Predicting Numerals in Natural Language Text Using a Language Model Considering the Quantitative Aspects of Numerals

Taku Sakamoto[1] and **Akiko Aizawa**[2,1]
[1]The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
[2]National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{t_sakamoto, aizawa}@nii.ac.jp

Abstract

Numerical common sense (NCS) is necessary to fully understand natural language text that includes numerals. NCS is knowledge about the numerical features of objects in text, such as size, weight, or color. Existing neural language models treat numerals in a text as string tokens in the same way as other words. Therefore, they cannot reflect the quantitative aspects of numerals in the training process, making it difficult to learn NCS. In this paper, we measure the NCS acquired by existing neural language models using a masked numeral prediction task as an evaluation task. In this task, we use two evaluation metrics to evaluate the language models in terms of the symbolic and quantitative aspects of the numerals, respectively. We also propose methods to reflect not only the symbolic aspect but also the quantitative aspect of numerals in the training of language models, using a loss function that depends on the magnitudes of the numerals and a regression model for the masked numeral prediction task. Finally, we quantitatively evaluate our proposed approaches on four datasets with different properties using the two metrics. Compared with methods that use existing language models, the proposed methods reduce numerical absolute errors, although exact match accuracy was reduced. This result confirms that the proposed methods, which use the magnitudes of the numerals for model training, are an effective way for models to capture NCS.

1 Introduction

Numerical common sense (NCS) is knowledge about the numerical features of objects in the real world, such as size, weight, or color, each of which has its own range and probability distribution (Yamane et al., 2020). Consider the following example sentence.

"John is 200 *cm tall."*

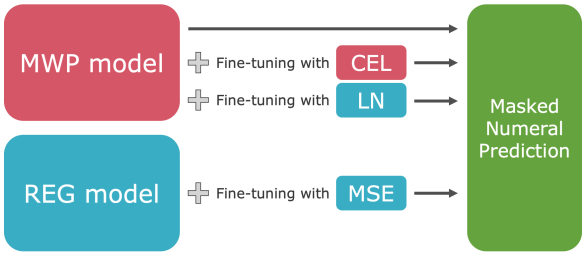


Figure 1 An overview of our proposed approaches for the masked numeral prediction task. We propose to use a new loss function $Loss_{\mathrm{NUM}}$ (LN) that is based on the magnitudes of numerals for fine tuning masked word prediction (MWP) model and a regression (REG) model that treats the masked numeral prediction as a regression task.

When we read this sentence, we can infer from it not only that John's height is *200 cm* but that John is a tall person. However, this kind of inference cannot be achieved by a system that does not have NCS about how tall people generally are. Therefore, it is essential to have knowledge about real-world numerical features for a deep understanding of natural language text containing numerals.

In recent years, BERT, GPT-3, and other neural language models have achieved a level of performance on par with or better than human performance in many natural language processing tasks (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Brown et al., 2020). Moreover, several studies have recently been conducted to investigate whether pre-trained neural language models have commonsense knowledge, and these studies often conclude that the language models have been successful in acquiring some commonsense knowledge (Petroni et al., 2019; Davison et al., 2019; Bouraoui et al., 2019; Zhou et al., 2019; Talmor et al., 2020).

However, it has also been reported that current neural language models still perform poorly in natural language processing tasks that require NCS and a deep understanding of numerals, such as numerical reasoning, numerical question answer-

Proceedings of Deep Learning Inside Out (DeeLIO):
The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 140–150

ing, and numerical error detection/correction (Dua et al., 2019; Chen et al., 2019). Numerals appear frequently in various forms, such as dates, numbers of people, percentages, and so on, regardless of the domain of passages. Hence, the acquisition of numerical common sense by neural language models and the analysis of the acquired numerical common sense are essential research topics to support systems for reasoning on text containing numerals and smooth conversation with humans at a high level.

One of the major problems that make it difficult for language models to understand the meaning of numerals and to acquire NCS is that naive language models treat numerals in text as string tokens, just like any other word (Spithourakis and Riedel, 2018). This makes it difficult to obtain a mapping between the string tokens and the magnitudes of the numerals, which is needed to capture NCS.

In this study, we use the masked numeral prediction task (Spithourakis and Riedel, 2018; Lin et al., 2020) to evaluate and verify the NCS acquired by neural language models. The task requires models to predict masked numerals in an input passage from their context. We use two types of evaluation metrics: hit@k accuracy (Lin et al., 2020) and MdAE and MdAPE (Spithourakis and Riedel, 2018) for this task. Hit@k accuracy calculates the percentage of predictions in which the groundtruth numeral is within the top k predicted numerals, and we can say that they evaluate language models in terms of the **symbolic** aspect of numerals. MdAE and MdAPE are calculated from the difference between the groundtruth numerals and the predicted numerals, and we can say that they evaluate language models in terms of the **quantitative** aspect of numerals.

To perform this task, we investigate the following two approaches to reflect the magnitudes of the numerals for fine-tuning language models on the masked numeral prediction task (Figure 1).

1. A masked word prediction model with a new loss function $Loss_{\mathrm{NUM}}$ that is based on the differences between the groundtruth numerals and predicted numerals;

2. A masked word prediction model, called the REG model, structured with an additional output layer to predict a numeral from an input passage containing a masked numeral.

We use the BERT-based masked word prediction model as a baseline and conducted experiments on four datasets, which differ from each other in the length and domain of the passages as well as the distribution and characteristics of the numerals appearing in the datasets. We compare the results and investigate the relationship between the characteristics of the numerals in the datasets and the performance of each method. Although fine-tuning with $Loss_{\mathrm{NUM}}$ causes a decrease in the exact match accuracy, we found that it reduces numerical absolute errors, which indicates the effectiveness of $Loss_{\mathrm{NUM}}$. The results of the REG model show the difficulty of predicting numerals in natural language text with the regression model. However, there were some numerals that the REG model predicted better than the existing language model, indicating that the REG model and existing language models are good at predicting numerals with different characteristics.

In our experiments, to eliminate the negative effects of the sub-word approach, we do not split the numerals into sub-words. The sub-word approach splits words into shorter tokens called sub-words. It has the advantage that even low-frequency words can be represented by a combination of sub-words that appear in a text more frequently. However, unlike the case of words, sub-words derived from numerals often have little relationship to the meaning of the original numerals, which can make it difficult to understand the meaning of numerals (Wallace et al., 2019). All other words are separated into sub-words in our experiments.

To summarize, in this work, we tackle the problem of dealing with numerals in naive language models on the masked numeral prediction task. Our contributions are as follows:

- We use two evaluation metrics (exact match accuracy and numerical absolute errors) for the masked numeral prediction task to evaluate the language models in terms of the symbolic and quantitative aspects of the numerals, respectively.

- We propose a new loss function to reflect not only the symbolic aspect but also the quantitative aspect of numerals in the training of language models. For the masked numeral prediction task, we also employ a regression model, which predicts numerals as quantities.

- We quantitatively evaluate our proposed approaches on four datasets with different properties using the two metrics. The reduction

in the numerical absolute errors of the predictions confirms the effectiveness of our proposed approaches.

2 Related Work

2.1 Masked Numeral Prediction

Masked numeral prediction is the task of predicting a masked numeral in an input passage from the context (e.g., "The movie I saw yesterday was [MASK] minutes long.") It can be used as an indicator to evaluate the NCS acquired by language models.

Lin et al. (2020) analyzed NCS captured by current language models using a masked numeral prediction task in which masked numerals were limited to numerals that could be uniquely determined, such as "A car usually has [MASK] wheels." They showed that even the current best pre-trained language models still perform poorly compared to humans on the task, which requires NCS. They also found that even though pre-trained language models seemingly make the correct predictions, the models are often unable to maintain the correct answer under even small changes, for instance, if the above target sentence changes to "A car usually has [MASK] *round* wheels."

Spithourakis and Riedel (2018) examined numeracy of neural language models using the masked numeral prediction task. Numeracy refers to the ability to understand the meanings of numerals and to deal with them properly. They conducted their experiments on scientific paper and clinical text datasets that include many numerals that represent the quantities of something. To improve the prediction accuracy for such numerals, they proposed a method that uses character-level recurrent neural networks (Graves, 2013; Sutskever et al., 2011) for prediction, a method that predicts the distribution of the numerals as a mixture of Gaussian distributions, and an ensemble method of these methods. They showed that the accuracy of the prediction of quantity-like numerals can be improved by methods that consider the magnitudes of the numerals.

2.2 Natural Language Processing Tasks That Involve Numerals

2.2.1 Machine Reading Comprehension Requiring NCS

Dua et al. (2019) created a machine reading comprehension dataset called DROP that contains questions requiring numerical operations such as addition, subtraction, and sorting to answer correctly. They used the DROP dataset to evaluate current machine reading comprehension models and showed that many questions requiring only simple numerical operations easily solved by humans cannot be answered correctly by current models. To improve the performance of the models on the DROP dataset, Hu et al. (2019) built a specialized architecture for numerical operations and achieved a significant improvement in accuracy, although not to human level. In contrast, Geva et al. (2020) showed that even if they use a generative model that is not specialized for numerical operations, they can improve the performance on DROP using additional data for numerical operation training. In our experiments, we use the passages in the DROP dataset for the masked numeral prediction task.

2.2.2 Numerical Error Detection

Numerical error detection is the task of determining whether or not numerals in input passages are errors (Chen et al., 2019; Spithourakis et al., 2016). To determine if a target numeral is an error, it is necessary to have knowledge of the range of values that the numeral can and cannot take. For example, to notice numerical errors in sentences with dates (for example, "Her birthday is December -3." or "Her birthday is December 20.5."), it is necessary to know that the range of possible values for numerals representing dates is generally an integer between 1 and 31. Therefore, the accuracy of numerical error detection can be used to quantitatively evaluate the NCS of the detection models. Chen et al. (2019) experimented with the BiGRU model to detect numerals multiplied by a random factor in Numeracy-600K, which is a dataset of market comments. They showed that the BiGRU model was able to detect numerical errors with less than 60% accuracy even with small numeral changes of approximately 10%. Moreover, it achieved an accuracy of only 76% even with large numeral changes of approximately 90%. In our experiments, we use the article titles from this dataset as one of the datasets for the masked numeral prediction task.

2.2.3 Numeral Type Prediction

Numeral type prediction is the task of classifying numerals in text into one of several fixed categories. Prediction models are required to classify numerals using their numerical values and contexts. Chen et al. (2018) aimed to understand the meanings of numerals in financial tweets for crowd-based

Token-type NCS	• Spiders have **8** legs. • A week has **7** days.
Quantity-type NCS	• The adult male is approximately **170** cm tall. • The length of movies is about **120** minutes.

Table 1 Two types of NCS.

forecasting, providing the dataset FinNum, which contains financial tweets in which numerals are annotated with their categories. Their categories include “Monetary," “Percentage," “Temporal" (date and time), and so on. They used a convolutional neural network (CNN), long short-term memory (LSTM), and bidirectional LSTM in experiments and concluded that character-level CNN performed the best. We use the FinNum dataset in our experiments for the masked numeral prediction task.

3 NCS

3.1 Two Types of NCS

NCS is the knowledge about numerical features of objects in the real world, such as size, weight, and price. NCS is required to understand natural language text that includes numerals or that refers to the real-world numerical features of some objects. We focus on the fact that numerals have two aspects, symbolic and quantitative, and hypothesize that there are two types of NCS: token type and quantity type (Table 1).

Token-type NCS refers to numerical knowledge involving numerals that can be appropriately understood as string tokens. This knowledge is definition-like or rule-like knowledge that cannot use other numerals instead, like “A week has 7 days." (Table 1). This kind of NCS is relatively easy to learn, even with conventional language models that treat numerals as string tokens in the same way as other words. Related work on the evaluation and analysis of token-type NCS acquired by current neural language models was reviewed in Section 2.1.

Quantity-type NCS refers to knowledge of numerical properties that have some kind of distribution or range, like “The adult male is approximately 170 cm tall." (Table 1). To acquire this kind of NCS, it is necessary to understand numerals as not only string tokens, but also quantities. Quantity-type NCS is more important for numerical error detection/correction and numerical reasoning than the token-type NCS. In recent years, there has been an increasing amount of research on the acquisition of quantity-type NCS, including the creation of datasets that collect the distributions of some attributes such as weight, length, and price of common objects as well as the verification of such NCS acquired by neural language models using these datasets (Elazar et al., 2019; Zhang et al., 2020; Yamane et al., 2020). In this paper, we aimed to acquire quantity-type NCS as well as token-type NCS with language models, focusing on the fact that there are these two types of NCS.

3.2 Masked Numeral Prediction

3.2.1 Task Description

Masked numeral prediction is the task of predicting a masked numeral in an input natural language text from the words around the masked numeral (e.g., “The movie I saw yesterday was [MASK] minutes long.") (Spithourakis and Riedel, 2018; Lin et al., 2020). In this paper, we use this task as an indicator to evaluate the NCS acquired by language models.

The masked numeral prediction 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

Language models take a passage that contains exactly one masked numeral as input, predict the numerals that could replace the mask token from the context words, and return the predicted numerals in the form of a ranking. The aim of the language models is to predict numerals that are closer to the groundtruth numerals. In the task considered in this paper, the target numerals are limited to numerals in arithmetic form such as “3.14" and “1,000," and numerical words such as “five" or “twenty" are not considered. For negative numerals, only the parts excluding signs were treated as target numerals; the signs were treated as context words (for example, in the case of the negative numeral “-10," only “10" was masked as the target numeral). For fractions, the denominator and numerator were treated as two different numerals in training and evaluation (e.g., the fraction “2/3" was masked in two ways: “[MASK]/3” and “2/[MASK]”).

3.2.2 Evaluation Metrics

Exact Match Accuracy A masked numeral prediction model generates a probability distribution over its vocabulary of numeral tokens using a softmax function and returns a ranking of them for each

mask. *Hit@k accuracy* calculates the percentage of predictions in which the groundtruth numeral is within the top k predicted numerals from the generated ranking (Lin et al., 2020). In our experiments, we used $k = 1, 3$, and 10 for evaluation.

Numerical Absolute Error The hit@k accuracy metric simply evaluates whether the groundtruth numerals are included in the top k predictions. It does not take into account how close the predicted numerals are to the groundtruth numerals. However, in the masked numeral prediction task, a prediction for a mask that is closer to the groundtruth numeral is generally considered to be a better prediction, even if it is incorrect, so we need an additional evaluation metric to evaluate language models in terms of the magnitude of the difference between the groundtruth numeral and the predicted numeral.

Therefore, in the evaluation in this paper, following a previous work (Spithourakis and Riedel, 2018), we use the *median absolute error* (MdAE) and *median absolute percentage error* (MdAPE). MdAE and MdAPE are commonly used to evaluate regression models. They evaluate closeness on the number line between groundtruth numerals and predicted numerals (Spithourakis and Riedel, 2018). We can say that they evaluate language models in terms of the quantitative aspects of numerals. MdAE and MdAPE are calculated as follows:

$$\mathrm{MdAE} = \mathrm{median}\{|\mathrm{ans}_i - \mathrm{pred}_i|\} \quad (1)$$

$$\mathrm{MdAPE} = \mathrm{median}\left\{\left|\frac{\mathrm{ans}_i - \mathrm{pred}_i}{\mathrm{ans}_i}\right|\right\} \quad (2)$$

where ans_i is the magnitude of a groundtruth numeral, pred_i is the magnitude of a predicted numeral, and N is the number of masked numerals.

4 Approach

4.1 $Loss_{\mathrm{NUM}}$

Naive masked word prediction (MWP) models return a probability distribution over their vocabulary (only numeric words) and they are trained using the cross entropy loss between their outputs and the distribution of the correct answers as a loss function. The usual cross entropy loss treats each token in the vocabulary except for the correct answer equally. However, in the case of the masked numeral prediction task, we are motivated to train language models with a loss function that yields a smaller error for predictions that are numerically closer to the groundtruth numeral and a larger error for predictions that are further away. This is because it is generally considered that a prediction of "9" is better than a prediction of "1" for a mask for which the correct answer is "10." Therefore, in this paper, we propose a loss function $Loss_{\mathrm{NUM}}$, that depends on the magnitudes of the numerals for fine-tuning MWP models.

$Loss_{\mathrm{NUM}}$ is defined as follows:

$$Loss_{\mathrm{NUM}} = \sum_{i=1}^{N} \{(\log(\mathrm{ans}_i) - \log(\mathrm{pred}_i))^2 \times \mathrm{CEL}_i\} \quad (3)$$

where ans_i is the numerical magnitude of a groundtruth numeral, pred_i is the magnitude of the initial numeral predicted by the MWP model, N is the number of masked numerals, and CEL_i is the cross entropy loss calculated for the i-th masked numeral. $Loss_{\mathrm{NUM}}$ is computed using the logarithmic differences between the groundtruth numerals and predicted numerals following the treatment of numerical errors in a previous study (Geva et al., 2020). This is because the logarithmic difference gives more weight to off-by-one errors in small numerals, which are considered to be more fatal than off-by-one errors in large numerals. These differences are then multiplied by the usual cross entropy loss to obtain the $Loss_{\mathrm{NUM}}$. If it is used when fine-tuning pre-trained language models, we expect that the models will be fine-tuned to return numeral tokens that are numerically closer to the groundtruth numerals.

4.2 REG Model

The approach described in Section 4.1 uses ordinary MWP models and the proposed loss, which reflects the magnitudes of the predicted and groundtruth numerals as the loss function during fine tuning. In this section, we propose to use a regression (REG) model for the masked numeral prediction task.

The REG model is structured with an additional numeric output layer as the final layer of BERT. The output layer generates a single numeral between 0 and MAX_NUM from an input passage processed by BERT, where MAX_NUM is the largest numeral occurring in training data. The mean squared error between groundtruth numerals and predicted numerals, which is often used as a loss function and an evaluation metric in regression

tasks, is adopted as the loss function ($Loss_{\mathrm{MSE}}$) for fine-tuning the REG model on the masked numeral prediction task. Similarly to the calculation of $Loss_{\mathrm{NUM}}$, $Loss_{\mathrm{MSE}}$ is calculated using the logarithmic values of both the groundtruth numerals and predicted numerals to give more weight to off-by-one errors in small numerals, which are considered to be more fatal than off-by-one errors in large numerals.

$$Loss_{\mathrm{MSE}} = \sum_{i=1}^{N} (\log(\mathrm{ans}_i) - \log(\mathrm{pred}_i))^2 \quad (4)$$

where ans_i is the numerical magnitude of a groundtruth numeral, pred_i is the magnitude of the initial numeral predicted by the REG model, and N is the number of masked numerals. For the evaluation, which includes exact match accuracy, the final output numeral is rounded to the nearest integer and is used as the initial predicted numeral. Next, the integers closest to the first predicted numeral are used as the second predicted numeral, the third predicted numeral, and so on, in order of closeness.

5 Experiments

5.1 Dataset

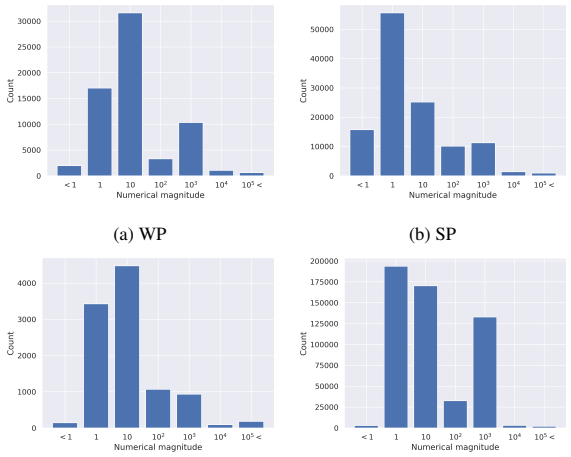
In our experiments, we used four datasets, DROP (Wikipedia) (Dua et al., 2019), arXiv (Science Papers) (Spithourakis and Riedel, 2018), FinNum (Financial Tweets) (Chen et al., 2018), and Numeracy-600K (Article Titles) (Chen et al., 2019). The data in these datasets differ in passage length, the domain of the passages, and the distribution of the numerals that appear in the datasets. These datasets were created and used for different numerical tasks such as numerical machine reading comprehension and numeral type prediction (Section 2). We use them for the masked numeral prediction in this work. We denote these datasets as “WP," “SP," “FT," and “AT," respectively.

Statistics about the passages and numerals in these four datasets are listed in Table 2. The percentage of numerals that appear only in each dataset (“% of one-time numerals"), the number of different numerals that appear in a dataset (“Variety of numerals"), and the number of numerals that appear more than once in the same passage (“Numeral duplication") are given. Every passage in all four datasets contains one or more numerals.

WP and SP have relatively long passages, and prediction models can make predictions based on hints from unmasked numerals around the masked numeral. In contrast, FT and AT have shorter passages, so there are fewer unmasked numerals in the same passage. In addition, WP and AT tend to contain more token-type numerals such as dates, years, and game scores, whereas SP and FT tend to contain more quantity-type numerals such as the scores of experimental results and stock prices.

Statistic	WP	SP	FT	AT
Number of passages	4,329	14,821	3,992	420,000
Ave. passage len [tokens]	281.8	278.2	36.7	12.9
Number of numerals	65,783	120,105	10,312	537,214
% of integers	96.4%	80.0%	86.7%	98.5%
% of one-time numerals	2.93%	3.69%	9.74%	0.36%
Variety of numerals	3,667	7,944	1,503	4,204
Numeral duplication	25,269	57,372	1,354	22,555
Mean	1.57e6	5.55e16	2.02e15	2.98e7
Median	24.0	5.0	20.0	17.0

Table 2 Dataset statistics across different four datasets (training set).



(a) WP (b) SP

(c) FT (d) AT

Figure 2 Distribution of the numerals in the training data.

The distribution of the numerals in each dataset is shown in Figure 2. The x-axis of each figure shows, from left to right, the counts of numerals less than 1, numerals between 1 and 10, ..., numerals between 10,000 and 100,000, and numerals greater than 100,000. We can see from this figure that WP and AT certainly contain many years, and thus the proportion of four-digit numerals in WP and AT is higher than in the other datasets, and FT has more numerals with six or more digits to represent high amounts of money.

5.2 Experimental Setup

In the experiments, we used the BERT-based MWP model as the baseline model. It consists of the BERT model with an additional softmax layer as

the final layer. Given an input passage processed by BERT, the softmax layer outputs the probability distribution over the model's vocabulary of numeric words. Each mask in a passage can be filled with a single numeric word, and the numeric vocabulary contains not numerals expressed in English words such as "ten" and "twenty-four" but numerals expressed in arithmetic characters such as "10," "2021," and "10,000."

In this experiment, we used the Adam optimizer with a learning rate of 5×10^{-5}. The batch size for fine-tuning and evaluation was 32 and the max-grad-norm was 1.0. All tokens except the numerals in the passages were tokenized by the BERT tokenizer and passages were truncated to sequences no longer than 512 tokens.

In this evaluation, we did not split numerals into sub-words using BERT but treated them as single words using our own additional rules. By treating numerals as single words, we believe that it becomes easier to learn mappings between strings of numerals and their corresponding numerical magnitudes, which is difficult to learn from sub-word segmented numerals. The single word segmentation of numerals also eliminates the need to use encoder–decoder models or other methods to predict sub-word sequences for masks when predicting numerals, which has the advantage of making the masked numeral prediction task easier to handle, even for naive MWP models.

6 Result and Discussion

6.1 $Loss_{\mathrm{NUM}}$

Table 3 shows the result of the naive BERT-based MWP model with pre-training but without fine-tuning (MWP), fine-tuned MWP with cross entropy loss (Ft. MWP w/ CEL), and fine-tuned MWP with $Loss_{\mathrm{NUM}}$ (3) (Ft. MWP w/ LN). Each dataset is divided into three parts: training set, validation set, and test set. Each fine-tuned model is fine-tuned first on the training set and then on the validation set of the corresponding dataset, and then it is evaluated on the test set of the same dataset.

First, comparing MWP and Ft. MWP w/ CEL, we can see that the scores of all metrics have been improved by fine-tuning for all datasets. Moreover, the increases in the scores obtained on FT and AT are substantially larger than those obtained on WP and SP. This is probably because the average passage lengths of WP and SP are longer than those of FT and AT, and the language models succeeded in predicting masked numerals in WP and SP to some extent from context words and surrounding unmasked numerals without fine-tuning (Table 2).

Dataset	Approach	hit@1↑	hit@3↑	hit@10↑	MdAE↓	MdAPE↓
WP	MWP	23.8	32.1	45.0	7.0	42.9
	Ft.MWP w/ CEL	**28.5**	36.6	49.0	**5.4**	**25.0**
	Ft.MWP w/ LN	**28.5**	**37.2**	**50.2**	6.0	28.6
SP	MWP	40.1	50.2	63.1	2.0	50.0
	Ft.MWP w/ CEL	45.5	55.5	67.7	**1.0**	33.3
	Ft.MWP w/ LN	**48.4**	**57.6**	**69.2**	**1.0**	**25.5**
FT	MWP	19.9	27.8	43.2	10.0	85.1
	Ft.MWP w/ CEL	**40.5**	**49.1**	**60.0**	**3.0**	50.0
	Ft.MWP w/ LN	40.0	48.2	59.4	**3.0**	**46.7**
AT	MWP	20.1	32.7	54.7	7.0	80.0
	Ft.MWP w/ CEL	**56.3**	**69.1**	**80.4**	**1.0**	**0.0994**
	Ft.MWP w/ LN	55.7	68.5	80.0	**1.0**	0.0995

Table 3 Hit@k accuracy (%), MdAE, and MdAPE (%) of the BERT-based MWP models on the four datasets.

Next, we compare Ft.MWP w/ CEL and Ft. MWP w/ LN. Focusing on the MdAE and MdAPE scores, it is confirmed that the reduction in the numerical absolute errors of the predictions, which is the objective of the proposed loss function $Loss_{\mathrm{NUM}}$, is achieved on the SP and FT datasets. In contrast, the MdAE and MdAPE scores of the WP and AT datasets increased. This may be due to the different nature of the numerals in these datasets. Because of the nature of the domains of these datasets, the WP and AT datasets contain many numerals that are better understood as string tokens, such as years, dates, and football game scores. Hence, fine-tuning with $Loss_{\mathrm{NUM}}$ does not improve the accuracy of masked numeral prediction in these datasets. In contrast, the SP and FT datasets contain more numerals that are better understood as quantities, such as the numerals representing scores of experimental results or detailed amounts of money, and it is thought that reflecting the magnitudes of these numerals in model training improves the prediction accuracy in SP and FT.[1] The proposed loss function $Loss_{\mathrm{NUM}}$, which is intended to help language models understand the magnitudes of the numerals and reduce the numerical absolute errors, also leads to a small but significant improvement in the hit@k accuracies on some datasets.

Passages **a)** and **b)** in Table 4 are examples where the MWP model fine-tuned with the cross entropy loss made largely incorrect predictions. Passage **a)** shows predictions in a context where it can be

[1] The percentage of integers in the dataset and the distribution of the numerals can also reveal the trend of the numerals in the dataset (Table 2, Figure 2).

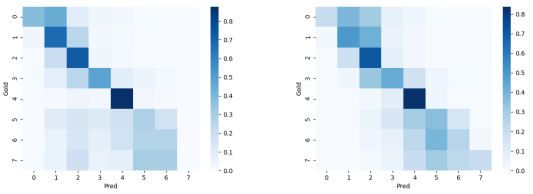
inferred that the masked numeral is greater than 1724 and not much larger than 1724. The MWP model fine-tuned with the cross entropy loss returned “1925," which is numerically far from the groundtruth numeral, although it is considered to be a numeral representing a year. In contrast, the MWP model fine-tuned with $Loss_{\mathrm{NUM}}$ returned “1727," which is not correct, but is above 1724 and not far from 1724. Note that “1925" and “1727" do not appear in the context passage, and the models chose these numerals out of their respective vocabularies. In passage **b)**, the context suggests that the masked numeral is considered to be a numeral representing a percentage between 0 and 100 (more specifically, between 75.6 and 100) from its context. However, for this mask, the MWP model fine-tuned with the cross entropy loss predicted “50,000," which substantially exceeds 100. In contrast, the MWP model fine-tuned with $Loss_{\mathrm{NUM}}$ successfully predicted a numeral less than 100, although it should be greater than 75.6. These are successful examples where language models were fine-tuned to predict numerals that are numerically close to the groundtruth numerals by fine-tuning them with $Loss_{\mathrm{NUM}}$, which imposes large penalties on numerically large errors. In some cases, fine-tuning language models with $Loss_{\mathrm{NUM}}$ caused them to fail to predict numerals that the models fine-tuned with the cross entropy loss predicted correctly. This could also cause them to predict numerals that were rather far from the groundtruth numerals.

6.2 REG Model

In this section, we compare and analyze prediction results of the naive BERT-based MWP model fine-tuned with the cross entropy loss and the REG model fine-tuned with $Loss_{\mathrm{MSE}}$ (4). We used the WP dataset to train and evaluate them.

The results of the fine-tuned MWP model with the cross entropy loss (Ft. MWP w/ CEL) and the fine-tuned REG model with $Loss_{\mathrm{MSE}}$ (Ft. REG w/ MSE) listed in Table 5 reveal that the REG model is substantially inferior to the MWP model with respect to prediction accuracy. [2] However, the REG model can predict large numerals better and has fewer large errors, indicating that the two models are good at predicting numerals with different characteristics (Figure 3). Figure 3 shows heat maps

[2]Note that the difference between the scores of “Ft. MWP w/ CEL" in WP on Table 3 and the scores of “Ft. MWP w/ CEL" on Table 5 is because the models in Table 5 are trained on half of the Wikipedia dataset.



(a) Fine-tuned MWP model (b) Fine-tuned REG model

Figure 3 Confusion matrices of the digits of the groundtruth numerals and the predicted numerals from the two models.

representing confusion matrices of the groundtruth numerals and the numerals predicted by the two models. The numerals are classified by the number of digits. In both heat maps, the y axis is the number of digits of the groundtruth numerals and the x-axis is that of predicted numerals. The darker the blue, the higher the percentage of numerals belonging to the corresponding cell in each row.

The percentage of substantially incorrect predictions that differ by more than one, two, and three digits from the groundtruth numerals are respectively 8.5%, 3.4%, and 1.5% for the MWP model, whereas they are significantly lower, that is, 7.7%, 1.8%, and 0.4% for the REG model (Table 6). This indicates that the overall prediction accuracies of the REG model are quite low, and for many numerals, the MWP model can provide better predictions. However, there are certain numerals that the REG model can predict more accurately than the MWP model. Moreover, the confusion matrices also indicate that the REG model is more suitable for predicting large numerals than the MWP model, suggesting that the MWP and REG models are good at predicting different types of numerals.

Table 4 shows examples of incorrect predictions made by the MWP models and the REG model. Passage **c)** is an example where the REG model made better predictions for a large numeral than did the MWP models. The reason why the MWP models predicted “94.7" and “10.7" is that the context in which the word “census" appears in the training data has many occurrences of numerals that represent percentages (including “94.7" and “10.7"), such as the percentage of population by age. From these results, it can be seen that the MWP models basically do not understand the magnitude of the numerals and learn relationships between numerals as string tokens and context words. Passage **d)** shows that the MWP models are effective in predicting a masked numeral where the groundtruth numeral also appears elsewhere in the passage.

Passage	Ans	CEL	LN	REG
a) Captain John Lovewell made three expeditions against the Indians. On the first expedition in December **1724**, he and his militia company of **30** men left Dunstable, . . . On December **10**, **1724**, they and a company of rangers killed two Abenakis. In February **[MASK]**, Lovewell made a second expedition to the Lake Winnipesaukee area. . . .	1725	1925	1727	762.0
b) Houston is considered an Automobile dependency city, with an estimated **[MASK]**% of commuters driving alone to work in **2016**, up from **71.7**% in **1990** and **75.6**% in **2009**. . . .	77.2	50,000	11	12.0
c) As of the census of **2010**, there were **[MASK]** people, **140,602** households, and **114,350** families residing in the county. . . .	516,564	94.7	10.7	118523.0
d) In September **1941**, Partisans organized sabotage at the General Post Office in Zagreb. . . . In November **[MASK]**, German troops attacked and reoccupied this territory, with the majority of Partisan forces escaping towards Bosnia. . . .	1941	1941	1941	1287.0

Table 4 Examples of incorrect predictions in the WP dataset. We list the context passages containing one masked numerals ("Passage"), the groundtruth numerals ("Ans") and the numerals predicted by the MWP model fine-tuned with the cross entropy loss ("CEL"), by the MWP model fine-tuned with $Loss_{\mathrm{NUM}}$ ("LN"), and by the REG model fine-tuned with $Loss_{\mathrm{MSE}}$ ("REG").

Model	hit@1↑	hit@3↑	hit@10↑	MdAE↓	MdAPE↓
Ft.MWP w/ CEL	27.4	35.8	48.6	6.0	28.6
Ft.REG w/ MSE	4.19	7.52	15.2	54.0	60.0

Table 5 Hit@k accuracy (%), MdAE, and MdAPE (%) of two approaches on the WP dataset.

Model	±2~ digits	±3~ digits	±4~ digits
Ft.MWP w/ CEL	8.5%	3.4%	1.5%
Ft.REG w/ MSE	**7.7**%	**1.8**%	**0.4**%

Table 6 Percentages of substantially incorrect predictions of the MWP and REG model.

6.3 Future Work

MdAE, which uses the numerical absolute errors between predicted numerals and groundtruth numerals, is sensitive to the scale of the data and is easily affected by the prediction accuracy for large numerals in a dataset that contains numerals of different scales and types. MdAPE, which evaluates absolute percentage errors, imposes large penalties on the overestimation of masked numerals. For example, a prediction of "1" for "31" in a sentence "Today is October 31." and a prediction of "31" for "1" in a sentence "Today is October 1." should both be equally wrong, but the former results in an error of approximately $\frac{|31-1|}{31} \times 100 \approx 100\%$, whereas the latter results in an error of $\frac{|1-31|}{1} \times 100 = 3000\%$. Because of these problems, there is room for consideration of the appropriate evaluation metrics for the masked numeral prediction task.

Although the REG model has a lower prediction accuracy than existing language models, there are certain numerals that the REG model can predict more accurately than the MWP model. This implies that the overall prediction accuracy can be improved by using the MWP model and the REG model differently depending on the target numerals. Such a combination method is also one task for future work.

7 Conclusion

In this paper, we used the exact match accuracy and numerical absolute errors metrics to evaluate the masked numerical prediction task, focusing on the fact that numerals have two aspects: symbolic and quantitative. Based on this fact, we proposed two methods to reflect the two aspects of numerals in the training of language models. Although the proposed loss function, $Loss_{\mathrm{NUM}}$, decreased the exact match accuracy slightly, it also reduced the numerical absolute errors on the masked numeral prediction task, indicating the effectiveness of $Loss_{\mathrm{NUM}}$. Furthermore, we analyzed the relationship between the properties of numerals in datasets and the performances of different prediction methods on four datasets with different properties. As a result, it was found that the types of numerals that are likely to be mistakenly predicted depend on which method is used.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by JSPS KAKENHI Grant Number 21H03502 and SIP2 NEDO Program "Big-data and AI-enabled Cyberspace Technologies."

References

Zied Bouraoui, José Camacho-Collados, and Steven Schockaert. 2019. Inducing relational knowledge from BERT. *CoRR*, abs/1911.12753.

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.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178, Hong Kong, China. 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.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. A multi-type multi-span network for reading comprehension that requires discrete reasoning. 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 1596–1606, Hong Kong, China. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language 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.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Georgios Spithourakis, Isabelle Augenstein, and Sebastian Riedel. 2016. 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 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.

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.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

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.

Hiroaki Yamane, Chin-Yew Lin, and Tatsuya Harada. 2020. Measuring numerical common sense: Is a word embedding approach effective?

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.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2019. Evaluating commonsense in pre-trained language models. *CoRR*, abs/1911.11931.