Probing Numerical Concepts in Financial Text with BERT Models

Shanyue Guo, Le Qiu and Emmanuele Chersoni
Department of Chinese and Bilingual Studies
The Hong Kong Polytechnic University
Yuk Choi Road 11, Kowloon, Hong Kong, China
{shanyue.guo,emmanuele.chersoni}@polyu.edu.hk,lani.qiu@connect.polyu.hk

Abstract

Numbers are notoriously an essential component of financial texts, and their correct understanding is key to automatic system for efficiently extracting and processing information.

In our paper, we analyze the embeddings of different BERT-based models, by testing them on supervised and unsupervised probing tasks for financial numeral understanding and value ordering.

Our results show that LMs with different types of training have complementary strengths, thus suggesting that their embeddings should be combined for more stable performances across tasks and categories.

1 Introduction

The analysis of the linguistic and conceptual knowledge contained in the representations of Transformer architectures (Vaswani et al., 2017) has become of general interest since the introduction of pre-trained language models (LMs) (Radford et al., 2019; Devlin et al., 2019). A common paradigm for testing such knowledge is represented by *probing tasks*: a simple classification model takes as input a representation of a word/sentence from a pre-trained LMs (i.e. an *embedding*) and it is asked to solve a task involving human linguistic knowledge (e.g. subject-verb number agreement, coreference resolution etc.), and a good performance is considered as an indicator that the LM encodes the target knowledge (see Belinkov (2022) for an overview).

Understanding numbers is even more essential for the analysis of financial texts, where they may denote different types of concepts (e.g. amounts, percentages, time periods etc.), each one with its own scale of values. Ideally, a model should be able to estimate the correct magnitude of a numeral for a category, and carry out comparisons between category members. Probing provides important insights about which models contain more information about a specific linguistic distinction, because it analyzes their performance in the most simple and controlled settings (Adi et al., 2017; Conneau et al., 2018; Chersoni et al., 2021), and therefore it can guide the choice on the base models for more sophisticated NLP pipelines and downstream tasks.

In our paper, we analyze the embeddings from three different BERTs with probing tasks for numeracy in the financial domain. Such tasks are meant to test numerical understanding, seen as the capacity of interpreting a numerical expression and assigning it to a specific conceptual category, and the capacity of ordering the values of each category on a scale. The BERT models were selected to assess the effect of different types of pretraining in handling numeracy: is a general numeracy-augmented pretraining sufficient to learn knowledge about numerals in the financial domain? Or the exposure to financial text is necessary for capturing the nuances of the meaning of numerals in this domain?

We show that the models perform similarly in a supervised probing task, where the LM embeddings are used to train a classifier. On the other hand, when tested with unsupervised tasks, more differences emerge: although the embeddings of MWP-BERT show more consistency in identifying numeral categories and ordering numeral values, there is not a single model doing consistently better in all categories and tasks. This might suggest the opportunity of combining different LM representations to achieve more stable performance in financial tasks.[1]

2 Related Work

With the rising popularity of Natural Language Processing and text mining for finance (Loughran and McDonald, 2016), researchers quickly adapted pre-

[1] Code and data available at: https://anonymfile.com/KV10e/code-submission.zip

Proceedings of the Joint Workshop of the 8th Financial Technology and Natural Language Processing (FinNLP) and the 1st Agent AI for Scenario Planning (AgentScen), Jeju, South Korea, August 3, 2024

Category	Count	Percentage	Sample Instance
Monetary	2646	37.34	$CY don't let it close below **14.77**
Temporal	2062	29.1	Alert sent to members at **9:59** AM
Percentage	1060	14.96	past Ecommerce sales up **50%**
Quantity	843	11.9	$MU Interesting that about **15k** shares
Indicator	198	2.79	The close over the **200** dma on heavy volume
Option	158	2.23	$ISRG bought weekly **387.50** puts
Product Number	120	1.69	a partnership announcement combined with sd-**101**

Table 1: Descriptive statistics and sample instances of FinNum-1 (target numeral in **bold**).

trained LMs to the financial domain, mostly starting from general-domain architectures (e.g. BERT, Devlin et al. (2019)) and then carrying out additional training on a corpus from the financial domain (e.g. the FinBERT models, Araci (2019); Yang et al. (2020)). The domain adaptation process led to performance improvements, although most evaluations focused on sentiment analysis and related tasks, and improvements were not always consistent (Peng et al., 2021).

Given the interest of the NLP field in numerical understanding, several works focused on improving the mathematical reasoning capacities of LMs (Geva et al., 2020; Thawani et al., 2021; Chen et al., 2022; Petrak et al., 2023). However, despite recent progress, LMs seem to be still struggling with numerals, especially if rare/unseen in the training data (Wallace et al., 2019; Sharma et al., 2024).

In our study, we are interested in seeing whether i) BERT embeddings of numerals can be used to assign them to the right conceptual category, or superordinate class (Chen et al., 2018); ii) the information they contain can be used to infer their values by putting them on a category-specific scale.

3 Experimental Settings

3.1 Dataset

The experiments are conducted on the FinNum-1 dataset (Chen et al., 2018), which was introduced for a shared task in numerical understanding, and consists of numeral expressions in financial tweets categorized in 7 classes: "Monetary", "Percentage", "Option", "Indicator", "Temporal", "Quantity", and "Product Number". Given a numeral in context, a model has to assign it to the right conceptual class. Descriptive statistics and sample instances from FinNum-1 can be seen in Table 1.

The representation of each numeral in the dataset is extracted from the last layer of a LM, resulting in an embedding representation of size 768.

3.2 Models

We focused on BERT-based models to assess the impact on number representations of specific types of training on the same architecture. Other reasons are that such models are relatively lightweight and faster to run, and that they are *bidirectional*, therefore they represent a better choice than autoregressive models for extracting contextualized representations, which is what our tasks require. Recent literature proved that the fact that autoregressive LMs cannot see future tokens represents a drawback for the representation quality of their embeddings (Springer et al., 2024).

The first model that we use is the standard **BERT** model by Devlin et al. (2019). The second one is a domain-adapted version of BERT, **FinBERT** (Yang et al., 2020), which was initialized from a BERT Base checkpoint and then further pretrained on financial corpora. Finally, we include the **MWP-BERT** model by Liang et al. (2022), which incorporate several numeracy grounded pre-training objectives and has been proved to improve the quality of number representations in several mathematical reasoning tasks. We chose these specific models (all in their Base version) because they exemplify different types of additional training of the same architecture, allowing us to ask to what extent training on financial corpora and numeracy injection improve number representations in financial text.

3.3 Probing Tasks

Numeral understanding. The purpose of the supervised numeral understanding task is to investigate whether the embedding representation of numerical data by LMs exhibits discernible variations across distinct numerical categories. This experiment is conducted by assessing the capability of a simple linear classifier (logistic regression, in our case) to identify the correct category of numerals using the embedding dimensions as input features.

We divide the dataset instances in 7 subsets, one for each category, and in each category subset, we

add negative examples in a 1:1 ratio by randomly sampling instances from the other 6 categories. Subsequently, we employ our probing classifier and use stratified 10-fold cross-validation to obtain the average performance for each category subset. We assess performance in terms of the standard metrics of *accuracy, precision, recall*, and *F1-macro*. [2]

This first task is supervised, as a simple classifier is trained on a linguistic distinction (the semantic category of the number) on the basis of the LM representations (the token embeddings of the number). However, such methodology has been criticised as it involves an external classifier, and thus the relation between the performance and the knowledge already in the embeddings is not clear (Levy et al., 2023). To probe more directly the extent to which number properties exist in the embedding spaces, we adopt two additional unsupervised tasks.

Number outlier detection. Outlier detection (Camacho-Collados and Navigli, 2016) relies on the hypothesis that embeddings of the same classes form coherent clusters in the vector space. In this task, we introduce an *outlier* in clusters of numerals that belong to the same category, and then measure the reciprocal similarities between all the numerals in each cluster. If a LM has an accurate representation of a category, the outlier should be the one with the lowest similarity to the other cluster members.

We generated from FinNum-1 an evaluation dataset for outlier detection with 6966 clusters of 8 instances (7 instances of a class + 1 outlier). For space reasons, the details of the process are provided in Appendix A. The performance is evaluated based on detection *Accuracy* and *Outlier Position Percentage (OPP)*.

For each cluster, cosine similarity is computed between each number embeddings pair. Detection accuracy is determined by sorting the numbers according to the average similarities to other cluster members. If the embedding with the smallest average similarity is the one of the outlier, the model gets a hit, and accuracy is given by the number of hits divided by the number of clusters. OPP is computed instead with the following equation:

$$OPP = \frac{\Sigma_{S \in D} \frac{OP(S)}{|S|}}{|D|} \quad (1)$$

D represents the evaluation subset of a category. *S* represents a cluster in D. *OP(S)* refers to the detected outlier position of this evaluation sample (the index in the sorted ranking, according to average cosine). |*S*| refers to the number of embeddings in each cluster (8 in our case). Given that the cosine similarity is sorted in descending order, the OPP value can be seen as an indicator of how close the outlier is to the bottom of the ranking.

Value ordering. Finally, we want to check how accurate are the LMs in representing the *values* of the numerals within each category, in a relative ordering task. Recently, Grand et al. (2022) proposed a method based on *semantic projections* to interpret the content of word embeddings, by means of identifying vector subspaces that corresponding to human-interpretable semantic scales.

The method works as follows: i) identify words that can represent extreme values of a target on a scale, e.g. for SIZE words like *big, huge, gigantic* on one extreme, and *tiny, small, minuscule* on the other extreme; ii) average the corresponding embeddings at the two extremes to obtain a "prototype" of an extreme value for that scale (the concepts of 'very small' and 'very big'), and then connect the averaged vectors with a line; this line was used to represent the scale of human measurements of SIZE; iii) iii) given a list of words/concepts to be ordered by their SIZE, project their embeddings onto the SIZE line and take the relative ordering of their values. Here, we adopt the same method to map number embeddings onto their values, and test if they can be ordered from the smallest to the biggest one in their respective category.

First, we identify the vectors corresponding to the maximum and minimum numbers within each category and we subtract them to obtain a scale vector of $\vec{value}$. Then, we have to calculate the projection of each remaining number in the category onto $\vec{value}$, defined via the classic scalar projection formula:

$$Proj(\overrightarrow{number}) = \frac{\overrightarrow{number} \cdot \vec{value}}{\|\vec{value}\|} \quad (2)$$

For each number in the dataset, we sort both the numbers and their corresponding representations based on the numerical value (the embeddings of

[2] Differently from Chen et al. (2018), here the task is simplified for the probing classifier: instead of a 7-way classification, we have 7 classification models that work in a one vs. all setting. Since models only have to make binary choices on whether instances belong to a class or not, we can expect similar or higher performance compared to Chen et al. (2018).

Model	Acc	Prec	Rec	F1
BERT	0.92	0.92	0.92	0.92
FinBERT	0.92	0.93	0.92	0.92
MWP-BERT	0.88	0.88	0.88	0.88

Table 2: Metrics for the numeral understanding task (averaged by class and rounded to the second decimal).

duplicated numbers are simply averaged). Subsequently, we compute a scale vector by subtracting the numerical representation of the smallest number from that of the largest number within the category. Finally, we calculate the projection of all the numbers in the category onto the $\vec{value}$ vector.

Performance is assessed using two different metrics: *pairwise ordering accuracy* and *Spearman correlation*. In the former, for a category of n instances, we generate n^2 evaluation pairs by pairing each instance with all the other ones in the same category and compare their numerical values, assigning a hit to a LM for every time it correctly picks the example with higher numerical value in a pair; in the latter, we measure the Spearman ρ between the order of the actual values of the numbers in the gold standard and the values obtained via projection of their embeddings.

4 Results

The scores for the numeral understanding task can be seen in Table 2. At a glance, it can be seen that all the models achieve a very high performance, with only MWP-BERT being slightly below 90% for all the evaluation metrics. All the model representations clearly contain relevant information for the identification of the right semantic class of a numeral expression, to the point that a performance around 90% can be obtained even with a linear classifier. More detailed, by-class figures can be found in Appendix B: unsurprisingly, Percentage is the easiest class for all models, probably because the presence in almost all contexts of the percentage sign provides a strong identification cue. [3]

Moving to the unsupervised tasks, we can observe in Table 3 that BERT Base is the best one for Accuracy in identifying the outlier, and MWP-BERT has an edge for the OPP metric. MWP-BERT does in general a better job in clustering a higher number of instances, as it gets the top scores for Monetary, Temporal and Percentage, the most frequent categories (they combine for more than 81% of the data points).

Category	BERT Base	FinBERT	MWP-BERT
Indicator	**0.68/0.92**	0.63/0.90	0.56/0.89
Monetary	0.38/0.75	0.39/0.77	**0.40/0.78**
Option	**0.61/0.87**	0.49/0.84	0.57/**0.87**
Percentage	0.63/0.88	0.51/0.85	**0.67/0.92**
Product Number	0.37/0.76	**0.43/0.79**	0.20/0.74
Quantity	**0.34**/0.71	0.31/**0.72**	0.26/0.69
Temporal	0.24/0.68	0.20/0.62	**0.35/0.77**
AVERAGE	**0.46**/0.79	0.42/0.78	0.43/**0.81**

Table 3: Outlier detection results for the metrics of Accuracy/OPP (best scores in **bold**).

Category	BERT Base	FinBERT	MWP-BERT
Indicator	0.60/0.43	**0.73/0.61**	0.68/0.55
Monetary	0.40/0.60	**0.46/0.73**	0.41/0.28
Option	**0.61**/0.46	0.53/0.47	0.49/**0.48**
Percentage	0.37/0.34	0.42/0.21	**0.46/0.58**
Product Number	0.69/0.37	**0.75**/0.36	0.65/**0.50**
Quantity	**0.71**/0.38	0.63/0.38	0.59/**0.51**
Temporal	**0.58**/0.35	0.55/0.31	0.50/**0.52**
AVERAGE	0.57/0.42	**0.58**/0.44	0.54/**0.49**

Table 4: Pairwise accuracy/Spearman scores on the value ordering task (best scores in **bold**).

Table 4, showing the scores of Pairwise ordering accuracy, displays almost a tie across categories between the LMs: MWP-BERT is the best model for Percentage; FinBERT does better in the most finance-specific categories (Indicator and Monetary) and in Product Number; the Base model is best for the remaining ones. All models display moderate correlations with the actual number values, with MWP-BERT being significantly better than both BERT and FinBERT ($p < 0.01$ for a two-tailed test with the Fisher r-to-z transformation). This suggests that the numeracy-augmented model is the best choice for handling value ordering. However, it also has a sharp drop on Monetary, the most "finance-specific" and frequent category.

In general, a trend of unsupervised tasks is that representations of different models do well in different categories, which suggests that combining them might lead to a more stable performance. To quickly test this hypothesis, we tried to repeat all the above experiments by combining the FinBERT and the MWP-BERT embeddings, using the simple methods of averaging and concatenation. While

[3] Reviewer 3 requests us to report the results of the original FinNum-1 shared task for comparison. The highest F1-Macro that was reported in the FinNum-1 shared task was around 0.90, achieved by Fortia-1 with a convolutional neural network combining different types of word embeddings (word level, character level, ELMo etc.) (Azzi and Bouamor, 2019). However, given that they were operating in an actual multiclass classification setting while we adopted a one vs. all approach, we do not think the scores are directly comparable.

averaging led to inconsistent results, we observed a slight increase of performance for embeddings concatenation in numeral understanding, with the F1-score going up to 0.93, and outlier detection, with 0.46 of Accuracy and 0.82 of OPP. The results are more ambivalent for value ordering: pairwise ordering accuracy goes down to 0.49, but the combined vectors achieve the highest Spearman correlation with the gold standard value with 0.54. This also includes a much higher correlation with the Monetary category, going up to 0.43 from 0.28.

We think this is good preliminary evidence of possible improvements by combining the information in the two vector types, and it is likely that larger improvements could be achieved by adding trainable layers on the top of the original embeddings representations.

5 Conclusion

In this work, we used to simple probing task to analyze the knowledge of financial numerals encoded in different types of BERT-based LMs, particularly in relation with the categories of the numerals in financial text and with the capacity of ordering their values on the scale proper of each category.

While with a supervised probe the numeracy-augmented MWP-BERT does worse, in unsupervised tasks it the representation quality across categories looks generally better. The fact that some models experience highs and lows in some categories might be related to limited exposure in the pretraining phase, which suggest that, in financial tasks, it might be wise to combine different types of embeddings to stabilize the representation and simultaneously account for different aspects of numerical knowledge.

Acknowledgements

The authors acknowledge the support from the project"Analyzing the semantics of Transformers representations for financial natural language processing"(ZVYU), sponsored by the Faculty of Humanities of the Hong Kong Polytechnic University.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *Proceedings of ICLR*.

Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *arXiv preprint arXiv:1908.10063*.

Abderrahim Ait Azzi and Houda Bouamor. 2019. Fortia1@ the NTCIR-14 FinNum Task: Enriched Sequence Labeling for Numeral Classification. In *Proceedings of the NTCIR Conference on Evaluation of Information Access Technologies*.

Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219.

José Camacho-Collados and Roberto Navigli. 2016. Find the Word that Does Not Belong: A Framework for an Intrinsic Evaluation of Word Vector Representations. In *Proceedings of the ACL Workshop on Evaluating Vector-Space Representations for NLP*.

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. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *arXiv preprint arXiv:2211.12588*.

Emmanuele Chersoni, Enrico Santus, Chu-Ren Huang, and Alessandro Lenci. 2021. Decoding Word Embeddings with Brain-based Semantic Features. *Computational Linguistics*, 47(3):663–698.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What You Can Cram into a Single Vector: Probing Sentence Embeddings for Linguistic Properties. In *Proceedings of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting Numerical Reasoning Skills into Language Models. In *Proceedings of ACL*.

Gabriel Grand, Idan Asher Blank, Francisco Pereira, and Evelina Fedorenko. 2022. Semantic Projection Recovers Rich Human Knowledge of Multiple Object Features from Word Embeddings. *Nature Human Behaviour*, 6(7):975–987.

Tal Levy, Omer Goldman, and Reut Tsarfaty. 2023. Is Probing All You Need? Indicator Tasks as an Alternative to Probing Embedding Spaces. In *Findings of EMNLP*.

Zhenwen Liang, Jipeng Zhang, Lei Wang, Wei Qin, Yunshi Lan, Jie Shao, and Xiangliang Zhang. 2022. MWP-BERT: Numeracy-augmented Pre-training for Math Word Problem Solving. In *Findings of NAACL*.

Tim Loughran and Bill McDonald. 2016. Textual Analysis in Accounting and Finance: A Survey. *Journal of Accounting Research*, 54(4):1187–1230.

Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, and Chu-Ren Huang. 2021. Is Domain Adaptation Worth your Investment? Comparing BERT and FinBERT on Financial Tasks. In *Proceedings of the EMNLP Workshop on Economics and Natural Language Processing*.

Dominic Petrak, Nafise Sadat Moosavi, and Iryna Gurevych. 2023. Arithmetic-Based Pretraining Improving Numeracy of Pretrained Language Models. In *Proceedings of *SEM*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language Models Are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8):9.

Mandar Sharma, Rutuja Murlidhar Taware, Pravesh Koirala, Nikhil Muralidhar, and Naren Ramakrishnan. 2024. Laying Anchors: Semantically Priming Numerals in Language Modeling. In *Findings of NAACL*.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition Improves Language Model Embeddings. *arXiv preprint arXiv:2402.15449*.

Avijit Thawani, Jay Pujara, and Filip Ilievski. 2021. Numeracy Enhances the Literacy of Language Models. In *Proceedings of EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In *Proceedings of EMNLP*.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. FinBERT: A Pretrained Language Model for Financial Communications. *arXiv preprint arXiv:2006.08097*.

A Outlier Detection: Dataset Construction

To construct an evaluation dataset for outlier detection, we created category clusters by following the steps in Camacho-Collados and Navigli (2016). We grouped 7 FinNum-1 instances from one class and then we randomly introduced one instance from another category as an outlier. To maximize the utilization of data, we employed a sliding window of size 7 through the list of the instances of each one of the 7 categories to create the clusters.

Following the segmentation of each category subset, we appended 6 data instances from different categories to each generated cluster (one outlier is sampled from each one of the other categories). Through this process, for every set of 7 data samples from each category, 6 distinct datasets containing outliers from different classes were generated. In total, the number of clusters is 6966.

B Full Scores for the Numeral Understanding Task

The specific scores for each system, broken down by class, can be seen in Tables 5, 6 and 7.

BERT

Category	Acc	Prec	Rec	F1
Indicator	0.91	0.92	0.91	0.91
Monetary	0.89	0.89	0.89	0.89
Option	0.92	0.92	0.92	0.92
Percentage	**0.98**	**0.98**	**0.98**	**0.98**
Product Number	0.91	0.91	0.91	0.91
Quantity	0.89	0.89	0.89	0.89
Temporal	0.92	0.92	0.92	0.92
AVERAGE	0.92	0.92	0.92	0.92

Table 5: Probing classifier results with BERT Base.

FinBERT

Category	Acc	Prec	Rec	F1
Indicator	0.93	0.93	0.93	0.93
Monetary	0.92	0.92	0.92	0.92
Option	0.87	0.88	0.87	0.87
Percentage	**0.98**	**0.98**	**0.98**	**0.98**
Product Number	0.93	0.93	0.93	0.93
Quantity	0.90	0.90	0.90	0.90
Temporal	0.94	0.94	0.94	0.94
AVERAGE	0.92	0.93	0.92	0.92

Table 6: Probing classifier results with FinBERT.

MWP-BERT

Category	Acc	Prec	Rec	F1
Indicator	0.91	0.91	0.91	0.91
Monetary	0.86	0.86	0.86	0.86
Option	0.89	0.89	0.89	0.88
Percentage	**0.98**	**0.98**	**0.98**	**0.98**
Product Number	0.81	0.82	0.81	0.81
Quantity	0.83	0.83	0.83	0.83
Temporal	0.89	0.89	0.89	0.89
AVERAGE	0.88	0.88	0.88	0.88

Table 7: Probing classifier results with MWP-BERT.