Revealing the Numeracy Gap: An Empirical Investigation of Text Embedding Models

Ningyuan Deng, Hanyu Duan, Yixuan Tang, Yi Yang
Department of Information Systems, Business Statistics and Operations Management, HKUST
ningyuandeng@ust.hk, hduanac@connect.ust.hk
ytangch@connect.ust.hk, imyiyang@ust.hk

Abstract

Text embedding models are widely used in natural language processing applications. However, their capability is often benchmarked on tasks that do not require understanding nuanced numerical information in text. As a result, it remains unclear whether current embedding models can precisely encode numerical content, such as numbers, into embeddings. This question is critical because embedding models are increasingly applied in domains where numbers matter, such as finance and healthcare. For example, "Company X's market share grew by 2%" should be interpreted very differently from "Company X's market share grew by 20%," even though both indicate growth in market share. This study aims to examine whether text embedding models can capture such nuances. Using synthetic data in a financial context, we evaluate 13 widely used text embedding models and find that they generally struggle to capture numerical details accurately. Our further analyses provide deeper insights into embedding numeracy, informing future research to strengthen the embedding model-based NLP systems with improved capacity for handling numerical content.

1 Introduction

Text embedding models are vital for modern NLP (Zhang et al., 2025a), which power semantic search (Muennighoff, 2022), enabling retrieval-augmented generation (RAG) (Fan et al., 2024). Recent models like E5-Mistral-7B (Wang et al., 2023) haved achieving top benchmark, including MTEB (Muennighoff et al., 2023) and BEIR (Thakur et al., 2021). However, current benchmarks often overlook evaluating embedding models' capability to handle nuanced numerical content in text. There is a clear need to assess how well embedding models capture numerical details, especially in areas like finance, healthcare, and scientific research, where numerical precision matters.

For example, clinical notes with nearly identical wording, like "blood pressure is 120/80 mmHg" and "blood pressure is 180/110 mmHg", indicate vastly different risks. If embeddings fail to preserve these numerical values, they could mislead critical clinical decisions, such as triggering alerts for dangerous readings.

This work aims to fill this evaluation gap. To support the evaluation, we introduce `EmbedNum-1K`, a financial domain-specific dataset designed to test whether numerical information in text is sufficiently preserved in embeddings. The dataset contains 1,000 synthetic samples, each comprising a question ($\mathbf{Q}$) and two candidate answers ($\mathbf{A}^+$ and $\mathbf{A}^-$). For example, a sample may look like:

- **Q:** "Who owns over 15% of the company?"
- $\mathbf{A}^+$**:** "Investor Alice owns a 20% stake."
- $\mathbf{A}^-$**:** "Investor Alice owns a 5% stake."

Given that pair, an ideal embedding model should encode the question ($\mathbf{Q}$) closer to the correct answer ($\mathbf{A}^+$) than to the incorrect one ($\mathbf{A}^-$) in the embedding space. This task is like NLP retrieval but unique: correctness depends on numerical accuracy, not just meaning. Thus, Higher accuracy[1] indicates the model better preserves critical numerical details.

Using `EmbedNum-1K`, we experiment with 13 widely used text embedding models, including BERT-like ones, LLM-based ones, and general purpose and financial domain specific ones, open-source or via commercial API calls. To enable fine-grained evaluation, we vary the numeric format of numbers. For example, numbers can appear in different formats, such as integers (e.g., 6), decimals (e.g., 0.6), percentages (e.g., 6%), and written numbers (e.g., "six").

[1]Accuracy on this task is calculated as the proportion of samples where the model correctly ranks $\mathbf{A}^+$ closer to $\mathbf{Q}$ than $\mathbf{A}^-$.

Findings of the Association for Computational Linguistics: EACL 2026, pages 5448–5461

We find embedding models perform only marginally better than random on numerical tasks. LLM-based models outperform encoder-based ones, and numerical format (e.g., "8%" vs. "0.08") impacts performance. Crucially, improved textual literacy does not guarantee better numeracy, and models struggle with out-of-vocabulary (OOV) and high-precision numbers.

Our subsequent analyses probe these numerical capabilities further, investigating different reasoning types, digit vs. written forms ("24" vs. "twenty-four"), and the role of linguistic context. We also explore whether frequent number exposure in training improves performance and if probing tests predict downstream task success, providing essential insights for building more numerically competent models.

This work makes three main contributions.

First, we introduce `EmbedNum-1K`, a dataset that fills an important evaluation gap by testing how well current embedding models preserve subtle numerical differences in embeddings, which are often overlooked in existing benchmarks. For example, current benchmarks may treat the embeddings of "Investor Alice owns a 20% stake." and "Investor Alice owns a 5% stake." as equally good matches to the embedding of "Who owns over 15% of the company?". In contrast, our evaluation explicitly differentiates such cases, testing whether embeddings capture the precise numerical relationships implied in the text.

Second, we reveal a key limitation of current embedding models: they often fail to accurately preserve subtle numerical information in their embeddings. This suggests that simply scaling up model size or training data, as is common in current embedding model training practices, is not sufficient. Targeted designs that explicitly account for fine-grained information, such as the numerical details studied here, are also necessary.

Third, our experiments generate key insights that may help to develop numeracy-aware text embedding models in the future. We believe this has important implications for advancing number-intensive NLP applications, such as RAG-based systems in finance and healthcare (Wong et al., 2025; Yepes et al., 2024), where accurate numerical understanding is critical. We will make the datasets and experimental code publicly available to ensure reproducibility.

2 Related Work

2.1 Benchmarking Embedding Models and the Evaluation Tasks

Numerous benchmarks have been introduced to evaluate embedding models. A well-known example is the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023). Others include MMTEB (Enevoldsen et al., 2025), which expands the original MTEB to cover more languages, and FinMTEB (Tang and Yang, 2025), which adapts MTEB for domain-specific settings such as finance. Generally, models more capable of encoding similar texts into similar embeddings tend to achieve higher scores on these tasks.

Although current benchmarks deliver various evaluation tasks, it's unclear how well embedding models handle numerical understanding. For instance, clustering papers on arXiv[2] is largely driven by overall topic and keywords rather than numerical details like years or statistics. Even in FinMTEB (Tang and Yang, 2025), tailored to the finance domain, the evaluation of embedding models' ability to deal with nuanced numerical content is not explicitly considered. Our work contributes to this line of research by emphasizing numerical evaluation and creating a dataset to assess how well embedding models capture, preserve, and interpret numerical details in text.

2.2 Numeracy in Embeddings

Numeracy involves understanding and working with numbers. NLP research on embedding numeracy can be traced back to 2019 (Naik et al., 2019) and (Wallace et al., 2019), focusing initially on word embeddings like GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013). Later studies produce number embeddings with improved numeracy preservation (Sundararaman et al., 2020; Duan et al., 2021; Jiang et al., 2020; Sivakumar and Moosavi, 2025). Our work falls within this line of research examining numeracy in embeddings and extends it by focusing on large-scale modern text embedding models, such as the Qwen and Mistral embedding series (Zhang et al., 2025b; Li et al., 2023; Kim et al., 2024; Meng et al., 2024), going beyond traditional word embeddings. This extension is meaningful in the following two key aspects.

First, numbers' interpretation in text varies greatly with the surrounding context. For in-

[2] https://arxiv.org/

stance, investors may respond very differently to "Stock A *rose* by 2%" versus "Stock A *fell* by 2%," even though the change is equal. Unlike word embeddings, which often treat numbers context-free, text embeddings allow numeracy analysis within context by encoding full sentences or paragraphs. Since real-world NLP rarely deals with isolated numbers, our context-based approach offers a more realistic evaluation. Second, compared to traditional word embedding models, text embedding models (often powered by LLMs) serve more prominently as core components in modern NLP systems, especially in RAG settings (Fan et al., 2024). Therefore, we argue that investigating numeracy in text embedding models is of greater practical relevance.

3 Dataset Construction

We describe the curation of `EmbedNum-1K` in the following four steps; the overall workflow is illustrated in Figure **??**.

Step 1: Seed Data Construction. We construct seed data by filtering the English portion of the BULL dataset.[3] This dataset consists of 4,966 natural language question-SQL pairs collected from real-world financial industry settings (Zhang et al., 2024). A question-SQL pair may look like: **Question:** "Which stocks have a P/E ratio above 200?" **SQL:** "select chinameabbr from qt_monthdata where pettm > 200;".

Here, we are interested only in the **Question** entries, specifically those whose **SQL** query counterparts meet the following three criteria: (1) Query includes either the ">" or "<" operator, (2) Excludes time-related fields,[4] and (3) Contains exactly one numeric value. This yields 545 questions.

Step 2: Question Augmentation. To expand the seed set, we generate paired variants by flipping numerical comparisons while preserving context. For example, the question "Which stocks have a P/E ratio *above* 200?" could be augmented with "Which stocks have a P/E ratio *below* 200?". We approach this by prompting the DeepSeek-V3.1 model (DeepSeek-AI, 2024)[5] in non-thinking mode. From the augmented data pool, we randomly select 500 sentences to create 1,000 questions, while ensuring balanced the "above" and "below" cases.

[3] `https://bull-text-to-sql-benchmark.github.io`

[4] Time (date) comparisons are excluded due to their higher complexity.

[5] `https://api-docs.deepseek.com`

Step 3: Answer Generation. For each of the 1K questions ($\mathbf{Q}$), we generate two candidate answers (using DeepSeek-V3.1 (DeepSeek-AI, 2024)): $\mathbf{A}^+$, which satisfies the condition stated in the question $\mathbf{Q}$, and $\mathbf{A}^-$, which differs from $\mathbf{A}^+$ only in its numeric value and does not satisfy the condition. For example, take $\mathbf{Q}$: "Which stocks have a P/E ratio above 200?"; the candidate answers could be:

- $\mathbf{A}^+$**:** "Stock ABC has a P/E of *220*."
- $\mathbf{A}^-$**:** "Stock ABC has a P/E of *180*."

This created the original version of `EmbedNum-1K`, comprising 1K such <$\mathbf{Q}$, $\mathbf{A}^+$, $\mathbf{A}^-$> triples, where all the numeric values involved are integers ranging from 1 to 999.

Step 4: Numeric Format Variation.

Following (Sivakumar and Moosavi, 2023), we create 17 variants of `EmbedNum-1K` by altering numeric formats (e.g., decimals, percentages, and other formats). A detailed description of all variants is provided in Table 1.

Format Category	Description	Example
Orig	Integer ranging from 1 to 999	235
(a) Integers	(a1) Single-digit integer	8
	(a2) Two-digit integer	77
	(a3) Three-digit integer	719
	(a4) Four-digit integer	3035
(b) Decimals	(b1) One decimal place	8.3
	(b2) Two decimal places	8.67
	(b3) Three decimal places	8.868
	(b4) Four decimal places	8.7713
(c) Scaled Down	(c1) 1-digit scaled down by 10	0.8
	(c2) 1-digit scaled down by 100	0.08
	(c3) 1-digit scaled down by 1000	0.008
(d) Scaled Up	(d1) 1-digit scaled up by 10	80
	(d2) 1-digit scaled up by 100	800
	(d3) 1-digit scaled up by 1000	8000
(e) Others	(e1) Written with "percentage"	8 percentage
	(e2) Written with % symbol	8%
	(e3) Comma-separated integer	3,035

Table 1: Descriptions of numeric formats used in the experiments with examples.

This yields 18 datasets in total, including the original, with numbers represented in diverse formats. Prompts for the data curation process are detailed in Appendix A.

4 Experiments

4.1 Experimental Setup

Task Description. Based on the `EmbedNum-1K` dataset, we formulate a retrieval task. Given an instance <$\mathbf{Q}$, $\mathbf{A}^+$, $\mathbf{A}^-$> from the dataset and a chosen embedding model, the aim of the task is to retrieve the correct answer $\mathbf{A}^+$ from the two candidate answers based on the question $\mathbf{Q}$. First, we encode the question and both candidate answers

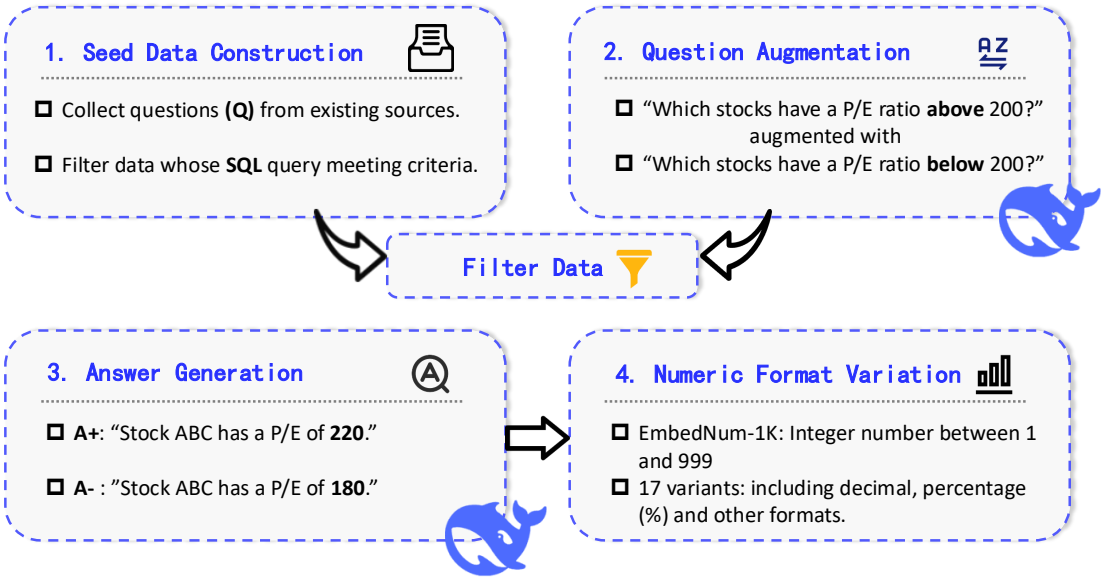

Figure 1: Workflow of the EmbedNum-1K dataset construction process.

into dense embedding vectors using the model. We then compute the cosine similarity between the question embedding and each answer embedding. The answer with the higher similarity score is selected as the retrieved result.

Evaluation Metric. We measure retrieval task performance using Accuracy. Here, Accuracy is defined as the proportion of instances in which the model under evaluation correctly retrieves $\mathbf{A}^+$ over $\mathbf{A}^-$. Higher accuracy reflects more faithful encoding of numerical information, as correct retrieval depends entirely on numeric values in the text.

Embedding Models. We evaluate a broad range of text embedding models. This includes transformer encoder-based models, such as RoBERTa (Liu et al., 2019), finance-embedding[6], SimCSE-BERT-unsup (Gao et al., 2021), MiniLM-L6[7], and MPNet (Song et al., 2020), as well as LLM-based models, including gte-Qwen2-7B-instruct (Li et al., 2023), Linq-Embed-Mistral (Kim et al., 2024), Qwen3-Embedding-8B (Zhang et al., 2025b), SFR-Embedding-Mistral (Meng et al., 2024), e5-mistral-7b-instruct (Wang et al., 2023), and NV-Embed-v2 (Lee et al., 2024). Additionally, we consider two proprietary embedding models, Fin-E5 (Tang and Yang, 2025) and text-embed-3-small (OpenAI, 2025).

[6] https://huggingface.co/FinLang/finance-embeddings-investopedia

[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

4.2 Main Results

We report the performance of all evaluated embedding models on the EmbedNum-1K retrieval task in Table 2. Our main findings are summarized below.

Overall, embedding models struggle to capture numerical details in text, with an average accuracy of merely 0.54 across 13 models, slightly above random guessing (0.5)

This is mainly because current embedding model training practices typically focus on capturing semantic similarities or differences at the sentence or paragraph level, neglecting fine-grained details such as numbers. This aligns with (Liu et al., 2024), who find that embedding models generally fall short in distinguishing nuanced differences between texts. We recommend treating numbers separately from words and developing specialized numerical handling to improve embeddings.

LLM-based embedding models show a clear advantage over encoder-based models. We observe that LLM-based models achieve, on average, about 5 percentage points higher accuracy than encoder-based ones (0.56 vs. 0.51).

We speculate that this advantage is largely due to the pre-existing superior natural language understanding capability of an LLM prior to its adaptation for embedding purposes. However, through what mechanisms this superiority meaningfully

Model Name	Orig	(a) Integers				(b) Decimals			
		(a1)	(a2)	(a3)	(a4)	(b1)	(b2)	(b3)	(b4)
Encoder-Based Models									
RoBERTa	50.8	50.2	50.7	49.5	49.7	48.9	51.2	49.9	50.6
finance-embedding	53.4	53.4	51.2	51.7	50.4	51.4	50.1	51.2	49.9
SimCSE-BERT-unsup	51.3	51.1	51.9	51.0	50.4	51.0	50.7	50.5	50.2
MiniLM-L6	50.6	50.2	51.1	49.0	50.4	51.0	49.8	50.3	50.1
MPNet	51.5	53.0	49.9	50.6	50.3	51.7	51.9	52.9	49.7
LLM-Based Models									
gte-Qwen2-7B-instruct	51.4	58.1	55.2	50.8	51.3	50.0	51.1	52.9	51.8
Linq-Embed-Mistral	51.4	58.0	52.8	50.0	48.4	50.7	50.5	50.4	49.0
Qwen3-Embedding-8B	**53.7**	**62.5**	**55.1**	**53.9**	**52.3**	51.5	52.9	50.9	52.2
SFR-Embedding-Mistral	52.1	59.0	52.4	50.1	49.2	51.8	50.6	51.4	50.9
e5-mistral-7b-instruct	51.5	59.8	51.3	50.2	49.2	52.5	51.3	49.7	50.6
NV-Embed-v2	52.1	62.0	**55.1**	50.8	49.1	53.5	52.2	51.2	51.2
Commercial Models									
Fin-E5	51.7	58.6	52.2	48.9	48.4	51.1	48.7	49.1	49.9
text-embedding-3-small	52.7	50.8	54.0	51.1	50.6	**54.2**	**53.8**	**54.5**	**54.9**
Average performance	51.9	55.9	52.5	50.6	50.0	51.5	51.1	51.1	50.8

Model Name	(c) Scaled Down			(d) Scaled Up			(e) Others		
	(c1)	(c2)	(c3)	(d1)	(d2)	(d3)	(e1)	(e2)	(e3)
Encoder-Based Models									
RoBERTa	50.4	50.2	51.6	52.4	50.8	47.3	52.5	51.9	50.7
finance-embedding	54.0	49.9	53.9	52.7	53.6	51.1	54.2	54.8	51.0
SimCSE-BERT-unsup	52.1	53.3	52.2	50.7	52.7	51.0	51.8	52.4	50.9
MiniLM-L6	51.4	50.1	49.2	52.5	50.4	51.1	50.7	51.4	50.7
MPNet	52.5	52.6	55.2	50.6	51.4	53.2	53.7	53.8	51.9
LLM-Based Models									
gte-Qwen2-7B-instruct	65.3	63.9	57.8	58.3	51.0	52.9	61.9	57.4	51.3
Linq-Embed-Mistral	67.3	58.2	54.8	55.3	49.8	46.9	64.4	64.9	49.7
Qwen3-Embedding-8B	69.2	67.9	64.7	61.0	**55.8**	**54.7**	**71.0**	**69.9**	**53.3**
SFR-Embedding-Mistral	67.9	61.5	57.6	53.1	49.9	48.6	65.5	64.7	48.6
e5-mistral-7b-instruct	66.5	61.2	57.4	53.5	50.8	49.3	67.2	64.8	49.3
NV-Embed-v2	**74.2**	**70.6**	**65.1**	**61.9**	54.7	51.7	69.3	66.6	49.7
Commercial Models									
Fin-E5	62.8	58.4	54.2	54.0	52.2	48.6	66.6	67.9	49.3
text-embed-3-small	67.6	59.6	58.0	57.9	51.3	50.2	59.2	55.0	50.5
Average performance	61.6	58.3	56.3	54.9	51.9	50.5	60.6	59.7	50.5

Table 2: Retrieval accuracy (%) on the original `EmbedNum-1K` data and its 17 variants across 13 evaluated text embedding models. For each numeric format, the highest accuracy is highlighted in **bold**, and the highest accuracy within each model category is underlined. A blue-white-red color gradient is overlaid on the "Average performance" row, with warmer colors indicating higher accuracy and cooler colors indicating lower accuracy across numeric formats.

translates into tangible gains in embedding quality is still an open question and certainly merits further investigation.

Embedding models interpret 8% and 0.08 differently. Our experiments reveal that embedding models are quite sensitive to numeric formats, with accuracy varying by up to 12%. They perform best on decimals (0.x format, 62% accuracy) but struggle most with 4-digit integers (near random guessing).

For this reason, we strongly recommend that future efforts in developing new number modeling strategies carefully account for numeric formats, given their critical role in how models interpret numbers as part of text.

Improved literacy does not necessarily translate into tangible gains in numeracy. Fin-E5, a finance-tuned version of e5-mistral-7b-instruct, shows no clear performance advantage over its base model on the finance-specific `EmbedNum-1K` task.

Complementary to the observation by (Thawani et al., 2021) that better numeracy leads to better literacy, our finding underscores the unique difficulty of numeracy compared to literacy and again highlights the need for specialized designs directly targeting numeracy..

OOV (out-of-vocabulary) numbers challenge embedding models. Models struggle with decimals, large integers, and comma-separated numbers, which are more likely to split into smaller tokens during tokenization (e.g., "1,100" "1", ",", "100").

This could be problematic because splitting a number can disrupt key information, such as its magnitude, which is essential for many downstream tasks. Thus, addressing OOV issues has great potential to improve the usefulness of current embedding models in numbers-intensive real-world applications.

Number precision affects model performance, reflecting human cognition. Models perform better on numbers like 0.8 and 0.0006 than , and our analysis reveals a strong inverse correlation between accuracy and the number of significant figures: as significant digits increase, model performance declines (Figure 2).

This finding suggests that embedding models, like humans, face greater cognitive load and errors with long, high-precision numbers, offering an intriguing parallel for research in language model-brain alignment (Yu et al., 2024).

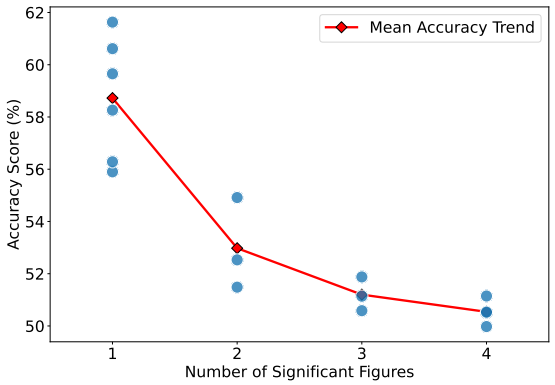

Figure 2: Each blue dot represents the mean performance across the 13 embedding models for a specific numeric format. Each numeric format is categorized by the number of significant figures in the numbers expressed under that format, shown on the X-axis.

5 Additional Analyses

5.1 Evidence of Performance Asymmetry Across Numeric Conditions

In our main experiments, we report overall accuracy across the dataset. Here, we analyze performance separately for two subsets: questions requiring answers above a threshold (greater-than condition, e.g., "P/E ratio > 200") and those requiring answers below a threshold (less-than condition, e.g., "P/E ratio < 200").

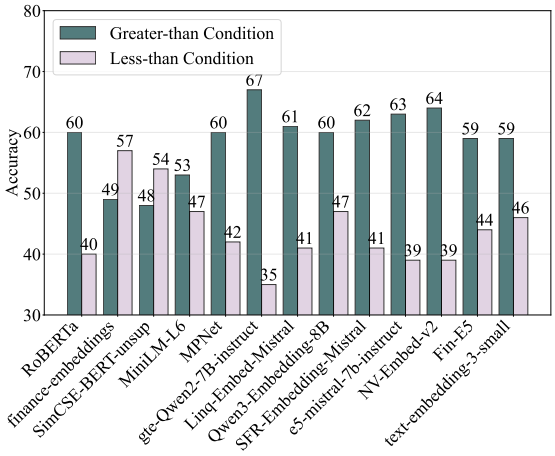

Figure 3: Model performance by numeric condition (greater-than vs. less-than).

Figure 3 shows embedding models perform significantly better on greater-than questions than less-than questions, indicating asymmetric reliability in numeric reasoning. This discrepancy highlights the need for caution in clinical and financial applications, where errors in certain numeric comparisons carry higher risks.

5.2 "Twenty-Four" versus "24"

We tested whether expressing numbers as words (e.g., "twenty-four" instead of "24") reduces embedding models' cognitive load. Evaluating numbers up to four digits, Figure 4 shows most models perform slightly better with written forms, likely because these use common vocabulary tokens that preserve numerical information more effectively.

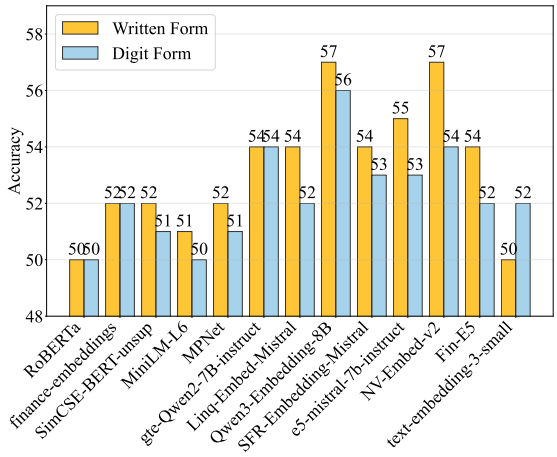

Figure 4: Model performance by numeric representation (digit form vs. written form).

However, even with written numbers, overall accuracy remains low, suggesting that **representing numerical content is an inherent limitation of embedding models rather than solely an OOV issue**. This calls for more reliable approaches to overcome the challenge, beyond simply expanding the model's vocabulary to cover numbers that were previously absent.

5.3 Does Seeing Numbers More Often Help Models Handle Them Better?

Previous research indicates that tokens with larger ℓ^2-norms appear more frequently in training data (Li et al., 2020; Yu et al., 2022). This inspires us to examine whether higher frequency exposure to numbers can translate into tangible numeracy gains. We split the evaluation data into high-frequency (top ℓ^2-norms half) and low-frequency (bottom ℓ^2-norms half) groups.

Figure 5 suggests that training data overrepresents extremes (small numbers like 1, 2 and large numbers like 1800, 2000), while mid-range values are underrepresented and have smaller embedding norms.

We present model performance across the two frequency-based groups in Figure 6. Surprisingly,

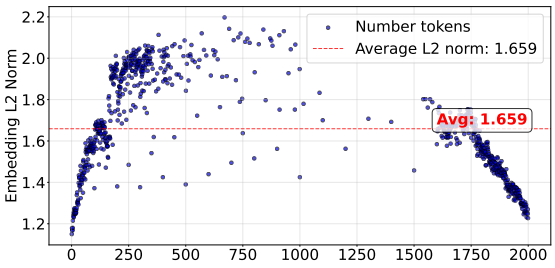

Figure 5: A visualization of the ℓ^2-norm distribution of number embedding from the finance-embedding token lookup table. The x-axis represents the numerical value.

the results do not indicate a clear, consistent performance advantage for high-frequency numbers over low-frequency ones. This suggests that **simply making pretraining data more number-intensive is unlikely to yield meaningful gains in numeracy and might be less cost-efficient**; instead, more targeted approaches for modeling numbers are needed.

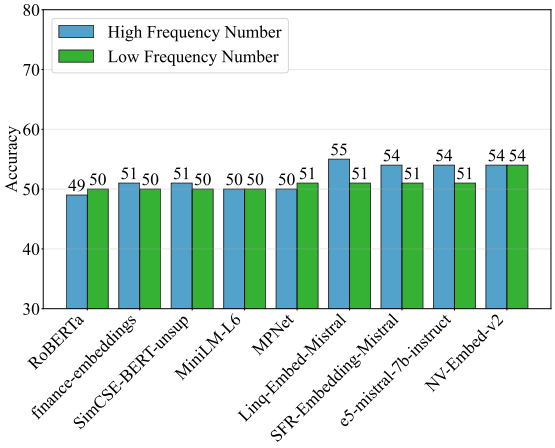

Figure 6: Model performance by number frequency (high-frequency numbers vs. low-frequency numbers).

5.4 Does Context Hide the Numbers? A Trade-Off in Embeddings

Previous work highlights a granularity trade-off in embeddings balancing broad semantics vs. fine details (Xu et al., 2025). We test whether contextualized embeddings (e.g., "P/E ratio > 200") lose numeric precision compared to context-free ones ("number > 200"). Using Qwen3-Embedding-8B (Zhang et al., 2025b), we encode these sentences and compute the cosine similarity between adjacent numbers (e.g., {X} and {X+1}). Results shown in Figure 7 .

In context-rich sentences, similarity scores between adjacent numbers cluster near 1.0, indicating

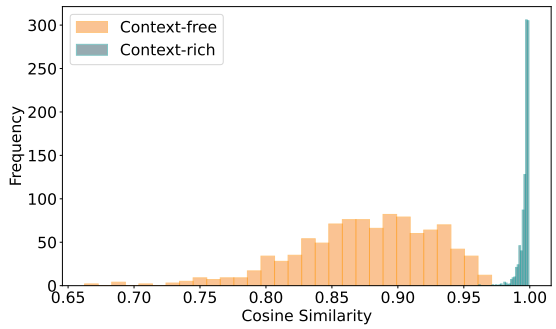

Figure 7: Distribution of embedding similarities by context condition (context-rich vs. context-free).

the embedding's capacity is dominated by semantic context, weakening the representation of fine-grained numerical details.

To further examine whether context accounts for low `EmbedNum-1K` performance, we remove context (e.g., adapting "Which stocks have a P/E ratio above 200?" to "Which number is above 200?" with answers "220" and "180". As shown in Figure 8).

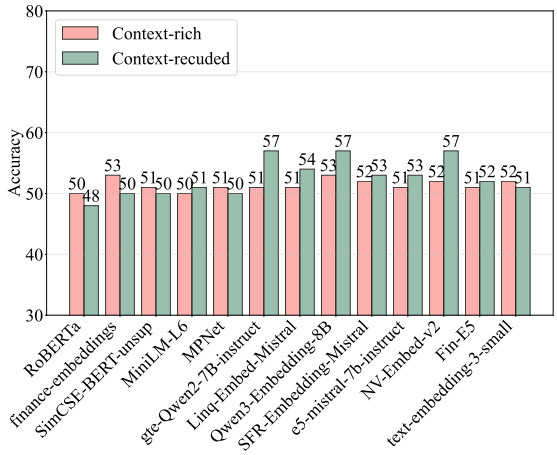

Figure 8: Model performance by context condition (context-rich vs. context-reduced).

Models performed better in context-reduced settings, implying that **additional context appears to attenuate the fine-grained numerical signals in embeddings**. This confirms the granularity dilemma (Xu et al., 2025) and, in turn, suggests that expanding embedding capacity or expressivity may constitute an essential step toward enhancing the numeracy of embeddings.

5.5 Can Probing Predict Downstream Numeracy Performance?

Probing tests (e.g., (Naik et al., 2019; Wallace et al., 2019)) decode numerical values from word embeddings (e.g., "seven" → "7"). We adapt this to text embeddings (e.g., "P/E of 220") by training a linear regressor on 70% of `EmbedNum-1K` embeddings to predict contained numbers, evaluated via Adjusted R^2 scores (Table 3). Encoder-based models outperform LLM-based ones, suggesting numerical signals are more accessible in encoder embeddings.

Model Name	Embedding Dimension	Adjusted R^2
Encoder-Based Models		
RoBERTa	768	1.25
finance-embeddings	768	1.99
SimCSE-BERT-unsup	768	1.13
MiniLM-L6	384	4.98
MPNet	768	1.15
LLM-Based Models		
gte-Qwen2-7B-instruct	3584	1.03
Linq-Embed-Mistral	4096	1.01
Qwen3-Embedding-8B	4096	1.02
SFR-Embedding-Mistral	4096	1.01
e5-mistral-7b-instruct	4096	1.01
NV-Embed-v2	4096	1.01
Commercial Models		
Fin-E5	4096	1.01
text-embedding-3-small	1536	1.03

Table 3: Results of probing numerical values from text embeddings. Higher Adjusted R^2 values indicate a more accurate prediction of numerical values. A blue-white-red color gradient is overlaid on the Adjusted R^2 column, with warmer colors representing higher values and cooler colors representing lower values.

This observation contrasts with our main finding in the earlier `EmbedNum-1K` retrieval task, where LLM-based models outperform encoder-based ones. This suggests that **embeddings from which numerical values are more easily decoded do not necessarily imply more faithful preservation of meaningful numeracy for downstream tasks**. This questions the reliability of probing tests as an evaluation metric in context-rich settings and highlights the need for new metrics to assess embedding numeracy in such environments.

6 Conclusion

In this study, we examine numeracy in text embeddings. Our work addresses an important evaluation gap, where fine-grained numerical nuances in text are often overlooked in existing embedding model benchmarking. Our experimental results reveal a significant limitation of modern text embedding models in preserving meaningful numerical details within their embeddings. This limitation is observed in both encoder-based and LLM-

based models. Through further analyses, we gain deeper insights into embedding numeracy and provide potential guidance for improving embedding models' ability to manage numerical content. We hope this work can shed light on future research to strengthen embedding-based NLP systems in number-intensive application scenarios.

Limitations

Our work also has limitations that can be improved in future research. First, we focus primarily on number magnitude comparison, which is a common type of numerical reasoning. Other forms of numerical reasoning, such as arithmetic operations, ratio and percentage calculations, or date and time comparisons, are not explored in this study. Future work could extend our evaluation to include these more complex reasoning tasks. Second, our study relies on synthetic datasets to evaluate embedding numeracy. While synthetic data enables controlled experiments and clear analysis, it may not reflect the full complexity of real-world contexts, where numbers interact with more complex linguistic and contextual information. Future research could identify representative real-world applications and develop related benchmarks, and investigate embedding numeracy across diverse domains, such as healthcare, scientific texts, and other areas where accurate numerical understanding is essential. Finally, we focus this work primarily on sentence-level embeddings and do not investigate embeddings of larger text units, such as paragraphs or full documents. Future studies could build on our evaluation framework to examine numeracy in embeddings that encode larger contexts, providing a more comprehensive understanding of how numerical information is preserved at scale.

Ethical Considerations

In this section, we address potential ethical considerations arising from the introduction of our new dataset, EmbedNum-1K.

Dataset Source. EmbedNum-1K is constructed based on the English dataset BULL (Zhang et al., 2024). We extended the original questions and generated corresponding answers with the aid of AI assistants. The dataset is intended solely for academic research purposes, and any commercial use is strictly prohibited.

Use of AI Assistants. The authors acknowledge the use of ChatGPT for the following purposes: grammatical correction, improving the overall coherence of the manuscript, and providing assistance in coding. All AI-generated content has been reviewed and refined by the authors to ensure accuracy and alignment with the research objectives.

References

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Hanyu Duan, Yi Yang, and Kar Yan Tam. 2021. Learning numeracy: A simple yet effective number embedding approach using knowledge graph. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2597–2602.

Kenneth C. Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemi'nski, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Sergeevich Solomatin, and 67 others. 2025. Mmteb: Massive multilingual text embedding benchmark. *ArXiv*, abs/2502.13595.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910. Association for Computational Linguistics.

Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, and Kewei Tu. 2020. Learning numeral embedding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2586–2599.

Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, Jy yong Sohn, and Chanyeol Choi. 2024. Linq-embed-mistral: Elevating text retrieval with improved gpt data through task-specific control and quality refinement. Linq AI Research Blog.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jiaxin Liu, Yi Yang, and Kar Yan Tam. 2024. Beyond surface similarity: Detecting subtle semantic shifts in financial narratives. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2641–2652.

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*, abs/1907.11692.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral: Enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

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*, pages 3374–3380.

OpenAI. 2025. Openai (august 25 version). `https://api.openai.com/v1/embeddings`.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jasivan Sivakumar and Nafise Sadat Moosavi. 2023. Fermat: An alternative to accuracy for numerical reasoning. *ArXiv*, abs/2305.17491.

Jasivan Alex Sivakumar and Nafise Sadat Moosavi. 2025. How to leverage digit embeddings to represent numbers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7685–7697.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *ArXiv*, abs/2004.09297.

Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, and Lawrence Carin. 2020. Methods for numeracy-preserving word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4742–4753.

Yixuan Tang and Yi Yang. 2025. Finmteb: Finance massive text embedding benchmark. *ArXiv*, abs/2502.10990.

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663.

Avijit Thawani, Jay Pujara, and Filip Ilievski. 2021. Numeracy enhances the literacy of language models. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 6960–6967.

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.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *ArXiv*, abs/2401.00368.

Lionel Wong, Ayman Ali, Raymond M Xiong, Zejiang Shen, Yoon Kim, and Monica Agrawal. 2025. Position: Retrieval-augmented systems can be dangerous medical communicators. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Liyan Xu, Zhenlin Su, Mo Yu, Jiangnan Li, Fandong Meng, and Jie Zhou. 2025. Dense retrievers can fail on simple queries: Revealing the granularity dilemma of embeddings. *arXiv preprint arXiv:2506.08592*.

Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. Financial report chunking for effective retrieval augmented generation. *arXiv preprint arXiv:2402.05131*.

Sangwon Yu, Jongyoon Song, Heeseung Kim, Seongmin Lee, Woo-Jong Ryu, and Sungroh Yoon. 2022. Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29–45, Dublin, Ireland. Association for Computational Linguistics.

Shaoyun Yu, Chanyuan Gu, Kexin Huang, and Ping Li. 2024. Predicting the next sentence (not word) in large language models: What model-brain alignment tells us about discourse comprehension. *Science advances*, 10(21):eadn7744.

Chao Zhang, Yuren Mao, Yijiang Fan, Yu Mi, Yunjun Gao, Lu Chen, Dongfang Lou, and Jinshu Lin. 2024. Finsql: Model-agnostic llms-based text-to-sql framework for financial analysis. *Companion of the 2024 International Conference on Management of Data*.

Meishan Zhang, Xin Zhang, Xinping Zhao, Shouzheng Huang, Baotian Hu, and Min Zhang. 2025a. On the role of pretrained language models in general-purpose text embeddings: A survey. *arXiv preprint arXiv:2507.20783*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Prompt

As shown in Figure 9, the answer generation prompt employs a strategic transformation to enhance the model's numerical reasoning capabilities. Specifically, it systematically inverts comparative relationships, converting "greater than" to "less than" and vice versa, while preserving the underlying semantic structure. Exposing the model to mirrored versions of the same quantitative concepts creates a balanced dataset that mitigates potential biases in its understanding of numerical comparisons.

As shown in Figure 10. It is designed to produce numerical responses that match the requirements of the given question exactly. This prompt engineering approach enforces strict output constraints, requiring the model to generate answers containing precisely one numerical value, with no supplementary text or explanations.

B Multi Possible Answers

We conducted two additional large-scale experiments with expanded candidate sets: 10-candidate

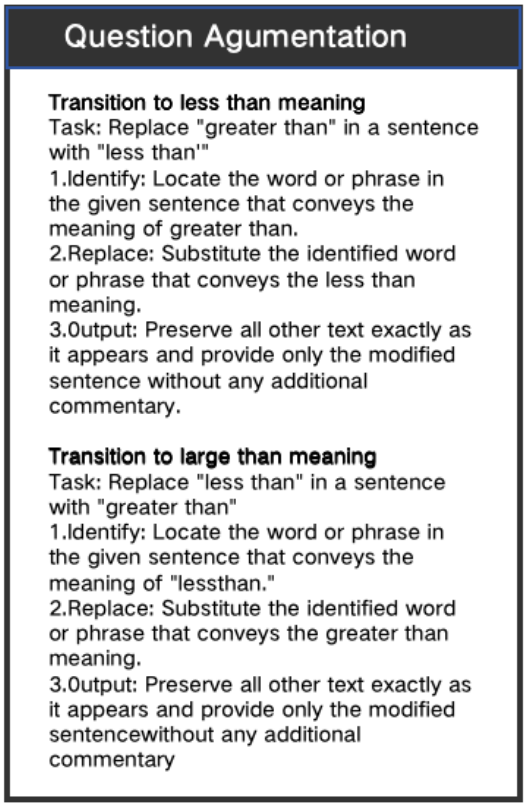

Figure 9: Prompt used for question augmentation.

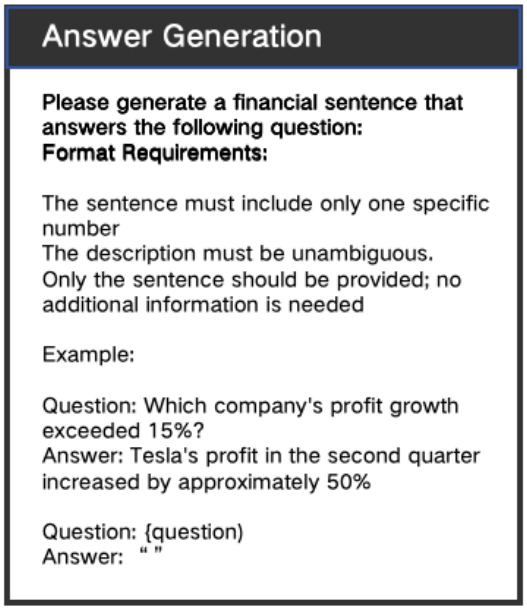

Figure 10: Prompt used for answer generation.

setting presented in Table 4;100 candidate setting presented in Table 5. Due to financial constraints, the 100-candidate experiment excluded commercial models that would have incurred substantial API costs

Table 4: Ranking Performance Across 10 Candidates

Model	P@2	P@5
Encoder-Based Models		
RoBERTa	49.75	49.78
finance-embedding	50.65	51.30
SimCSE-BERT-unsup	51.55	50.84
MiniLM-L6	49.80	50.38
MPNet	51.10	50.76
LLM-Based Models		
gte-Qwen2-7B-instruct	50.35	50.60
Linq-Embed-Mistral	49.80	50.22
Qwen3-Embedding-8B	53.05	52.18
e5-mistral-7b-instruct	50.30	50.20
NV-Embed	52.40	51.28
Commercial Models		
fine5	51.80	50.66
text-embedding-3-small	51.45	51.28

Table 5: Ranking Performance Across 100 Candidates

Model	P@5	P@10	P@20
Encoder-Based Models			
RoBERTa	49.60	49.67	49.72
finance-embedding	51.86	51.33	51.33
SimCSE-BERT-unsup	51.24	50.91	51.21
MiniLM-L6	50.10	50.18	49.75
MPNet	50.86	50.54	50.51
LLM-Based Models			
gte-Qwen2-7B-instruct	50.20	50.48	50.53
Linq-Embed-Mistral	50.70	50.66	50.20
Qwen3-Embedding-8B	**53.38**	**53.23**	**52.70**
e5-mistral-7b-instruct	51.00	50.70	50.11
NV-Embed-v2	52.44	52.27	51.62

Model performance rankings remain stable across candidate scales, with Qwen3-Embedding-8B consistently leading. Crucially, even in realistic settings with 10–100 candidates, the highest accuracy is only marginally above chance (53%), confirming a severe numeracy gap. This reflects a fundamental model limitation—not an artifact of the experimental design—as evidenced by consistently low performance across varying task complexities. This inherent deficiency critically undermines the reliability of embedding models in real-world numerical reasoning applications.

C Statistical Significance Testing

We performed statistical tests to evaluate our results. For each model and numeric format, a one-sample t-test compared model accuracy against the random baseline (0.5). The p-values (Table 6) show that only some models—predominantly LLMs—achieved statistically significant performance ($p < 0.05$, bolded), despite many accuracies being marginally above chance.

D Extreme Number

We extend our experiments to include evaluations with extreme values, implemented as follows: We randomly sampled integers between 2 and 998 as reference values, and generated corresponding values that were larger numbers and smaller numbers. These values were then scaled up or down by a factor of 10^{30}. The results show as Table 7:

Model Name	Large	Small
Encoder-Based Models		
RoBERTa	49.2	48.5
finance-embedding	49.1	52.6
SimCSE-BERT-unsup	50.1	50.0
MiniLM-L6	50.3	49.2
MPNet	50.0	50.1
LLM-Based Models		
gte-Qwen2-7B-instruct	53.0	55.9
Linq-Embed-Mistral	52.2	52.1
Qwen3-Embedding-8B	52.0	49.7
SFR-Embedding-Mistral	50.5	53.7
e5-mistral-7b-instruct	49.9	52.9
NV-Embed-v2	52.4	54.1
Commercial Model		
text-embedding-3-small	49.4	51.3

Table 7: Performance on Extreme Numerical Values.

The queries used were: "Which number is greater than query number?" and "Which number is less than query number? The candidate's answer is a number. Most models perform at around random chance (50% accuracy), with the best performing

Table 6: Statistical Significance Testing

Model Name	Orig	(a) Integers				(b) Decimals			
		(a1)	(a2)	(a3)	(a4)	(b1)	(b2)	(b3)	(b4)
Encoder-Based Models									
RoBERTa	0.32	0.46	0.34	0.64	0.59	0.77	0.23	0.54	0.36
finance-embedding	**0.02**	**0.02**	0.23	0.15	0.41	0.20	0.49	0.23	0.54
SimCSE-BERT-unsup	0.21	0.25	0.12	0.27	0.41	0.27	0.34	0.39	0.46
MiniLM-L6	0.36	0.46	0.25	0.75	0.41	0.27	0.56	0.44	0.49
MPNet	0.18	**0.03**	0.54	0.36	0.44	0.15	0.12	**0.04**	0.59
LLM-Based Models									
gte-Qwen2-7B-instruct	0.20	**0.00**	**0.00**	0.32	0.21	0.51	0.25	**0.04**	0.13
Linq-Embed-Mistral	0.20	**0.00**	**0.04**	0.51	0.85	0.34	0.39	0.41	0.75
Qwen3-Embedding-8B	**0.01**	**0.00**	**0.00**	**0.01**	0.08	0.18	**0.04**	0.30	0.09
SFR-Embedding-Mistral	0.10	**0.00**	0.07	0.49	0.70	0.13	0.36	0.20	0.30
e5-mistral-7b-instruct	0.18	**0.00**	0.21	0.46	0.70	0.06	0.21	0.59	0.36
NV-Embed-v2	0.10	**0.00**	**0.00**	0.32	0.73	0.01	0.09	0.23	0.23
Commercial Models									
Fin-E5	0.15	**0.00**	0.09	0.77	0.85	0.25	0.80	0.73	0.54
text-embedding-3-small	**0.05**	0.32	0.01	0.25	0.36	**0.00**	**0.01**	**0.00**	**0.00**

Model Name	(c) Scaled Down			(d) Scaled Up			(e) Others		
	(c1)	(c2)	(c3)	(d1)	(d2)	(d3)	(e1)	(e2)	(e3)
Encoder-Based Models									
RoBERTa	0.41	0.46	0.16	0.07	0.32	0.96	0.06	0.12	0.34
finance-embedding	**0.01**	0.54	**0.01**	**0.05**	**0.01**	0.25	**0.00**	**0.00**	0.27
SimCSE-BERT-unsup	0.10	**0.02**	0.09	0.34	**0.05**	0.27	0.13	0.07	0.30
MiniLM-L6	0.20	0.49	0.70	0.06	0.41	0.25	0.34	0.20	0.34
MPNet	0.06	0.05	**0.00**	0.36	0.20	**0.02**	**0.01**	**0.01**	0.12
LLM-Based Models									
gte-Qwen2-7B-instruct	**0.00**	**0.00**	**0.00**	**0.00**	0.27	**0.04**	**0.00**	**0.00**	0.21
Linq-Embed-Mistral	**0.00**	**0.00**	**0.00**	**0.00**	0.56	0.98	**0.00**	**0.00**	0.59
Qwen3-Embedding-8B	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.02**
SFR-Embedding-Mistral	**0.00**	**0.00**	**0.00**	**0.03**	0.54	0.82	**0.00**	**0.00**	0.82
e5-mistral-7b-instruct	**0.00**	**0.00**	**0.00**	**0.01**	0.32	0.68	**0.00**	**0.00**	0.68
NV-Embed-v2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.15	**0.00**	**0.00**	0.59
Commercial Models									
Fin-E5	**0.00**	**0.00**	**0.00**	**0.01**	0.09	0.82	**0.00**	**0.00**	0.68
text-embedding-3-small	**0.00**	**0.00**	**0.00**	**0.00**	0.21	0.46	**0.00**	**0.00**	0.39

at just over 55%. The models fail when faced with astronomically large or extremely small numbers.

E Currency Unit Sensitivity Analysis

We conducted an additional experiment by converting monetary values between USD(Dollars), JPY(Yen), YUAN(Yuan), and EURO(Euro). Based on the main experimental data, we removed all original units and systematically converted numerical values into four major currencies (Yuan, Dollar, Euro, Yen) using predefined exchange rates, shown in 8.

Table 8: Model Performance Across Different Currency Units (%)

Model Name	USD	EURO	JPY	YUAN
Encoder-Based Models				
RoBERTa	51.0	51.1	49.2	50.2
finance-embedding	51.3	51.7	49.4	51.2
SimCSE-BERT-unsup	50.9	50.5	50.0	51.0
MiniLM-L6	51.1	51.3	50.3	50.5
MPNet	51.3	51.8	51.7	51.3
LLM-Based Models				
gte-Qwen2-7B-instruct	52.8	54.5	48.7	49.5
Linq-Embed-Mistral	**56.2**	**55.2**	51.1	51.2
Qwen3-Embedding-8B	54.9	53.1	50.4	51.5
SFR-Embedding-Mistral	55.6	54.4	50.7	50.0
e5-mistral-7b-instruct	55.6	53.3	50.3	50.9
NV-Embed-v2	**56.3**	54.7	50.9	50.5
Commercial Models				
fine5	55.9	50.1	50.2	50.1
text-embedding-3-small	52.8	52.5	49.9	**52.4**

Our results demonstrate that embedding models exhibit varying performance across different currency units, even when the underlying numerical values remain identical. Most models perform better in dollars than in yen values. Linq-Embed-Mistral achieves 56.2% accuracy with dollars but drops to 51.1% with yen.

F Model Parameters and Sizes

Table 9 summarizes the parameter sizes of the embedding models evaluated in this study. The models are categorized into three groups: encoder-based models (ranging from 23M to 125M parameters), LLM-based models (all around 7–8B parameters), and commercial models. The parameter counts reflect the scale and computational requirements of each model. The commercial model text-embed-3-small is not publicly disclosed.

Model Name	Parameters
Encoder-Based Models	
RoBERTa	125M
finance-embedding	110M
SimCSE-BERT-unsup	110M
MiniLM-L6	23M
MPNet	110M
LLM-Based Models	
gte-Qwen2-7B-instruct	7B
Linq-Embed-Mistral	7B
Qwen3-Embedding-8B	8B
SFR-Embedding-Mistral	7B
e5-mistral-7b-instruct	7B
NV-Embed-v2	7B
Commercial Models	
Fin-E5	7B
text-embedding-3-small	–

Table 9: Model Parameters and Sizes.