:** A ranking of predicted numerals

There is exactly one masked numeral per passage, and the prediction model can see the other numerals in the same passage when making predictions for passages with more than one numeral. We initially considered masking multiple numerals in a passage; however, we decided to limit the number of masked numerals to one because masking multiple numerals would make the prediction difficult even for humans (e.g., "Restaurant reservations are preferred after [MASK] p.m. because the movie starts at [MASK] p.m.") and a single mask is more suitable for investigating whether LMs can capture the semantic relationship between the numerals.

### 3.2 Evaluation

The LMs (including $k$NN-LM) generate a probability distribution over numeral tokens in their vocabulary using a softmax function. The top-$k$ accuracy is a metric that evaluates the predicted ranking of the numeral tokens created from the generated probability distribution (Lin et al., 2020). It calculates the percentage of predictions such that

| Dataset | Numeracy-600K | ACLsent | FinNum | DROP |
|---|---|---|---|---|
| #passages | 420,000 | 1,753 | 3,992 | 4,329 |
| Ave. passage length [token] | 13.0 | 53.2 | 37.1 | 285.2 |
| #numerals | 523,425 | 8,521 | 8,043 | 65,063 |
| #numerals per passage | 1.2 | 4.9 | 2.0 | 15.0 |
| % of integers | 98 % | 82 % | 83 % | 96 % |
| % of OOV numerals | 4 % | 25 % | 22 % | 12 % |

Table 1: Statistics across four different datasets (training set).

| Dataset | Numeracy-600K | ACLsent | FinNum | DROP |
|---|---|---|---|---|
| #decimals | 8,680 (33%) | 1,503 (69%) | 1,301 (72%) | 2,416 (30%) |
| #numerals with commas | 3,888 (14%) | 299 (13%) | 186 (10%) | 2,436 (31%) |
| #large numerals | 5345 (20%) | 217 (9%) | 225 (12%) | 1,796 (22%) |
| #OOV numerals | 26,045 (100%) | 2,178 (100%) | 1,790 (100%) | 7,852 (100%) |

Table 2: Statistics on OOV numerals across four different datasets (training set).

the ground-truth numeral token is within the top $k$ predicted tokens in the ranking.

The top-$k$ accuracy simply evaluates whether the ground-truth numerals are included in the top $k$ predictions. It does not consider how close the predicted numerals are to the corresponding ground truth. However, in the MNP task, a model that predicts numerals closer to the ground truth is generally considered to be a better model, even if the predictions are incorrect. Therefore, in this study, we used the top-$\boldsymbol{k}$ accuracy with a fixed numerical error percentage allowed in each calculation to evaluate the LMs in terms of the magnitude of the difference between the ground-truth numeral and the predicted numeral. In our experiments, we used $k = 1, 3, 5$, and 10 for evaluation.

## 4 Nearest Neighbor Language Model

$k$NN-LM (Figure 2) predicts masked tokens in input sentences $y$ using two different approaches, namely an LM and a $k$NN search (Khandelwal et al., 2020b). It then adds these prediction scores together with a mixture ratio $\lambda$ to obtain a final prediction score $p(y)$:

$$p(y) = \lambda p_{\mathrm{kNN}}(y) + (1 - \lambda) p_{\mathrm{LM}}(y) \quad (1)$$

where $\lambda$ is a fixed parameter, $p_{\mathrm{kNN}}(y)$ is the prediction score of $k$NN search calculated using the softmax function on the negative distance between the test context and the top $k$ similar contexts in the datastore, and $p_{\mathrm{LM}}(y)$ is the prediction score reported by the LM.

In $k$NN search, two types of context ranges are used: the numeral mask and its surrounding $n$ words (see Figure 3 (a)) and the numeral mask and its subsequent $n$ words (see Figure 3 (b)). For both ranges, the average of the embedding vectors of all the words in the range is defined as the context representation of the masked numeral. While Khandelwal et al. (2020b) used only the words before the mask to calculate context representations, we used the aforementioned two types of context ranges for the following two reasons. First, in our experiments, we used BERT as the base LM, which is a bidirectional LM. Second, we hypothesized that words that are more closely related to the magnitudes of numerals, such as units, tend to appear around the numerals, especially after them.

## 5 Experiments

### 5.1 Dataset

In this study, we used the following four datasets with different domains and passage lengths:

- Numeracy-600K (article titles) (Chen et al., 2019),
- ACLsent (scientific papers) (Abekawa and Aizawa, 2016),
- FinNum (financial tweets) (Chen et al., 2018),
- DROP (Wikipedia) (Dua et al., 2019).

From each dataset, 70% of the total passages were used as training data, 10% as validation data, and the remaining 20% as evaluation data. The main results on the FinNum and DROP datasets are shown in Appendix C, considering that their trends were generally the same as those of the other datasets.

The statistics of the passages and numerals contained in the aforementioned datasets are shown in

| % of NE | Window size (n) | before and after the mask Top1↑ | Top3↑ | Top10↑ | only after the mask Top1↑ | Top3↑ | Top10↑ |
|---|---|---|---|---|---|---|---|
| $= 0\%$ | 0 (only [MASK]) | **40.0%** | **60.7%** | **77.1%** | **40.0%** | **60.7%** | **77.1%** |
| | 1 | 38.1% | 58.7% | 76.0% | 38.5% | 58.9% | 75.9% |
| | 2 | 37.8% | 58.6% | 76.0% | 37.4% | 57.9% | 75.4% |
| | 5 | 35.8% | 55.1% | 73.5% | 35.3% | 55.5% | 73.9% |
| | max length | 33.7% | 50.8% | 70.0% | 33.9% | 53.5% | 72.1% |
| $\leq 10\%$ | 0 (only [MASK]) | **56.0%** | **71.8%** | **87.4%** | **56.0%** | **71.8%** | **87.4%** |
| | 1 | 54.0% | 70.4% | 86.9% | 54.4% | 70.5% | 86.6% |
| | 2 | 53.8% | 70.4% | 86.9% | 52.9% | 69.6% | 86.2% |
| | 5 | 50.6% | 67.3% | 85.1% | 50.5% | 67.3% | 84.9% |
| | max length | 46.5% | 63.3% | 82.6% | 48.5% | 65.3% | 83.3% |

Table 3: Top-$k$ accuracy of $k$NN search on the Numeracy-600K dataset when two different context ranges are used to compute the contextual representation: one with the mask and the $n$ words before and after it (Figure 3 (a)), and one with the mask and its subsequent $n$ words (Figure 3 (b)). "% of NE" indicates the percentage of numerical error allowed in each top-$k$ accuracy calculation.

Table 1. Numeracy-600K, ACLsent, and FinNum have only a few sentences per passage compared to DROP, which has longer passages. ACLsent and DROP contain 5–15 numerals per passage, while Numeracy-600K and FinNum contain less than 5 numerals per passage. The types of numerals appearing in the passages also differ depending on the dataset domain. Numeracy-600K and DROP contain more four-digit numerals, such as year numbers, compared with the other datasets. Partly because of this reason, they also have a relatively lower percentage of decimals and OOV numerals, which are not included in the BERT vocabulary. ACLsent and FinNum contain many decimals and infrequent numerals, such as numerals from experimental results and statistics and monetary values and percentage changes in stock prices, as reported by the statistics in Table 1.

Table 2 shows the statistics on OOV numerals across the four datasets. In particular, the percentages of the three main categories of numerals that are not included in the BERT vocabulary are presented, namely decimals, numerals with commas, and large numerals. The category "#large numerals" includes numerals larger than 6,000, which is the largest numeral in the BERT vocabulary. The aforementioned categories have intersections with each other. The trend of OOV numerals appearing in the dataset varies significantly depending on the domain and writing style. It can also be observed that decimals account for the majority of OOV numerals in all datasets.

### 5.2 Experimental Setup

In the experiments, we used the BERT model "bert-base-uncased" from HuggingFace Transformers (Wolf et al., 2020) as the base LM for $k$NN-LM. This base LM was used to make predictions from the context of masked numerals. The word embeddings for $k$NN search were the output of the second-to-last layer of this model. In this paper, the $k$NN-LM using a BERT model fine-tuned by the MNP task as the base LM is called the $k$NN-LM fine-tuned by the MNP task.

BERT-DExp (Berg-Kirkpatrick and Spokoyny, 2020) and NumGPT (Jin et al., 2021) are powerful baselines that deal with the prediction of numerals from context. These methods reflect the numeral's magnitudes in the numeral embeddings and have improved the ability to roughly predict numerals (i.e., the rate of agreement for the number of digits). However, we did not adopt these models as the base LM for $k$NN-LM in this study because we believe that methods that reflect the numeral's magnitudes in the numeral embeddings can have a negative impact on the accurate prediction of numerals, thereby losing the advantages of retrieval-based approaches, which are beneficial in terms of accuracy.

Infrequent numerals, decimals, and numerals with commas are not included in the naive BERT vocabulary; thus, such numerals in the datasets are split into multiple numeral tokens by the BERT tokenizer in the preprocessing stage. Tables 1 and 2 show the percentages and statistics of OOV numerals; such numerals in the datasets are split into multiple numeral tokens by the BERT tokenizer in the preprocessing stage. However, in the test set, to prevent partial masking, the numerals are masked with a single token (i.e., without splitting them first). Consequently, it may be impossible for naive

| Method | % of NE | pre-trained Top1↑ | pre-trained Top3↑ | pre-trained Top5↑ | pre-trained Top10↑ | fine-tuned Top1↑ | fine-tuned Top3↑ | fine-tuned Top5↑ | fine-tuned Top10↑ |
|---|---|---|---|---|---|---|---|---|---|
| $k$NN | $= 0\%$ | 32.9% | 52.7% | 62.0% | 72.2% | **40.0%** | 60.7% | 69.2% | 77.1% |
| | $\leq 10\%$ | 48.6% | 65.5% | 73.9% | 84.4% | **56.0%** | 71.8% | **78.9%** | **87.4%** |
| | $\leq 30\%$ | 57.8% | 78.0% | 85.9% | 93.2% | **64.9%** | **82.5%** | **89.0%** | **94.5%** |
| | $\leq 50\%$ | 69.0% | 86.7% | 92.1% | 96.5% | **75.1%** | **89.7%** | **93.8%** | **97.0%** |
| LM | $= 0\%$ | 12.6% | 28.6% | 38.7% | 55.3% | 37.4% | 58.2% | 65.8% | 73.4% |
| | $\leq 10\%$ | 24.2% | 41.2% | 51.6% | 67.8% | 53.8% | 68.3% | 74.5% | 81.9% |
| | $\leq 30\%$ | 28.7% | 50.2% | 61.8% | 77.1% | 63.2% | 77.6% | 82.5% | 87.5% |
| | $\leq 50\%$ | 40.4% | 60.5% | 70.4% | 82.9% | 73.2% | 83.6% | 87.1% | 90.7% |
| $k$NN+LM | $= 0\%$ | 32.7% | 47.1% | 54.6% | 66.1% | 39.4% | **61.6%** | **69.6%** | **77.6%** |
| | $\leq 10\%$ | 48.0% | 62.0% | 68.9% | 79.3% | 55.4% | **71.9%** | 78.0% | 85.6% |
| | $\leq 30\%$ | 56.6% | 73.3% | 80.6% | 88.8% | 64.2% | 81.5% | 86.3% | 91.2% |
| | $\leq 50\%$ | 67.9% | 83.5% | 88.4% | 93.4% | 74.2% | 87.9% | 90.8% | 93.9% |

Table 4: Top-$k$ accuracy of $k$NN-LM on the Numeracy-600K dataset. "$k$NN," "LM," and "$k$NN+LM" indicate the accuracy of $k$NN search alone, the accuracy of the base LM, and the accuracy of the entire $k$NN-LM, respectively. "% of NE" indicates the percentage of numerical error allowed in each top-$k$ accuracy calculation.

| Method | % of NE | pre-trained Top1↑ | pre-trained Top3↑ | pre-trained Top5↑ | pre-trained Top10↑ | fine-tuned Top1↑ | fine-tuned Top3↑ | fine-tuned Top5↑ | fine-tuned Top10↑ |
|---|---|---|---|---|---|---|---|---|---|
| $k$NN | $= 0\%$ | 22.3% | 37.7% | 45.3% | 54.3% | 27.8% | 43.3% | 49.5% | 56.5% |
| | $\leq 10\%$ | 32.1% | 47.8% | 56.2% | 68.1% | 37.9% | 54.1% | 61.5% | 70.4% |
| | $\leq 30\%$ | 39.9% | 59.4% | 68.9% | 81.2% | 45.3% | 63.6% | **73.0%** | **83.2%** |
| | $\leq 50\%$ | 50.0% | 70.7% | 79.1% | 88.8% | 54.2% | **73.7%** | **81.4%** | **90.0%** |
| LM | $= 0\%$ | 20.2% | 37.3% | 45.8% | 56.0% | 30.1% | 46.0% | 52.3% | 58.4% |
| | $\leq 10\%$ | 29.4% | 47.2% | 56.0% | 68.1% | 38.4% | 53.7% | 60.6% | 67.9% |
| | $\leq 30\%$ | 36.1% | 56.4% | 66.5% | 76.0% | 45.6% | 61.2% | 68.0% | 74.6% |
| | $\leq 50\%$ | 47.5% | 65.8% | 73.1% | 80.4% | 53.8% | 68.8% | 73.9% | 79.4% |
| $k$NN+LM | $= 0\%$ | 25.3% | 43.1% | 50.9% | 59.9% | **31.4%** | **47.9%** | **54.5%** | **61.3%** |
| | $\leq 10\%$ | 34.8% | 52.7% | 61.2% | 71.9% | **40.6%** | **57.4%** | **64.2%** | **72.6%** |
| | $\leq 30\%$ | 41.8% | 63.0% | 72.3% | 80.3% | **47.6%** | **65.3%** | 72.6% | 79.7% |
| | $\leq 50\%$ | 51.8% | 73.4% | 79.8% | 84.4% | **56.5%** | 73.6% | 78.8% | 84.7% |

Table 5: Top-$k$ accuracy of $k$NN-LM on the ACLsent dataset.

LMs to predict the masks of OOV numerals with zero-error rate. However, the frequency of numeral tokens in the BERT vocabulary ensures predictions with an error of less than 10% (except for large numerals; see Table 2). Since OOV numerals rarely appear in the datastore for $k$NN search, a zero-error rate would be hardly possible regardless of the single token masking. Therefore, we believe that our methods can be fairly compared to the others even with this masking strategy.

For $k$NN search, we set $k = 50$ and used the $L^2$ norm for calculating the distance of the context vectors. The mixing ratio of $k$NN search results and LM prediction scores was set to $\lambda = 0.2$ based on the results of our preliminary experiments. The experimental results are given in terms of average scores of two or more runs. Other experimental settings are shown in Appendix A.

## 6 Results and Discussion

### 6.1 Methods for Representing the Context for $k$NN Search

The results of $k$NN search using different context ranges are shown in Table 3. $k$NN search with only the embedding vector of mask tokens for masked numerals achieved the highest accuracy in the MNP task, in both context ranges. We suggest that this may be because the embedded representations of the mask tokens of numerals contain sufficient information to predict the masked numerals near the last layer of the fine-tuned LM. The results of the experiment comparing two context ranges on the ACLsent dataset are shown in Appendix B. Initially, we expected that the context range after the mask would be more efficient than the range before and after the mask because it can effectively utilize units that often follow numerals. However, we did not observe a significant difference between them.

| Method | % of NE | Known numerals Top1↑ | Top3↑ | Top5↑ | Top10↑ | OOV numerals Top1↑ | Top3↑ | Top5↑ | Top10↑ |
|---|---|---|---|---|---|---|---|---|---|
| $k$NN | $= 0\%$ | **41.3%** | 62.7% | 71.5% | 79.5% | **14.0%** | **20.4%** | **24.3%** | **30.6%** |
| | $\leq 10\%$ | 57.3% | 73.3% | **80.4%** | **88.8%** | **30.2%** | **42.9%** | **49.5%** | **60.8%** |
| | $\leq 30\%$ | 66.2% | **83.8%** | **90.2%** | **95.4%** | **39.6%** | **56.4%** | **65.0%** | **77.3%** |
| | $\leq 50\%$ | **76.5%** | **90.8%** | **94.7%** | **97.7%** | **49.0%** | **67.6%** | **75.9%** | **85.6%** |
| LM | $= 0\%$ | 39.1% | 60.8% | 68.7% | 76.6% | 4.5% | 7.8% | 9.9% | 12.7% |
| | $\leq 10\%$ | 56.1% | 70.9% | 77.1% | 84.6% | 11.1% | 19.4% | 24.2% | 31.2% |
| | $\leq 30\%$ | 65.7% | 80.3% | 85.1% | 90.0% | 16.6% | 28.4% | 33.7% | 41.4% |
| | $\leq 50\%$ | 75.9% | 86.2% | 89.6% | 93.0% | 24.0% | 36.1% | 41.4% | 48.6% |
| $k$NN+LM | $= 0\%$ | 41.1% | **63.8%** | **72.1%** | **80.4%** | 7.3% | 18.0% | 21.1% | 25.2% |
| | $\leq 10\%$ | **57.5%** | **73.6%** | 79.8% | 87.5% | 15.4% | 37.7% | 43.2% | 51.3% |
| | $\leq 30\%$ | **66.5%** | 83.2% | 88.0% | 92.7% | 21.9% | 49.6% | 55.5% | 63.4% |
| | $\leq 50\%$ | **76.5%** | 89.4% | 92.3% | 95.2% | 30.0% | 59.7% | 64.5% | 71.4% |

Table 6: Top-$k$ accuracy of the fine-tuned $k$NN-LM for numerals included in and out of the vocabulary numerals in the Numeracy-600K dataset.

| | | |
|---|---|---|
| **Test** | **[MASK]**-magnitude earthquake hits Taiwan, no injuries or tsunami reported | ANS: **6.5** |
| ***k*NN** | **7.2**-magnitude earthquake hits near Guam and Mariana Islands; Tsunami warning not expected | PRED: **7.2** |
| | **5.7**-magnitude earthquake hits near Mindoro, Philippines | PRED: **5.7** |
| | **7.4**-magnitude earthquake hits Indonesia's Sumatra island, Aceh province [PHOTOS] | PRED: **7.4** |
| | **5.0**-magnitude earthquake hits Canada and Northern US and leaves residents shaken (video) | PRED: **5.0** |
| | **7.2**-magnitude earthquake strikes Vanuatu Islands; tsunami warning issued | PRED: **7.2** |
| **Test** | A theory that 'Guardians of the Galaxy' could take place in the year **[MASK]** | ANS: **2045** |
| ***k*NN** | What is God saying about the Year **2012** ? | PRED: **2012** |
| | Google May Venture Into Retail Store Business In Year **2013** | PRED: **2013** |
| | Pew Research Center Survey: 41% of Americans believe Jesus (pbuh) will return by the year **2050** | PRED: **2050** |
| | 577,190 Americans will die of cancer in year **2012** | PRED: **2012** |
| | Prayers for the year **2010** | PRED: **2010** |
| **Test** | Lowest mass extrasolar planet discovered, HD 10180 b has **[MASK]** times Earth's mass | ANS: **1.4** |
| ***k*NN** | Astronomers discover record massive star, **320** times the mass of Sun in R136 cluster | PRED: **320** |
| | New biofuel process creates **20** times more energy than existing methods | PRED: **20** |
| | Russia's meteor pieces worth **40** times more than the price of gold today | PRED: **40** |
| | Japanese Damaged nuclear power plant **1,000** times higher than normal | PRED: **1,000** |
| | UFC 108 fighter salaries: Rashad Evans makes **75** times more than least paid fighter | PRED: **75** |

Table 7: Top-5 output examples of $k$NN search for masks of OOV numerals in the Numeracy-600K dataset.

In the following experiments, the results of $k$NN search were obtained when using only the embeddings of mask tokens, which exhibited the best accuracy in the experiment presented in this section.

### 6.2 Masked Numeral Prediction

Tables 4 and 5 show the top-$k$ accuracy of $k$NN-LM on the Numeracy-600K and ACLsent datasets for the MNP task (without and with fine-tuning on the task). The results for the FinNum and DROP datasets are shown in Appendix C.

By comparing the prediction accuracy before and after, we observed that fine-tuning the base LM on the MNP task improved the prediction accuracy of $k$NN search and $k$NN-LM on both datasets. This confirms the effectiveness of fine-tuning the base LM in $k$NN search and $k$NN-LM. In both cases, before fine-tuning, $k$NN search outperformed the LM in terms of accuracy. In particular, on the Numeracy-600K dataset, the largest dataset used in our experiments, $k$NN search significantly outperformed the LM both before and after fine-tuning. However, on the ACLsent dataset, the smallest dataset used in our experiment, the performance difference was not as pronounced, indicating that dataset size can influence the extent of improvement through fine-tuning. These findings demonstrate that with a sufficiently large datastore, $k$NN search can achieve moderate prediction accuracy

| Test Dataset | Datastore for kNN Search | Top1↑ | Top3↑ | Top5↑ | Top10↑ |
|---|---|---|---|---|---|
| **Numeracy-600K** | **Numeracy-600K** | **40.0%** | **60.7%** | **69.2%** | **77.1%** |
| | ACLsent | 7.9% | 18.1% | 26.7% | 39.6% |
| | DROP | 10.9% | 22.0% | 28.1% | 37.8% |
| **ACLsent** | **ACLsent** | **27.8%** | **43.3%** | **49.5%** | **56.5%** |
| | Numeracy-600K | 5.9% | 12.1% | 16.3% | 23.7% |
| | DROP | 5.4% | 14.2% | 20.4% | 31.6% |

Table 8: Top-$k$ accuracy with 0% error of $k$NN search in cross-domain settings.

without additional fine-tuning, unlike LM.

When comparing the prediction accuracy of each method after fine-tuning, we found that on both datasets, the prediction accuracy of $k$NN search alone or $k$NN-LM exceeded that of the base LM alone by approximately 2% to 5% in all settings. This confirms the effectiveness of $k$NN search in the MNP task. In particular, $k$NN search demonstrated superior accuracy after Top3 and achieved a margin of error of 10% or more, suggesting that it can retrieve a more diverse set of numerals as predictions compared to LM.

Table 6 lists the top-$k$ accuracy of the fine-tuned $k$NN-LM for numerals in and out of the BERT vocabulary in the Numeracy-600K dataset. The results for the numerals included in the vocabulary show almost the same trend as the overall results (Table 4). By contrast, $k$NN search significantly outperformed the LM in predicting the OOV numerals. Although it is challenging to accurately compare their performance owing to the vocabulary and datastore limitations affecting LM and $k$NN search, respectively, we believe that in settings allowing for a small margin of numerical error, their performance can be considered fairly comparable. The results for OOV numerals in the ACLsent dataset are shown in Appendix D.

### 6.3 Output Examples of *k*NN Search for OOV Numerals

Table 7 shows the top-5 output examples of $k$NN search for masks of OOV numerals in the Numeracy-600K dataset. An LM fine-tuned on this dataset with the MNP task was used for $k$NN search. In each sentence, one numeral is shown in bold, indicating that $k$NN search was performed with the bold numeral masked.

The OOV numerals are masked, and their low frequency of occurrence makes it difficult to find contexts in the datastore wherein the same numerals appear. However, this result shows that $k$NN search could find contexts that are remarkably close to that of the test context, although the exact match accuracy of numerals was not high. In the first example, $k$NN search found a context for an earthquake of similar magnitude, and the test "no injuries or tsunami reported" and the first-predicted context "Tsunami warning not expected" are extremely close. The second example is considered one of the most difficult for $k$NN search because the answer "2065" does not appear in the datastore. However, despite the short context, it correctly estimated that the masked numeral is a future year and succeeded in finding a considerably close numeral in the datastore, although it was the third prediction.

However, the results also reveal a limitation of $k$NN search. In the third test sentence, the numeral with "times" as the unit is masked. Although $k$NN search outputted contexts that contain numerals with "times," as the unit in all of the top 5 cases, it failed to find contexts that contain numerals that are close to the answer. This may be because a deeper understanding of the contexts is required for masked numerals with units such as "times" which allow for a wider range of preceding numerals. Similarly, there were cases where $k$NN search was not extremely effective in predicting the amount of money that followed the "$", which was considered to allow for a wider range of numerals.

While $k$NN search achieved successful predictions in some cases and faced challenges in others, as shown in the table, humans can easily understand the rationale behind the predictions (e.g., same units or similar contexts). This improved interpretability stands as a significant advantage of $k$NN search over LMs.

### 6.4 *k*NN Search in Cross-Domain Setting

Table 8 shows the results of $k$NN search with a datastore from different domains. We performed $k$NN search with Numeracy-600K, ACLsent, and DROP as datastores for the Numeracy-600K and

ACLsent datasets. The results show that the accuracy of $k$NN search in the cross-domain setting was significantly lower than that achieved using the same-domain datasets as the datastore. This indicates that the types and properties of the numerals in these three datasets differ greatly, and in many cases, similar contexts and numerals were not found by $k$NN search. These results suggest that $k$NN search can achieve the best performance only with a datastore that contains a larger number of diverse sentences compared to those used in this study. Future developments should focus on experiments and analysis of $k$NN search using a large-scale datastore.

## 7 Conclusion

In this study, we applied $k$NN-LM to the MNP task and quantitatively evaluated its prediction accuracy. The results show that the numerical absolute errors were reduced by utilizing $k$NN search for numeral prediction compared to existing methods. In particular, the prediction accuracy greatly improved for numerals not included in the model vocabulary, which are difficult to predict with naive LMs. We also experimented with two different context ranges and confirmed that the most effective method for $k$NN search is the one using only the word embedding of the mask token for the masked numeral as a representation of the context.

## Limitations

One of the limitations of our study is that the performance of $k$NN search is highly dependent on the domain of the datastore used. As shown in Section 6.4, $k$NN search, like standard LM, does not work well for contexts and numerals for out-of-domain data. This dependence can be reduced by increasing the size of the datastore and introducing passages from various domains; however, this strategy may bolster another limitation, as discussed hereafter.

The second limitation is that $k$NN-LM requires more memory usage for the datastore and higher latency for search during inference compared with standard LMs. Although the search process itself can be executed swiftly by leveraging efficient similarity search libraries like Faiss (Johnson et al., 2017), as the size of the datastore expands, the time required to obtain their representation vectors is expected to increase.

The third limitation pertains to the lack of language variety in the utilized datasets. While we deliberately selected datasets from different domains for our experiments, they shared a common language, namely English. Consequently, it is expected that $k$NN-LM will exhibit similar effectiveness in languages with linguistic structures similar to English. However, conducting experiments on non-English datasets is necessary to provide evidence for the language-independent impact of $k$NN-LM. This aspect will be addressed in future research endeavors.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by JSPS KAKENHI Grant Numbers 21H03502 and 22K19818.

## References

Takeshi Abekawa and Akiko Aizawa. 2016. SideNoter: Scholarly paper browsing system based on PDF restructuring and text annotation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 136–140, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Chung-Chi Chen, Hen-Hsen Huang, Yow-Ting Shiue, and Hsin-Hsi Chen. 2018. Numeral understanding in financial tweets for fine-grained crowd-based forecasting. *CoRR*, abs/1809.05356.

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.

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.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training.

Zhihua Jin, Xin Jiang, Xingbo Wang, Qun Liu, Yong Wang, Xiaozhe Ren, and Huamin Qu. 2021. Numgpt: Improving numeracy ability of generative pre-trained models.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020a. Nearest neighbor machine translation.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020b. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6862–6868, Online. 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.

Yuxian Meng, Shi Zong, Xiaoya Li, Xiaofei Sun, Tianwei Zhang, Fei Wu, and Jiwei Li. 2021. Gnn-lm: Language modeling based on global contexts via gnn.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

OpenAI. 2023. Gpt-4 technical report.

Taku Sakamoto and Akiko Aizawa. 2021. Predicting numerals in natural language text using a language model considering the quantitative aspects of numerals. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 140–150, Online. Association for Computational Linguistics.

Georgios Spithourakis, Isabelle Augenstein, and Sebastian Riedel. 2016a. Numerically grounded language models for semantic error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 987–992, Austin, Texas. Association for Computational Linguistics.

Georgios Spithourakis, Steffen Petersen, and Sebastian Riedel. 2016b. Clinical text prediction with numerically grounded conditional language models. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pages 6–16, Auxtin, TX. 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.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Liyan Xu, and Lawrence Carin. 2022. Improving downstream task performance by treating numbers as entities.

Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 1017–1024, USA. Omnipress.

Avijit Thawani, Jay Pujara, and Filip Ilievski. 2021. 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.

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.

Shuhe Wang, Xiaoya Li, Yuxian Meng, Tianwei Zhang, Rongbin Ouyang, Jiwei Li, and Guoyin Wang. 2022. $k$nn-ner: Named entity recognition with nearest neighbor search.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yongwei Zhou, Junwei Bao, Chaoqun Duan, Haipeng Sun, Jiahui Liang, Yifan Wang, Jing Zhao, Youzheng Wu, Xiaodong He, and Tiejun Zhao. 2022. OPERA: Operation-pivoted discrete reasoning over text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1655–1666, Seattle, United States. Association for Computational Linguistics.

## A Experimental Setup

We used the Adam optimizer with learning rate and max-grad-norm set to $5 \times 10^{-5}$ and 1.0, respectively. All the words in the passages were tokenized with the BERT tokenizer; passages were then truncated to sequences of 512 tokens or less. In this study, only numerals expressed in arithmetic digits, such as "1" and "2022," were treated as target numerals to be predicted, and numerals expressed in English words, such as "one" and "ten," were not included.

## B Results of Context Range Comparison on ACLsent

The result of using different context ranges for $k$NN search on the ACLsent dataset is shown in Table 9. The same trends observed in the Numeracy-600K dataset were confirmed (Table 3).

## C Results of MNP Task on Other Datasets

Prediction results for $k$NN-LM on the FinNum and DROP datasets are shown in Tables 10 and 11. On the FinNum dataset, $k$NN search exhibited a better accuracy than LM without fine-tuning, and $k$NN search only or $k$NN-LM had the best accuracy in most settings with fine-tuning. This is the same trend observed in the Numeracy-600K and ACLsent datasets (Tables 4 and 5). By contrast, the results show a different trend for the DROP dataset. With fine-tuning, $k$NN search alone or $k$NN-LM almost always had the best accuracy in most settings, but without fine-tuning, the LM significantly outperformed $k$NN search. This may be because the DROP dataset differs from the other datasets in that each passage is longer (Table 1). When the passage is long, it is possible to check numerals other than the masked one in the same passage, and if there are answers or near-answer numerals among them, it can be solved as a simple reading comprehension task, which LMs perform well, without $k$NN search.

## D Results of MNP Task for OOV Numerals in ACLsent

Table 12 shows the top-$k$ accuracy of fine-tuned $k$NN-LM for numerals in and out of the BERT vocabulary in the ACLsent dataset. The same trends observed in the Numeracy-600K dataset were confirmed (Table 6).

| % of NE | Window size (n) | before and after the mask Top1↑ | Top3↑ | Top10↑ | only after the mask Top1↑ | Top3↑ | Top10↑ |
|---|---|---|---|---|---|---|---|
| $= 0\%$ | 0 (only [MASK]) | **27.8%** | **43.3%** | **56.5%** | **27.8%** | **43.3%** | **56.5%** |
| | 1 | 22.3% | 36.4% | 52.2% | 24.7% | 38.7% | 53.0% |
| | 2 | 23.0% | 36.5% | 52.6% | 22.4% | 36.5% | 51.4% |
| | 5 | 18.2% | 32.3% | 49.6% | 18.5% | 33.1% | 48.3% |
| | max length | 14.7% | 28.2% | 46.5% | 13.8% | 26.3% | 44.3% |
| $\leq 10\%$ | 0 (only [MASK]) | **37.9%** | **54.1%** | **70.4%** | **37.9%** | **54.1%** | **70.4%** |
| | 1 | 31.3% | 46.1% | 66.6% | 34.1% | 48.6% | 66.4% |
| | 2 | 32.9% | 47.5% | 66.3% | 32.7% | 46.6% | 66.2% |
| | 5 | 28.3% | 43.3% | 64.6% | 28.1% | 43.8% | 63.0% |
| | max length | 22.9% | 36.7% | 59.6% | 22.5% | 35.1% | 57.3% |

Table 9: Top-$k$ accuracy of $k$NN search on the ACLsent dataset when two different context ranges are used to compute the contextual representation: one with the mask and the $n$ words before and after it (Figure 3 (a)), and one with the mask and its subsequent $n$ words (Figure 3 (b)). "% of NE" indicates the percentage of numerical error allowed in each top-$k$ accuracy calculation.

| Method | % of NE | pre-trained Top1↑ | Top3↑ | Top5↑ | Top10↑ | fine-tuned Top1↑ | Top3↑ | Top5↑ | Top10↑ |
|---|---|---|---|---|---|---|---|---|---|
| $k$NN | $= 0\%$ | 14.3% | 23.1% | 28.3% | 37.1% | 17.2% | 27.5% | 33.0% | 39.6% |
| | $\leq 10\%$ | 19.6% | 32.9% | 41.5% | 54.9% | 23.5% | 38.3% | 46.5% | 57.8% |
| | $\leq 30\%$ | 27.3% | 46.6% | 59.3% | 76.0% | 31.9% | 53.1% | **64.0%** | **77.0%** |
| | $\leq 50\%$ | 37.0% | 61.1% | 74.0% | 87.7% | 41.8% | **66.1%** | **76.8%** | **87.6%** |
| LM | $= 0\%$ | 10.4% | 21.6% | 27.4% | 37.8% | 18.1% | 30.4% | 36.4% | 44.7% |
| | $\leq 10\%$ | 15.7% | 29.1% | 37.8% | 52.3% | 25.1% | 40.1% | 47.7% | 58.0% |
| | $\leq 30\%$ | 23.4% | 40.9% | 52.1% | 67.7% | 35.1% | 52.3% | 60.8% | 71.4% |
| | $\leq 50\%$ | 33.9% | 53.3% | 64.1% | 76.0% | 46.3% | 62.8% | 69.6% | 78.6% |
| $k$NN+LM | $= 0\%$ | 15.1% | 26.9% | 33.7% | 42.8% | **19.8%** | **30.8%** | **37.4%** | **46.2%** |
| | $\leq 10\%$ | 20.5% | 35.2% | 43.4% | 56.9% | **27.0%** | **41.3%** | **49.7%** | **60.7%** |
| | $\leq 30\%$ | 28.2% | 49.0% | 58.6% | 72.3% | **36.5%** | **54.3%** | 62.2% | 74.1% |
| | $\leq 50\%$ | 37.8% | 62.5% | 70.7% | 80.8% | **47.3%** | 65.7% | 72.4% | 81.5% |

Table 10: Top-$k$ accuracy of $k$NN-LM on the FinNum dataset.

| Method | % of NE | pre-trained Top1↑ | Top3↑ | Top5↑ | Top10↑ | fine-tuned Top1↑ | Top3↑ | Top5↑ | Top10↑ |
|---|---|---|---|---|---|---|---|---|---|
| $k$NN | $= 0\%$ | 8.0% | 14.7% | 19.1% | 27.7% | 13.8% | 22.5% | 27.8% | 36.7% |
| | $\leq 10\%$ | 32.9% | 45.6% | 53.1% | 64.8% | 37.7% | 50.4% | 57.8% | **68.8%** |
| | $\leq 30\%$ | 44.9% | 61.7% | 70.4% | 81.7% | 49.4% | 66.0% | **74.1%** | **83.7%** |
| | $\leq 50\%$ | 53.7% | 72.0% | 80.2% | 89.1% | 58.4% | 75.6% | **82.6%** | **90.1%** |
| LM | $= 0\%$ | 17.1% | 28.7% | 35.2% | 45.2% | 20.6% | 31.7% | 37.5% | 45.6% |
| | $\leq 10\%$ | 38.3% | 49.3% | 56.1% | 66.8% | **42.3%** | 53.0% | 58.4% | 66.0% |
| | $\leq 30\%$ | 48.0% | 61.9% | 68.8% | 77.2% | **53.7%** | 64.3% | 69.0% | 74.9% |
| | $\leq 50\%$ | 57.5% | 70.6% | 76.1% | 81.9% | 62.1% | 71.5% | 75.1% | 79.5% |
| $k$NN+LM | $= 0\%$ | 16.5% | 28.4% | 35.7% | 46.2% | **21.4%** | **32.0%** | **38.5%** | **47.8%** |
| | $\leq 10\%$ | 37.2% | 51.3% | 58.4% | 68.9% | 42.2% | **54.6%** | **60.6%** | 68.7% |
| | $\leq 30\%$ | 48.1% | 66.1% | 73.1% | 80.2% | 53.6% | **68.1%** | 72.9% | 78.7% |
| | $\leq 50\%$ | 56.9% | 75.7% | 80.8% | 85.3% | **62.2%** | **76.0%** | 79.5% | 83.6% |

Table 11: Top-$k$ accuracy of $k$NN-LM on the DROP dataset.

| Method | % of NE | Known numeral Top1↑ | Top3↑ | Top5↑ | Top10↑ | OOV numeral Top1↑ | Top3↑ | Top5↑ | Top10↑ |
|---|---|---|---|---|---|---|---|---|---|
| $k$NN | $= 0\%$ | 36.0% | 56.2% | 64.6% | 74.0% | **6.2%** | **9.3%** | **9.7%** | **10.5%** |
| | $\leq 10\%$ | 45.8% | 63.6% | 71.9% | 80.8% | **16.9%** | **29.1%** | **34.0%** | **43.1%** |
| | $\leq 30\%$ | 53.3% | 71.6% | 80.6% | 89.5% | **24.0%** | **42.7%** | **52.7%** | **66.5%** |
| | $\leq 50\%$ | 62.7% | 81.0% | 87.7% | **94.2%** | **31.8%** | **54.5%** | **64.7%** | **78.9%** |
| LM | $= 0\%$ | 40.8% | 62.4% | 70.7% | 78.8% | 1.9% | 2.9% | 4.0% | 5.0% |
| | $\leq 10\%$ | 50.0% | 68.6% | 76.2% | 84.2% | 8.0% | 14.7% | 19.5% | 25.1% |
| | $\leq 30\%$ | 57.8% | 74.9% | 82.3% | 88.8% | 13.6% | 25.1% | 30.4% | 37.1% |
| | $\leq 50\%$ | 66.7% | 82.4% | 87.1% | 92.6% | 19.8% | 32.9% | 39.1% | 44.7% |
| $k$NN+LM | $= 0\%$ | **42.1%** | **63.2%** | **72.1%** | **80.9%** | 3.4% | 7.7% | 8.3% | 9.9% |
| | $\leq 10\%$ | **51.5%** | **69.9%** | **77.7%** | **86.1%** | 11.8% | 24.6% | 28.6% | 36.9% |
| | $\leq 30\%$ | **58.6%** | **76.6%** | **84.0%** | **90.8%** | 18.5% | 35.8% | 42.7% | 50.5% |
| | $\leq 50\%$ | **67.7%** | **84.0%** | **88.5%** | 94.0% | 26.8% | 46.3% | 53.4% | 60.2% |

Table 12: Top-$k$ accuracy of the fine-tuned $k$NN-LM for numerals included in and out of the vocabulary numerals in the ACLsent dataset.

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*Limitations section*

☒ A2. Did you discuss any potential risks of your work?
*This paper is a foundational research. Our method uses a retrieval-based approach to improve the accuracy of the masked numeral prediction task. We do not think there is any potential risk in our work.*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Abstract and Section 1*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*Sections 4 and 5.1*

☑ B1. Did you cite the creators of artifacts you used?
*Sections 4 and 5.1*

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*We have confirmed the license of the artifacts used in our experiments.*

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*We have confirmed the intended use of the existing artifacts we used. We do not publish any new artifacts.*

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*The datasets we used are often used in studies in the NLP field, and we think that they do not contain such contents due to the way they were constructed.*

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Section 5.1 and Limitations section*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Section 5.1*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

**C ☑ Did you run computational experiments?**

*Sections 5, 6, and Appendix*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*Sections 5.2 and Appendix A*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*Section 5.2*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*Section 5.2*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*Sections 5.2 and Appendix A*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*No response.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*No response.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*No response.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*No response.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*No response.*