Pre-training and Evaluation of Numeracy-oriented Language Model

Fuli Feng[12*], Xilin Rui[3], Wenjie Wang[2], Yixin Cao[4], Tat-Seng Chua[2]
[1]Sea-NExT Joint Lab, [2]National University of Singapore,
[3]Tsinghua University, [4]Nanyang Technological University
{fulifeng93,ruixilin98,wenjiewang96,caoyixin2011}@gmail.com,dcscts@nus.edu.sg

ABSTRACT

Pre-trained language model (LM) has led to significant performance gains in various natural language processing (NLP) applications due to its strong literacy, *e.g.,* the ability to capture word dependencies. However, the existing pre-trained LMs largely ignore numeracy, *i.e.,* treating numbers within text as plain words and without understanding the basic numerical concepts. The weak numeracy has become a barrier to the use of pre-trained LMs in NLP applications over financial documents such as annual filings and analyst reports that are number intensive. However, the understanding and analysis of financial documents are becoming gradationally important. To bridge this gap, this work explores the central theme of numerical pre-training to empower LM with numeracy. In particular, we propose two numerical pre-training methods with objectives that encourage the LM to understand the magnitude and value of numbers and encode the dependency between a number and its context. By applying the proposed methods on BERT, we pre-train two LMs, named *BERT-M* and *BERT-V*. Moreover, we construct four datasets of financial documents for evaluating the numeracy of pre-trained LM, which focus on three fundamental perspectives of numeracy: a) number embedding; b) number-text composition; and c) number-number composition. Extensive experiments on the datasets validate the effectiveness of the pre-trained BERT-M and BERT-V, which outperform the state-of-the-art LM for financial documents (FinBERT) by 4.83% and 4.34% on average. Furthermore, their aggregation named *BERT-MV* increases the gain to 10.88%.

CCS CONCEPTS

• **Information systems** → **Document representation**; • **Computing methodologies** → **Learning latent representations**.

KEYWORDS

representation learning, language model, numeracy

ACM Reference Format:
Fuli Feng, Xilin Rui, Wenjie Wang, Yixin Cao, Tat-Seng Chua. 2021. Pre-training and Evaluation of Numeracy-oriented Language Model. In *2nd ACM International Conference on AI in Finance (ICAIF'21), November 3–5, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3490354.3494412

*Corresponding author. This research is supported by the Sea-NExT Joint Lab.

ICAIF'21, November 3–5, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-9148-1/21/11…$15.00
https://doi.org/10.1145/3490354.3494412

1 INTRODUCTION

Pre-trained LM such as BERT [9] has become a widely used backbone in various NLP applications such as document retrieval [7, 19, 23] and question answering [21, 40]. Pre-trained LM aims to encode a document to capture contextual semantics from word co-occurrence through self-supervised training over large-scale unlabeled corpus [16]. As existing LMs treat numbers within the document equally as plain words, they cannot recover the numerical information and infer the semantics beyond the surface form of co-occurrence [44]. Take annual reports as an example, given the sentence "the revenue increases 3% while the expectation is 4.5%", the model may fail to capture the gap between real increase and expectation, and cannot identify the document as an influential negative signal for the investment on the relevant stock. As such, the existing pre-trained LM is insufficient for constructing NLP solutions for financial applications.

In this work, we explore the central theme of numerical pre-training of LM that empowers the model with numeracy so as to understand the numerical information within documents. According to the common measure of numeracy for human beings [2], we highlight that the LM should achieve the following three fundamental objectives[1]: 1) *Understanding Number*, which is a basic requirement of numeracy to compare and discriminate numbers. 2) *Number-Text Composition*, which is the ability to capture the connection between a number and its surrounding text. That is what number would be typically used within a context, *e.g.,* the number following "revenue increase" is typically at the magnitude of 10e-2 or 10e-1. 3) *Number-Number Composition*, which is the ability to further capture the connection between different numbers within the document, *e.g.,* the real revenue increase (3%) is smaller than the expectation (4.5%) by a large percentage.

Towards this end, we propose two numerical pre-training methods, *magnitude model* and *value model*, inspired by the widely used pre-training objective *masked LM* [46]. In particular, 1) *magnitude model* randomly masks number in the document, and predicts the magnitude of the masked number based on its context. Similarly, 2) *value model* predicts the value of the masked numeric token based on its context. Both objectives encourage the LM to encode the connection between number and its context, especially the numbers within the context, which naturally forces the LM to understand the magnitude and value of the numbers. By applying

[1]We ignore high-level numeracy skills such as understanding math equations, which are typically achieved by additional components over LM [1].

the proposed methods to BERT, we pre-train two language models, named *BERT-M* and *BERT-V*, respectively.

To evaluate the numeracy of pre-trained LMs, we further construct four datasets of financial documents for tasks of *Number Forecasting*, *Magnitude Inference*, *Numerical Decoding*, and *Numerical Fact Checking*, respectively. Prediction performance on these datasets reveals the quality of number embedding, number-text composition, and number-number composition, which is thus able to bridge the research gap of numeracy evaluation. Extensive experiments on the four datasets show that the numeracy of BERT can be enhanced by further pre-training over the number intensive corpus (*i.e.,* FinBERT [8]). Furthermore, across the datasets, BERT-M and BERT-V outperform FinBERT by 4.83% and 4.34% on average, validating the effectiveness of the proposed numerical pre-training methods. Lastly, the performance gain can be increased to 10.88% by simply aggregating BERT-M and BERT-V (*i.e.,* BERT-MV) in a late-fusion manner.

The main contributions are summarized as follows:

- We emphasize the importance of numeracy in pre-training LM and propose two numerical pre-training methods: magnitude model and value model.
- We apply the proposed methods to BERT and pre-train two LMs: BERT-M and BERT-V, which are further aggregated to BERT-MV.
- We construct four datasets for the evaluation of numeracy and conduct extensive experiments on the datasets, which validate the effectiveness of our proposal.

2 METHODOLOGY

We first introduce the conventional pre-training method, followed by the two proposed numerical pre-training methods.

2.1 Masked LM Pre-training

Given a tokenized document with N tokens $[Tok_1, \cdots, Tok_N]$, which are typically word-pieces [34], the LM projects each token Tok_n into an latent representation $\boldsymbol{H}_n$. *Masked LM* (MLM) [9] is a widely used objective to optimize the parameters of LM [8, 9, 22, 47]. As illustrated in Figure 1(a), MLM randomly masks tokens in the document according to a given percentage, and aims to predict the vocabulary ID of the masked tokens. In particular, MLM randomly replaces input tokens with a special token [MASK], and feeds the edited document into the LM. In this way, MLM encourages the LM to capture the hidden connections between the masked token and its context tokens [16]. Formally, the objective of MLM is:

$$\Gamma_{MLM} = \min_{\boldsymbol{\Theta}} \sum_{n \in \mathcal{N}} l\left(Tok_n, f_{ID}(\boldsymbol{H}_n)\right), \tag{1}$$

where $f_{ID}(\cdot)$ is a classification function that projects the latent representation of the masked token [MASK] that replaces Tok_n (*i.e.,* output of the LM corresponds to the masked token) to a vocabulary ID; $l(\cdot)$ is a classification loss such as the cross-entropy loss; $\mathcal{N}$ denotes the set of masked tokens in the training corpus; and $\boldsymbol{\Theta}$ denotes all model parameters to be trained which is typically in a deep bidirectional Transformer [9].

2.2 Numerical Pre-training

Our target is to empower the LM with numeracy in a self-supervised manner *w.r.t.* three fundamental numeracy skills : 1) understanding number, 2) number-text composition, and 3) number-number composition. Similar to MLM, our focus is to design training tasks targeting at the skills of numeracy.

Magnitude Model. *Magnitude Model* (MM) randomly masks numbers in the input document, and predicts the magnitude by a factor of 10 of the masked number based on its context. Figure 1(b) shows a toy example of MM where Tok_2 (85) and Tok_3 (06) are word-pieces of a number (*i.e.,* 8506). The two tokens are masked and replaced with a special token [NUM], and MM encourages the model to predict a magnitude of $10e3$. By optimizing the LM to make correct prediction, MM enforces the LM to discover clues from the context, including the other numbers, that indicate the magnitude of the target number. For instance, the LM will capture the connection among *Spend*, *GPU*, and $10e3$ from the toy example in Figure 1. Toward this end, the LM is taught to understand the magnitude and perform number-text association and number-number association. Formally, the objective of MM is:

$$\Gamma_{MM} = \min_{\boldsymbol{\Theta}} \sum_{m \in \mathcal{M}} l\left(Mag_m, f(\boldsymbol{H}_{[NUM]})\right), \tag{2}$$

where $f(\cdot)$ is a classification function that projects the latent representation of the masked number $\boldsymbol{H}_{[NUM]}$ to its magnitude; Mag_m is a label within $\{10e\text{-}10, ..., 10e10\}$. This range is intuitively set according to the distribution of numbers in the corpus for pre-training [8][2]; and $\mathcal{M}$ denotes the set of masked numbers in the training corpus.

Value Model. We further devise a similar training task *Value Model* (VM) that predicts the value of a randomly masked number instead of its magnitude. VM is much harder than MM since the label space of VM (real numbers in a wide range) is much larger than that of MM. Formally, the objective of VM is:

$$\Gamma_{VM} = \min_{\boldsymbol{\Theta}} \sum_{m \in \mathcal{M}} r\left(Val_m, g(\boldsymbol{H}_{[NUM]})\right), \tag{3}$$

where $g(\cdot)$ is a regression function that predicts the number value; $r(\cdot)$ is a regression loss such as mean square error (MSE); and Val_m is the ground truth of number value. Note that it is almost infeasible to predict the exact value of the number from the context. VM is thus mainly to encourage the LM to understand number in a finer grain than the order of magnitude as compared to MM. Besides, VM only masks a token of the number (*e.g.,* Tok_2 in Figure 1(c)) to accelerate the training where the LM can quickly learn a rough range of the target value from the remaining tokens of the number.

Although the standard MLM also has the chance to mask a token within number, the proposed MM and VM are inherently different from MLM due to the number-oriented prediction targets. For instance, for the number 8506 in Figure 1 where the token "85" is masked and "06" is given, VM can distinguish that a prediction of 7506 is better than that of 1506, while MLM treats "75" and "15" similarly as two wrong vocabulary IDs. Note that both MM and VM require the identification of numbers to generate training data, which can be easily achieved by heuristic rules [33, 38].

[2] A large range can be easily achieved if necessary by simply adjusting the classification function and further training our LM over corpus with numbers in a large range.

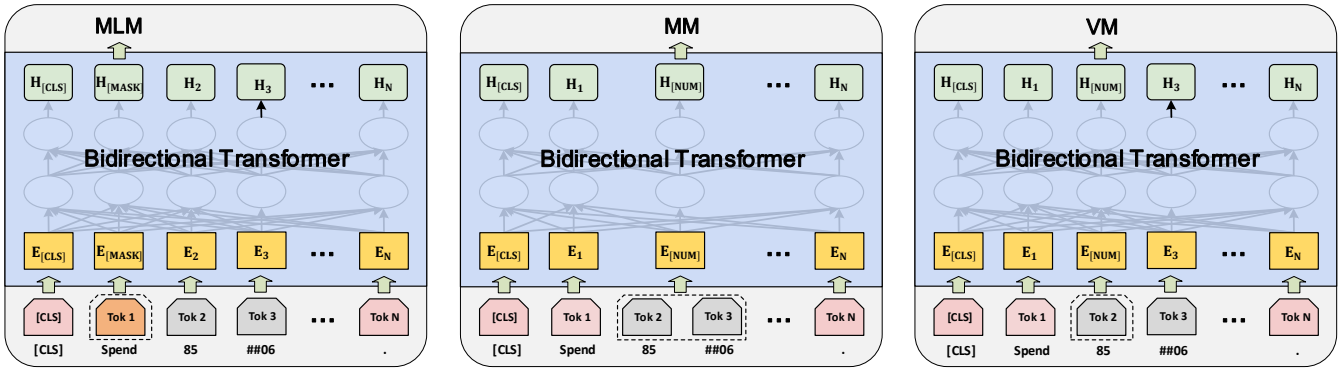


Figure 1: Illustration of different pre-training methods: (a) the existing MLM; (b) magnitude model (MM); (c) value model (VM). Note that tokens in grey color are numeric tokens. Masked tokens are highlighted with dotted frame where $E_{[MASK]}$ and $E_{[NUM]}$ are token embedding. The raw text of the example is "Spend 8506 dollars to buy a Tesla V100 Volta GPU.".

Instantiation. The deep bidirectional Transformer[3] used in BERT has become a dominant architecture of LMs. Considering the computation cost of pre-training a LM from scratch, an energy efficient and feasible way to perform numerical pre-training is to train upon the model parameters released by previous work. For example, by taking BERT as the initiation, we train two new LMs according to Equation 2 and 3, which are named as *BERT-M* and *BERT-V* . Considering that MM and VM are focused on different properties of number, one further improvement is integrating *BERT-M* and *BERT-V*. We aggregate them in a late fusion manner [35] instead of training a new LM with an objective that combines Equation 2 and 3. This is because MM and VM have different masking strategies where MM masks the whole number while VM masks word-pieces. In particular, for a downstream task, we separately fine-tune *BERT-M* and *BERT-V* and calculate the weighted sum of their predictions. The aggregation is named as *BERT-MV*.

3 DATASETS

To evaluate the numeracy of pre-trained LM, we adopt four number-oriented prediction tasks: 1) *Number Forecasting* (NF); 2) *Numerical Decoding* (ND); 3) *Magnitude Inference* (MI) [5]; and 4) *Numerical Fact Checking* (NFC) [5], where better prediction performance can reflect the numeracy of the pre-trained LM from at least one of following perspectives: 1) *Number embedding*, which reflects whether the embedding of numeric tokens in a LM indeed encodes the magnitude and value information of numbers. 2) *Number-text composition*, which shows whether the LM capture the meaningful connections between common words and numbers. 3) *Number-number composition*. It reflects whether the LM is able to capture connections between different numbers. In particular:

Number Forecasting. The task [44] tests the expressiveness of numeric token embedding. It takes the embeddings of number including integer, decimal and fraction as inputs to predict the scientific notation value. We utilize scientific notation value to separate magnitude and value precision, so as to avoid possible negative impacts from different scales. We propose to use MSE as the measurement, where lower MSE implies a higher quality of embeddings, *i.e.,* the numeric values are recovered with higher precision. In particular, a number is typically segmented into multiple tokens by the LM, which typically have 1-3 digits. The embedding of these tokens are fed into a prediction model such as an LSTM.

Numerical Decoding. It is coherent with the MLM pre-training objective [9], which evaluates the quality of contextual numeric token representations. It takes a masked document as input and predicts the masked numeric tokens according to the context information. This task, to some extent, requires the LM to capture the connections between the target numeric token and the context. As a classification task, we use accuracy (Acc) as measurement, where a higher value indicates a stronger composition ability of the contextual numeric token representations.

Magnitude Inference. The MI task aims to predict the magnitude of a number according to its contexts. To some extend, MI is a harder task than ND since we perform number-level masking instead of single tokens so as to encourage the model to fully rely on the context. MI is a M-way classification problem where M is the number of magnitude orders considered. To avoid the potential impacts from class imbalance, we use Micro-F1 and Macro-F1 scores as measurements where a higher value means better performance.

Numerical Fact Checking. This task aims to identify whether the numerical statements in a document hold true or false [5]. Given a document with at least one number, it predicts whether there is a number with exaggerated or understated values. In other words, the task is to check whether the value of number is larger, smaller or equal to the one in reality. We utilize Acc as measurement since the task only has 3 classes and it is easy to achieve class balance.

Regarding the selected tasks, we construct labeled datasets of financial documents. In particular, we collect a corpus of 100,000 financial news on Tiingo[4] from Jun. 2016 to Jun. 2019, which includes both title and abstract. In addition to the raw text, each document is pre-processed with word-piece tokenization according to the vocabulary in BERT [5] . As illustrated in Figure 2(a), each document

[3]We refer the reader to the origin paper [9] for more details of the model architecture.

[4]https://www.tiingo.com/.

[5]https://github.com/google-research/bert.

Table 1: Description of selected evaluation datasets of NF, ND, MI, and NFC.

Task	Type	Evaluation Metric	Description	Dataset (#Documents)		
				Training	Validation	Testing
NF	Regression	MSE	Predict the value of number	17,600	2,200	2,200
ND	Classification	Acc	Predict the ID of masked numeric token	-	-	100,000
MI	Classification	Micro-F1 Macro-F1	Predict the magnitude of masked number	60,000	20,000	20,000
NFC	Classification	Acc	Predict whether a numerical statement is a fact	180,000	60,000	60,000

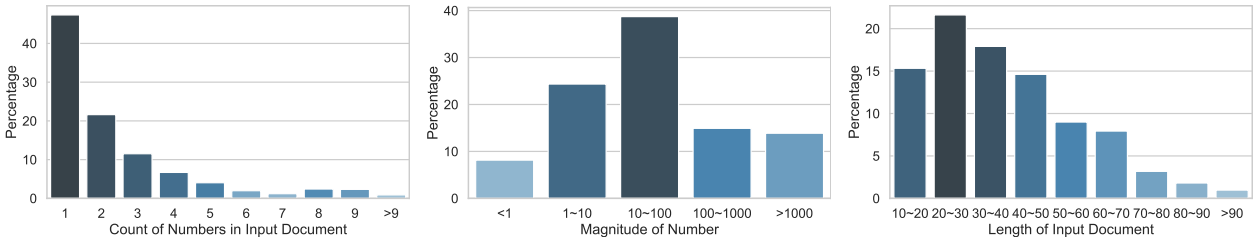


Figure 2: Distribution of the documents *w.r.t.* : (a) the count of numbers; (b) the magnitude of number; and (c) document length.

contains at least one number. Moreover, this corpus contains documents with number intensity at different levels (from 1 number to more than 10 numbers). In addition, as shown in Figure 2(b), most of the numbers have value within the range of [1, 1,000], which can be used to evaluate numeracy to some extend. Lastly, Figure 2(c) shows the distribution of documents *w.r.t.* the total number of tokens (*i.e.,* document length). As can be seen, this corpus is more about short documents since we only consider the title and abstract of news articles. In this way, the document still contains enough information, but will not take too much memory for fine-tuning and testing LMs. Upon this corpus, we first construct labeled datasets for the ND, MI, and NFC tasks.

Numerical Decoding. Following the conventional masked language model [9], we randomly replace 15% of the tokens with [MASK] to test pre-trained LMs. For example, the sentence "*FuelCell Energy Announces Pricing of $40 Million Registered Direct Offering of Common Stock and Warrants*" will be converted to: *[fuel, [MASK], ##l, energy, [MASK], pricing, of, $, [MASK], million, registered, [MASK], offering, of, ##com, ##mon, stock, [MASK], warrant, ##s]*. The LM is expected to predict the ID of masked tokens, where we evaluate the prediction accuracy over the numeric tokens.

Magnitude Inference. In this dataset, we mask a number in each document and label the document with the magnitude of the masked number. For example, the same example sentence will be changed to: *[fuel, ##cel, ##l, energy, announces, pricing, of, $, [MASK], million, registered, direct, offering, of, com, ##mon, stock, and, warrant, ##s]*. The model is expected to correctly predict the magnitude of the masked number ("40"), *i.e.,* the label is $10e1$. As shown in Figure 2(b), the numbers in the corpus is mainly at five orders. We thus confine the MI task to be a 5-way classification.

Numerical Fact Checking. This dataset is close to the one of MI. Instead of masking a number in each document, we construct fake documents by editing the number and keeping the remaining unchanged to simulate numerical frauds. In particular, the original document, and the fake ones whose number is enlarged or reduced by 50% are labeled as *true*, *exaggerate*, and *understate*, respectively. Again, for the example document, we will replace the original number "40" to "60" and "20" to generate the exaggerate and understate document. Note that we intuitively set the scale of perturbation as 50% according to the results in previous work [5][6].

Number Forecasting. The NF task takes number (*e.g.,* "8605") as input document, which is tokenized according to the vocabulary of the LM (*e.g.,* ["86", "05"]). Considering that the number in our corpus is mainly within a narrow range of [1, 1,000], we randomly select numbers in $[10e\text{-}10, 10e10]$ to reflect the generalization ability of the embeddings of the LM. Furthermore, we enforce the numbers to be in various formats including integer, decimal, and fraction. Without loss of generality, we randomly sample 1,000 numbers from each magnitude to restrict the scale of dataset to be affordable. Moreover, we omit the negative numbers since it is trivial to discriminate the sign of number.

4 EXPERIMENTS

We test the proposed numerical pre-training methods on the constructed datasets to investigate their effectiveness and shed light on the performance improvement *w.r.t.* the number intensity of documents.

Pre-training Dataset (fin10-K)[7] is a large-scale corpus in financial domain with SEC fillings of annual 10-K reports from companies listed in U.S. stock markets [8]. In total, fin10-K consists 497 million words with 7.9 million of them are recognized as numbers. We follow the prior work on numeral modeling [38] and number standardization [33] to recognize numbers and parse their value. Note that fin10-K is only used for pre-training and No document in the four testing datasets has appeared in fin10-K.

Compared Methods. We compare the proposed method with four off-the-shelf LMs: BERT [9], ALBERT [20], XLNet [47], and FinBERT [8]. We initialize the proposed BERT-M and BERT-V with

[6]This work does not employ the datasets of MI and NFC in [5] since their datasets are insufficient for numeracy evaluation. For instance, a large portion of the target numbers are years, where the LM can easily recognize the order of magnitude and identify the numerical fraud.

[7]http://people.ischool.berkeley.edu/~khanna/fin10-K/.

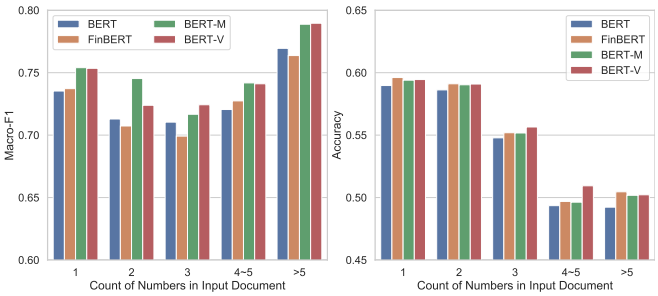
the parameters of FinBERT which is pre-trained over the fin10-K corpus. For BERT-MV, we simply average the prediction of BERT-M and BERT-V on the downstream task. Note that all methods train LM with 12 transformer layers with 12 self-attention heads and hidden size of 768. Note that we omit potential baselines that consider numeracy in the training of conventional LM such as [38] since they cannot be applied to the recent LMs, which have shown superior performance [9].

Implementation Details. We implement the proposed methods based on BERT[8]. (1) **Pre-training.** BERT-M and BERT-V are pre-trained over fin10-K for 1 epoch on 4 Tesla V100 GPUs with batch size of 96 per GPU and the max token length of 128 where we set the learning rate as 1e-6 and omit the warmup. (2) **ND.** For all the LMs, we use the decoding head corresponding to MLM to predict the ID of the masked tokens. Note that both BERT-M and BERT-V inherit the MLM decoding head from their initialization. (3) **MI and NFC.** These two tasks are solved by fine-tuning the pre-trained LM with an additional prediction layer. For MI and NFC, the prediction layers are fed in the output embedding corresponding to the [NUM] (*i.e.,* the masked number) and [CLS] (*i.e.,* the start token), respectively. Note that we use [CLS] token for the NFC since we do not know which number could be exaggerated or understated. For both tasks, we fine-tune the model for 5 epochs with learning rate of 2e-5, batch size of 64, and max token length of 128. (4) **NF.** We lookup the embedding table of the LM and extract a sequence of embeddings corresponding to the numeric tokens of the input number. The embeddings are fed into a prediction model with one Bi-LSTM layer and three fully-connected layers to predict the value and magnitude of the number. We set the size of hidden layers as 20 and train the model for 100 epochs through Adam with batch size of 128 and learning rate of 5e-3.

4.1 Performance Comparison

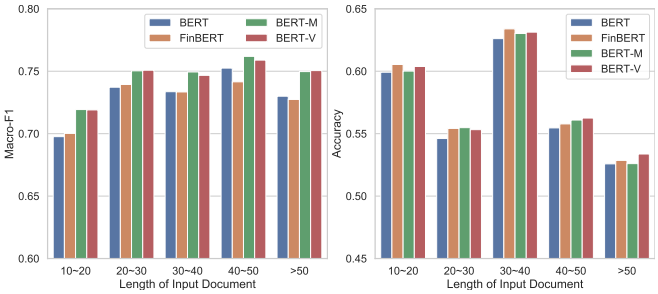
Table 2 summarizes the performance of compared methods on the four datasets. Note that we omit the result of ALBERT and XLNET on the ND task since their token vocabulary are distinct from the other methods. From the table, we have the following observations:

- Across all the four tasks, FinBERT achieves an average improvement over BERT by 8.08%, which is attributed to the pre-training over fin10-K. As fin10-K is more number-intensive than the corpus for pre-training BERT, the result indicates that training with more numbers indeed can enhance the numeracy of LM.
- BERT-M and BERT-V outperform FinBERT with a relative improvement of 4.83% and 4.34%, respectively. As all the three models are trained on the fin10-K corpus, the improvements validate the effectiveness of the proposed pre-training methods. Furthermore, this result justifies the rationality of enhancing the numeracy of LM in a self-supervised manner, *i.e.,* through additional training tasks of predicting number properties.
- In all cases, BERT-MV outperforms both BERT-M and BERT-V, which is attributed to the model aggregation. This improvement is reasonable and consistent with the observations in previous work [22] that model aggregation in a late fusion manner is effective for solving downstream tasks. This result reflects the potential of aggregating more LMs considering different skills of numeracy in the pre-training such as numerical operations [13].
- Among the remaining baselines, BERT outperforms ALBERT, which is a parameter reduction version of BERT. Although previous work shows that the parameter reduction does not hurt the literacy of LM, this result indicates the potential harm of parameter reduction on the numeracy of the LM. We postulate the reason to be the factorized embedding parameterization, which might cause the information loss on numeric tokens. Besides, XLNet performs slightly better than BERT, which is consistent with the observation on standard text understanding tasks [47].
- As to different tasks, BERT-M and BERT-V achieve the largest performance gain over FinBERT on NF (>12.47%). The result shows that the embeddings learned by the proposed MM and VM can better recover the value information of numbers.



(a) Magnitude Inference (b) Numerical Fact Checking

Figure 3: Performance on MI and NFC *w.r.t.* documents with different count of numbers. The count of numbers means how many numbers appear in the document.



(a) Magnitude Inference (b) Numerical Fact Checking

Figure 4: Performance on MI and NFC *w.r.t.* documents with different lengths.

4.2 In-depth Analysis

To investigate where numerical pre-training works better, we perform group-wise analysis regarding the number intensity of the input documents. In particular, we split the testing documents into groups according to the *count of numbers* or *document length*. We select MI and NFC to conduct the study and omit the results on ND for saving space, which have similar trends as the results on the MI task. Intuitively, a number intensive document is an easy case in the MI task (more numerical clues visible), but a hard case in the NFC task (more candidate numerical frauds).

4.2.1 Study w.r.t. *the Count of Numbers.* Figure 3 shows the performance of BERT, FinBERT, BERT-V, and BERT-M (other methods

[8]https://github.com/huggingface/transformers.

Table 2: Prediction performance of the compared methods on the four datasets. RI means the relative improvement achieved by the corresponding method over FinBERT. Values in bold font means the best performance on the dataset. ↑ and ↓ denote that better performance is reflected by a higher and lower value, respectively.

Method	NF	ND	MI		NFC	RI
	MSE↓	Acc↑	Micro-F1↑	Macro-F1↑	Acc↑	
BERT	0.2921	0.1094	0.7292	0.7319	0.5653	-
ALBERT	0.2977	-	0.7139	0.7151	0.5391	-
XLNet	**0.1608**	-	0.7282	0.7326	0.5685	-
FinBERT	0.2574	0.1309	0.7275	0.7301	0.5712	-
BERT-M	0.2155	0.1322	0.7437	0.7473	0.5697	4.83%
BERT-V	0.2253	0.1342	0.7438	0.7468	0.5720	4.34%
BERT-MV	0.1804	**0.1390**	**0.7448**	**0.7477**	**0.5999**	10.88%

are omitted for better illustration) on MI and NFC. From the figures, we have the following observations: 1) For MI, all models achieve the best performance on the more number-intensive group with more than 5 numbers. This result is as expected since numbers in the document could be closely connected to each other. As such, more visible numbers in the document means more clues for predicting the magnitude of the masked number. On the NFC task, all models perform worse when the input document contains more numbers. Recall that the NFC task requires the model to recognize which number is intentionally exaggerated or understated before classifying the document. As such, documents with more numbers is harder to be classified.

4.2.2 Study w.r.t. *Length of Document.* Figure 4 shows the performance of compared methods on MI and NFC.

Magnitude Inference. From Figure 4(a), we can see that: a) All models achieve the worst performance on short documents with less than 20 tokens. This result is reasonable since shorter documents might not have enough clues in the context to infer the magnitude of the masked number. This result points out a shortage of the pre-trained LMs on short documents, suggesting the consideration of additional information in the pre-training such as domain knowledge [39, 50]. b) BERT-M and BERT-V achieve the biggest improvement over FinBERT on long documents with more than 50 tokens where the context information is much richer. The improvement reflects the better composition ability of BERT-M and BERT-V, *i.e.,* making the usage of context information more effective, which is attributed to the magnitude model and the value model. c) In all cases, BERT-M and BERT-V consistently outperform FinBERT, which further validates the effectiveness and rationality of injecting numeracy into LM.

Numerical Fact Checking. In Figure 4(b), the performance across document groups does not show a clear trend. We postulate the reason to be the existence of testing samples that require information beyond the input to check the numerical statement. For instance, in the document of "*China GDP growth in 2019 was 3.1 percent*", it needs the knowledge that China GDP growth in the first three quarters is already much large than 3.1 percent to classify it as understate. It thus would be beneficial to encode such numerical domain knowledge into the pre-training of LM.

4.2.3 Study Beyond Financial Documents. In addition to the constructed datasets, we then evaluate the pre-trained LMs on public benchmark dataset of numerical question answering [10]. The task aims to answer number-involved questions according to a relevant passage, which require discrete operations over the numbers in the passage. In particular, we adopt the widely used dataset DROP[9] and select the questions with numeric answers (51,967 in total). We randomly split the dataset into training, validation, and testing with a ratio of 8:1:1 and report the testing performance *w.r.t.* relative mean absolute error (rMAE) where a lower value indicates better performance. We follow the origin paper of DROP [10] to fine-tune FinBERT and BERT-MV over the training question-answer pairs where a fully-connected layer is adopted to focus the value of answer. Under this setting, FinBERT and BERT-MV achieve rMAE of 0.7184 and 0.6897. The superior performance of BERT-MV further validates the effectiveness of our pre-trained LM, which is attributed to the numerical pre-training.

5 RELATED WORK

LM Pre-training. Following the success of pre-training word embedding in various NLP tasks [25, 28, 29, 42], extensive efforts are dedicated to LM pre-training [8, 9, 30, 47], which focus on two perspectives: model architecture and training objective. Along model architecture, the overall trend is shifting from recurrent neural networks [14, 29, 30] to Transformer [43] due to its superior capability in capturing context semantics and its friendlies to large-scale distributed training. As such, Transformer has become the dominant architecture of LM [9, 31, 47]. As to the training objective, most recent LMs aim to capture bidirectional context. These training objectives include MLM [9, 22], next sentence prediction [9, 22], span prediction [18], cross-lingual pre-training [6], bidirectional autoregressive pre-training [36, 47], and knowledge-aware pre-training [39, 50]. While the existing LMs show strong literacy, enhancing the numeracy of these LMs has received relatively rate scrutiny, which is the focus of this work.

Numeracy of LM. A few existing work has considered the numeracy of LM [13, 17, 38], which is based on the autoregressive LM and largely focused on overcoming the out-of-vocabulary (OOV) issue when learning number representations. However, the OOV

[9] https://allennlp.org/drop.

issue has been largely resolved by the Byte Pair Encoding [34] word segmentation, which is a default setting for the recent advance of pre-trained LM. As such, these methods [38] cannot be applied to enhance the numeracy of recent LMs. This work is focused on the numeracy of recent LMs such as BERT. A very recent work [13] has a similar target, but focuses on the skill of numerical calculations, and is restricted to special inputs with a heterogeneous format of table and text. Moreover, a few work has explored the evaluation of LM numeracy [26, 38, 44]. However, these researches typically focus on one objective of numeracy, *e.g.*, number embedding, lacking a thorough consideration of numeracy. Beyond evaluation, this work also provides numeracy enhanced LMs.

Numerical Text Understanding. Numerical text understanding means applications on number intensive documents. In addition to the variant of common text understanding applications in special domains such as sentiment analysis over financial reports [24] and recommendation of financial news [12], there are also several number-orient tasks, including number classification [4], numeral attachment [3], numeric fused-heads constructions [11], and discrete reasoning [32]. In addition, extensive efforts are dedicated to numerical question answering which requires numerical inference to answer a given question [51] or solve a math word problem [48]. This line of research is focused on the development of neural components to perform numerical inference [1, 15, 32]. For instance, Ran *et al.* [32] proposed to model the number comparison and numerical reasoning by a numerically-aware graph neural network, namely NumNet. Technically speaking, this work is on an orthogonal direction which aims to improve the numeracy of pre-trained LM and is able to be applied to various numerical text understanding applications rather than a specific task.

6 CONCLUSION

In this paper, we highlighted the importance of numeracy for NLP applications in finance. We explored the central theme of enhancing the numeracy of pre-trained LM. We proposed two numerical pre-training methods, *magnitude model* and *value model*, that teach the LM to understand number and learn number-text composition and number-number composition. We applied the proposed methods to pre-train BERT, obtaining *BERT-M* and *BERT-V*, which are further aggregated to be *BERT-MV*. Moreover, we constructed four datasets to evaluate the numeracy of LM, where extensive experiments validate the effectiveness of the proposed methods.

Insights from the experimental results indicate several potential future directions. For instance, we will consider more numerical objectives into LM pre-training such as recognizing the number category [4], considering fine-grained number connections, and incorporating numerical domain knowledge [45]. In addition, we will evaluate the pre-trained LMs on more real-world applications with number intensive texts such as news generation [41], information extraction [27], and dialogue system [49]. Besides, we would like to extend to multiple modalities [37] to account for the figures and tables in financial documents.

REFERENCES

[1] Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving BERT a Calculator: Finding Operations and Arguments with Reading Comprehension. In *EMNLP-IJCNLP*. ACL, 5949–5954.

[2] Margaret E Brooks and Shuang Yueh Pui. 2010. Are individual differences in numeracy unique from general mental ability? A closer look at a common measure of numeracy. *Individual Differences Research* (2010).

[3] Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019. Numeral attachment with auxiliary tasks. In *SIGIR*. ACM, 1161–1164.

[4] Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Final Report of the NTCIR-14 FinNum Task: Challenges and Current Status of Fine-Grained Numeral Understanding in Financial Social Media Data. In *NII Conference on Testbeds and Community for Information Access Research*. Springer, 183–192.

[5] 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*. 6307–6313.

[6] Alexis CONNEAU and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *NeuIPS*. Curran Associates, Inc., 7059–7069.

[7] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 985–988.

[8] Vinicio DeSola, Kevin Hanna, and Pri Nonis. 2019. FinBERT: pre-trained model on SEC filings for financial natural language tasks. *arXiv e-prints* (2019). arXiv:1908.10063

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. ACL, 4171–4186.

[10] 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 *NAACL-HLT*. ACL, 2368–2378.

[11] Yanai Elazar and Yoav Goldberg. 2019. Where's My Head? Definition, Data Set, and Models for Numeric Fused-Head Identification and Resolution. *TACL* 7 (2019), 519–535.

[12] Fuli Feng, Moxin Li, Cheng Luo, Ritchie Ng, and Tat-Seng Chua. 2021. Hybrid learning to rank for financial event ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 233–243.

[13] Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting Numerical Reasoning Skills into Language Models. *ACL* (2020).

[14] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *ACL*. ACL, 328–339.

[15] Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. A Multi-Type Multi-Span Network for Reading Comprehension that Requires Discrete Reasoning. In *EMNLP-IJCNLP*. ACL, 1596–1606.

[16] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language?. In *ACL*. ACL, 3651–3657.

[17] Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, Haofen Wang, and Kewei Tu. 2019. Learning Numeral Embeddings. *arXiv preprint arXiv:2001.00003* (2019).

[18] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *arXiv e-prints* (2019). arXiv:1907.10529

[19] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 39–48.

[20] 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. In *ICLR*.

[21] Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised Question Answering by Cloze Translation. In *ACL*. 4896–4910.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv e-prints* (2019). arXiv:1907.11692

[23] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 49–58.

[24] Xiliu Man, Tong Luo, and Jianwu Lin. 2019. Financial Sentiment Analysis (FSA): A Survey. In *ICPS*. IEEE, 617–622.

[25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NeuIPS*. Curran Associates, Inc., 3111–3119.

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

[27] Guoshun Nan, Zhijiang Guo, Ivan Sekulić, and Wei Lu. 2020. Reasoning with Latent Structure Refinement for Document-Level Relation Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 1546–1557.

[28] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. ACL, 1532–1543.

[29] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *ACL*. ACL, 2227–2237.

[30] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *TACL* (2018).

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv e-prints* (2019). arXiv:1910.10683

[32] Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine Reading Comprehension with Numerical Reasoning. In *EMNLP-IJCNLP*. ACL, 2474–2484.

[33] Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about quantities in natural language. *TACL* 3 (2015), 1–13.

[34] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1715–1725.

[35] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. 2005. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*. 399–402.

[36] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv e-prints* (2019). arXiv:1905.02450

[37] Xue Song, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. 2021. Spatial-temporal Graphs for Cross-modal Text2Video Retrieval. *IEEE Transactions on Multimedia* (2021).

[38] Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers. In *ACL*. ACL, 2104–2115.

[39] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv e-prints* (2019). arXiv:1904.09223

[40] Alon Talmor and Jonathan Berant. 2019. MultiQA: An Empirical Investigation of Generalization and Transfer in Reading Comprehension. In *ACL*. ACL, 4911–4921.

[41] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From Neural Sentence Summarization to Headline Generation: A Coarse-to-Fine Approach.. In *IJCAI*. 4109–4115.

[42] Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *ACL*. ACL, 384–394.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeuIPS*. Curran Associates, Inc., 5998–6008.

[44] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In *EMNLP-IJCNLP*. ACL, 5310–5318.

[45] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. *ICLR* (2020).

[46] Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *NAACL-HLT*. ACL, 2324–2335.

[47] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeuIPS*. Curran Associates, Inc., 5754–5764.

[48] Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2019. The gap of semantic parsing: A survey on automatic math word problem solvers. *TPAMI* (2019).

[49] Yangjun Zhang, Pengjie Ren, and Maarten de Rijke. 2021. A Human-machine Collaborative Framework for Evaluating Malevolence in Dialogues. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 5612–5623.

[50] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*. ACL, 1441–1451.

[51] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 3277–3287.