Investigating the Performance of Dense Retrievers for Queries with Numerical Conditions

Haruki Fujimaki[1][0009−0000−2209−7171] Makoto P. Kato[1][0000−0002−9351−0901]
s2313638@u.tsukuba.ac.jp mpkato@acm.org

University of Tsukuba, Tsukuba, Japan

Abstract. This study investigates the performance of dense retrieval models for queries with numerical conditions, such as "mountains higher than 2000m" and "laptops between 500 and 1000 dollars." Our experimental results revealed that while dense retrieval models were able to change search results according to numerical conditions, there was a large gap between the ideal performances and those achieved by dense retrieval models. We also found that the retrieval effectiveness varied between models depending on the type of numerical condition and expression. Furthermore, experimental results suggest that knowledge acquired during pre-training might influence the search results for queries containing numerical conditions, which could lead to biased search results. This study highlights the need for further improvements in the capability of dense retrieval models to handle numerical information, which is often required in e-commerce, medical, and financial domains.

Keywords: dense retriever · numeracy · numerical conditions

1 Introduction

Dense retrieval models have recently seen increased use across a wide variety of domains including e-commerce [17], financial [5], and medical [9] domains. However, it has been pointed out that language models have shown limited performance in understanding numerical information [6, 7, 19, 21]. This limitation could be inherited by dense retrieval models as they are built on such language models. Therefore, there is a concern that dense retrieval models cannot retrieve relevant search results for queries containing numerical information, which is often required in e-commerce, medical and financial information retrieval.

This paper investigates the performance of dense retrieval models particularly focusing on queries with numerical conditions, which we refer to as *NumQ* for short. Examples of NumQ include "mountains higher than 2000m" and "laptops between 500 and 1000 dollars." These types of numerical information could be easily handled if they were stored in appropriate fields in search systems. However, they are sometimes described only in unstructured texts. For instance, in an e-commerce domain, a user might issue a query "a drone with a maximum wind resistance of 30 km/h or more". This type of specific numerical information is often described only in product descriptions or reviews, rather than being

directly represented as a structured attribute. In a finance domain, a user may search for "companies with a market capitalization between 100 million and 1 billion dollars," or "articles mentioning interest rates above 5%." In a medical domain, a user may look for "studies on patients with blood pressure readings higher than 140/90 mmHg" or "medications with a dosage of 200 mg." In these domains, document retrieval models are expected to understand a numerical condition in a query and estimate the relevance of a document based on numerical values in it.

To study the performance of dense retrievers for NumQs from various perspectives, we formulate the following research questions:

- **RQ1**: Can dense retrievers effectively handle NumQs?
- **RQ2**: What types of NumQs can be effectively processed by dense retrievers?
- **RQ3**: Does the internal knowledge of language models influence the retrieval effectiveness for NumQs?

The main contribution of this paper is the in-depth analysis of dense retrievers for NumQs, which highlights the need for further improvements in the capability of dense retrieval models to handle numerical information.

2 Related Work

There are two categories of related work: (1) quantity-centric retrieval systems, and (2) quantity-awareness of retrieval systems. While the numerical understanding capabilities of language models have been a subject of recent interest, with works such as [21] exploring their limitations in numerical tasks, our focus is specifically on the retrieval performance of dense models given queries with numerical conditions, which we refer to as NumQs.

Quantity-centric retrieval systems process NumQs by extracting numerical facts from documents and matching these numerical facts to numerical conditions in NumQs [8, 1, 15]. Although these works succeeded in processing NumQs by introducing an additional pipeline specialized for numerical information, we are rather interested in the capability of existing dense retrievers for NumQs.

Quantity-awareness of retrieval systems has recently been studied by Almasian et al. [2]. Although our study has partial overlap with theirs in that both investigated the retrieval effectiveness of dense retrievers for NumQs, this paper extended their work in three ways: (1) we investigated the retrieval effectiveness of dense retrievers for different types of NumQs, (2) we explored the influence of language models to search results returned in response to NumQs, and (3) we studied whether language models specialized for numerical information could improve the retrieval effectiveness of dense retrievers for NumQs.

3 Experiments

3.1 Experimental Settings

To investigate the performance of dense retrieval models for NumQs, we constructed datasets by using both real-world and synthetic data. Real-world data

Table 1. Overview of the datasets.

	Movie Rev.	**Job Post**	**Comp. Emp.**	**Unit**	**Corp. Value**
Research questions	1, 3	1, 3	1, 3	2	2
Source	Movie	LinkedIn	LinkedIn	Synthetic	Synthetic
# of documents	6,048	17,614	16,287	134,919	53,946
# of queries	30,240	25,000	25,000	134,865	270,000

we used includes Movie dataset [4], containing movie titles, genres, production companies, and box office revenues, and LinkedIn dataset [12], including job postings and company employee counts collected from LinkedIn.

We generated document collections based on those real-world datasets by using several types of templates, e.g., "{Movie title} is an {Genres} movie produced by {Production company}. It has earned {Box office revenue} dollars in revenue." Assigning values of the Movie dataset to variables in this template, we obtained **Movie Revenue** dataset. Taking the same approach to the LinkedIn dataset, we obtained **Job Post** and **Company Employee** datasets. While documents generated by this approach is artificial, the distribution of entities and numerical values should be realistic.

Furthermore, we also developed fully synthetic datasets to conduct in-depth analysis. **Unit** dataset was developed to investigate the numeracy of retrieval models for different types of numerical values, and consisted of a wide variety of numerical values and named entities generated by Faker[1]. Template examples are "{Entity} are sold for {Value} dollars" and "{Entity} has an area of {Value} square meters." **Corporate Value** dataset was constructed to study the impact of the value scale to retrieval effectiveness, and contained documents formatted as follows: "{Company name} is a {Industry} company in {Country}. The company's value is {Corporate value}," where the corporate value can be one of 1-1,000, 1-1,000K, and 1-1,000M with different currency expressions, i.e., dollar, USD, and $. The company name was generated by Faker, while the industry was derived from the LinkedIn dataset.

The overview of each dataset are shown in Table 1. Full details can be found in our code repository[2].

All the queries used in our experiments contain numerical conditions. Letting v_d and v_q be the numerical value in a document and a query, respectively, five numerical conditions we used are defined as follows: Equal ($v_d = v_q$), Less ($v_d \leq v_q$), More ($v_d \geq v_q$), Around ($0.85 \times v_q \leq v_d \leq 1.15 \times v_q$), and Between ($v_q \leq v_d \leq v'_q$). These conditions are expressed in natural languages when they are used as queries, for example, "What are jobs that have a salary equal to 50,000 USD?" for an Equal query with $v_q = 50,000$. A numerical value in a query is randomly selected from those in a document collection unless otherwise stated. Since we have access to *ground-truth* numerical information of each document, we can systematically define a set of relevant documents for each query.

[1] `https://faker.readthedocs.io/`

[2] `https://github.com/kasys-lab/numQs_for_dense_retriever`

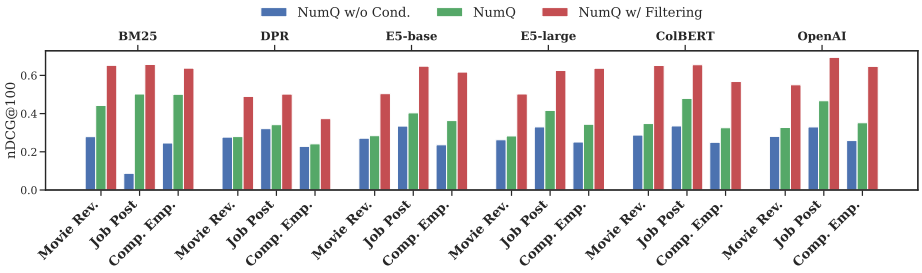

Fig. 1. Performance of each retrieval model in different conditions.

The retrieval models used in this study are BM25, DPR [10] (bert-base-uncased as a foundation model, fine-tuned with Natural Questions [13]), E5 [20] (E5-base[3] and E5-large[4]), and ColBERT[5] [11, 18], and OpenAI's embeddings [16] (text-embedding-3-small[6]). Given the potential for a large number of relevant documents in numerical condition queries, we chose nDCG@100 as our primary evaluation metric.

3.2 RQ1: Can Dense Retrievers Effectively Handle NumQs?

Figure 1 shows the performance of each retrieval model in three conditions: (1) NumQs without numerical conditions, (2) NumQs, and (3) NumQs with post-filtering by numerical conditions. In the condition (3), we filtered out irrelevant results of each retrieval model by referring to ground-truth numerical information of each result. Hence, (3) simulated the case where each retrieval model performed perfectly in terms of numerical conditions. The difference between (1) and (2) indicates how well each dense retrieval model handle numerical conditions in NumQ, while the difference between (2) and (3) indicates room for improvement. A Tukey's HSD test revealed significant differences of (1)-(2) and (2)-(3), except for DPR in the **Movie Revenue** dataset.

In most dense retrieval models, the performance of (2) was higher than that of (1), meaning that dense retrieval models is aware of numerical conditions in NumQs. However, there is a large gap between (2) and (3), indicating low capability of dense retrieval models for NumQs. Among the compared retrieval models, ColBERT showed relatively high performance, possibly because ColBERT employs term-level matching, enabling a more granular query-document matching. This trend accords with the finding from [2]. BM25 achieved superior performances compared to dense retrieval models, possibly due to its high effectiveness for Equal queries, as we will discuss shortly in the next subsection.

[3] `https://huggingface.co/intfloat/e5-base-v2`
[4] `https://huggingface.co/intfloat/e5-large-v2`
[5] `https://huggingface.co/colbert-ir/colbertv2.0`
[6] `https://platform.openai.com/docs/guides/embeddings/what-are-embeddings`

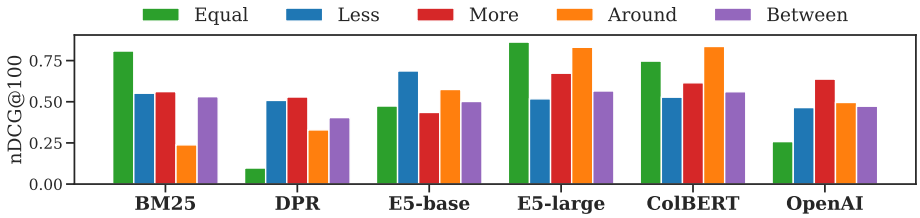

Fig. 2. Performance of retrieval models for each numerical condition.

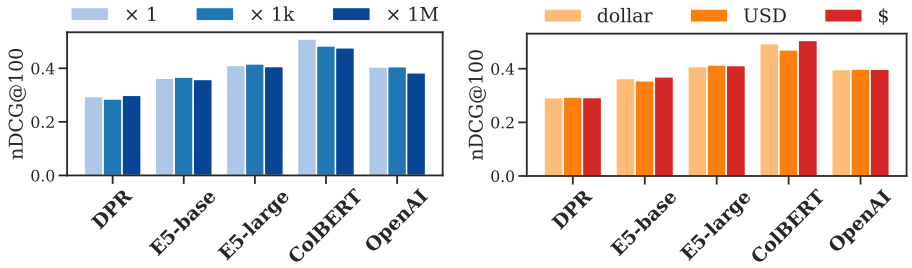

Fig. 3. Performance of retrieval models for each numerical expression.

3.3 RQ2: What Types of NumQs Can be Effectively Processed by Dense Retrievers?

To address RQ2, we conducted two types of analysis to explore which types of numerical conditions and expressions can be effectively handled by dense retrievers. The first analysis investigated the performance of each dense retriever across different numerical conditions, while the second analysis assessed the models' ability to process varied numerical expressions.

Numerical Conditions. Figure 2 illustrates the retrieval performance of each dense retriever for different numerical conditions in the **Unit** dataset. The differences of numerical conditions are statistically significant according to a Tukey's HSD test ($\alpha = 0.05$). Relatively effective retrievers, ColBERT and E5-large, performed well for Equal and Around queries, while their performance was limited for Less, More, and Between queries. This finding might suggest that these models can measure the similarity of numerical values, but are not fully capable of recognizing their magnitude relationship. Compared to these models, DPR and OpenAI showed significantly low performance for Equal queries. As expected, the lexical matching approach, BM25, achieved high effectiveness for Equal queries but not for the others.

Numerical Expressions. Figure 3 illustrates the search performance of each model for different numerical expressions in the **Corporate Value** dataset. A one-way ANOVA test was conducted for each model, revealing significant differences of E5-base, E5-large, and ColBERT, for different numerical expressions

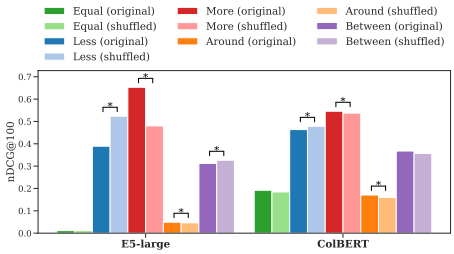

Fig. 4. Performance differences of the original and shuffled Movie Rev. datasets.

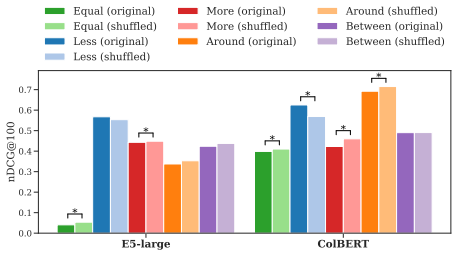

Fig. 5. Performance differences of the original and shuffled Job Post datasets.

($\alpha = 0.05$). In particular, the most effective retriever, ColBERT, showed significantly different performances: lower performance for M (the abbreviation for millions) than the other expressions, and USD than the other US dollar expressions. This finding suggests that different number expressions in training data could be useful for building robust quantity-aware retrievers.

3.4 RQ3: Does the Internal Knowledge of Language Models Influence the Retrieval Effectiveness for NumQs?

To address RQ3, we conducted two types of experiments. The first experiment was carried out to investigate the influence of the entity knowledge of language models in dense retrievers to the retrieval effectiveness for NumQs. Whereas, the second experiment was conducted to show the effectiveness of different language models, especially those pre-trained for understanding numerical values.

Entity Knowledge of Language Models. To investigate the influence of the entity knowledge of language models, we observed the performance difference between the original dataset and that with named entities shuffled across all the documents. As numerical information remained unchanged in the shuffled dataset, there should be no performance difference for the same set of NumQs. A significant performance difference indicates that dense retrievers heavily rely on non numerical information (i.e., named entities) to identify relevant documents for a numerical condition within a given NumQ.

Figures 4 and 5 show the performance differences between the original and shuffled versions of the **Movie Revenue** and **Job Post** datasets, respectively. We randomly shuffled *movie titles* and *job names* in these datasets. Results of only E5-large and ColBERT were presented in these figures due to a space constraint. Paired t-tests were conducted for each model's original and shuffled datasets, and Holm correction was applied to the p-values. Statistically significant differences ($\alpha = 0.05$) are indicated by $*$ in these figures. After the shuffling, in particular, E5-large showed a higher performance for Less queries, but a lower performance for More queries in Figure 4. This indicates that E5-large estimated the relevance based on movie titles, even though NumQs only included numerical

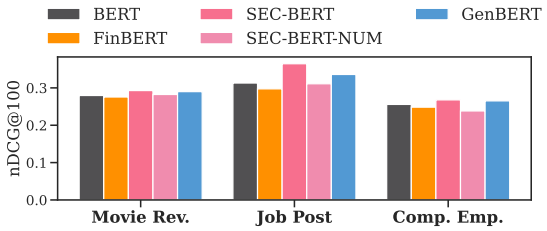

Fig. 6. Performance of DPR with different language models.

conditions for box office revenues. Since the performance for Less queries was improved in the shuffled dataset, we conjecture that E5-large tends to give a high score for major movies, which usually achieved high revenues. This might have a positive effect for More queries before the shuffling, while it might change to a negative effect after the shuffling. The opposite might happen to Less queries, resulting in the better performance after the shuffling. ColBERT shows smaller performance differences than E5-large in Figure 4, large differences are found for More and Less queries in Figure 5.

Numeracy of Language Models. In addition to BERT, we tested several language models pre-trained for higher numeracy: FinBERT [3], SEC-BERT (SEC-BERT and SEC-BERT-NUM) [14], and GenBERT [7]. FinBERT and SEC-BERT were additionally pre-trained with financial documents. SEC-BERT-NUM was further pre-trained with documents of which numerical values were replaced with pseudo tokens. GenBERT was additionally pre-trained with a large amount of artificial data specialized for numerical inference. With these language models as foundation models, DPR was trained with Natural Questions [13].

Figure 6 shows the performance of DPR with different language models. Two-way ANOVA tests revealed significant differences for all the datasets, and Tukey's HSD tests showed significant differences between all the model pairs except for (BERT, SEC-BERT-NUM) and (SEC-BERT, GenBERT) ($\alpha = 0.05$). Notably, SEC-BERT and GenBERT achieved stable improvements over the vanilla BERT. However, these performances are still far lower than the ideal performances presented in Figure 1. Therefore, a radical solution is possibly required for dense retrievers to accurately process NumQs.

4 Conclusion

This study investigated the performance of dense retrieval models for queries with numerical conditions. Our experimental results highlighted the need for further improvements in the capability of dense retrieval models to handle numerical information, which is often required in e-commerce, medical, and financial domains.

References

1. Almasian, S., Bruseva, M., Gertz, M.: QFinder: a framework for quantity-centric ranking. In: SIGIR. pp. 3272–3277 (2022)
2. Almasian, S., Bruseva, M., Gertz, M.: Numbers matter! bringing quantity-awareness to retrieval systems. arXiv preprint arXiv:2407.10283 (2024)
3. Araci, D.: FinBERT: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063 (2019)
4. Banik, R.: The movies dataset. `https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset` (2017), accessed: 2024-10-15
5. Chen, Z., Chen, W., Smiley, C., Shah, S., Borova, I., Langdon, D., Moussa, R., Beane, M., Huang, T.H., Routledge, B., Wang, W.Y.: Finqa: A dataset of numerical reasoning over financial data. In: EMNLP. pp. 3697–3711 (2021)
6. Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M.: Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In: NAACL-HLT. pp. 2368–2378 (2019)
7. Geva, M., Gupta, A., Berant, J.: Injecting numerical reasoning skills into language models. In: ACL. pp. 946–958 (2020)
8. Ho, V.T., Ibrahim, Y., Pal, K., Berberich, K., Weikum, G.: Qsearch: Answering quantity queries from text. In: ISWC. pp. 237–257 (2019)
9. Jin, D., Pan, E., Oufattole, N., Weng, W.H., Fang, H., Szolovits, P.: What disease does this patient have? a large-scale open domain question answering dataset from medical exams. Applied Sciences **11**(14), 6421 (2021)
10. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering (2020), `https://arxiv.org/abs/2004.04906`
11. Khattab, O., Zaharia, M.: 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. p. 39–48. SIGIR '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3397271.3401075, `https://doi.org/10.1145/3397271.3401075`
12. Kon, A., Zou, Z.Y.: Linkedin job postings (2023 - 2024). `https://www.kaggle.com/datasets/arshkon/linkedin-job-postings` (2023), accessed: 2024-10-15
13. Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al.: Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics **7**, 453–466 (2019)
14. Loukas, L., Fergadiotis, M., Chalkidis, I., Spyropoulou, E., Malakasiotis, P., Androutsopoulos, I., George, P.: FiNER: Financial numeric entity recognition for xbrl tagging. In: ACL (2022)
15. Maiya, A.S., Visser, D., Wan, A.: Mining measured information from text. In: SIGIR. pp. 899–902 (2015)
16. Neelakantan, A., Xu, T., Puri, R., Radford, A., Han, J.M., Tworek, J., Yuan, Q., Tezak, N., Kim, J.W., Hallacy, C., Heidecke, J., Shyam, P., Power, B., Nekoul, T.E., Sastry, G., Krueger, G., Schnurr, D., Such, F.P., Hsu, K., Thompson, M., Khan, T., Sherbakov, T., Jang, J., Welinder, P., Weng, L.: Text and code embeddings by contrastive pre-training (2022), `https://arxiv.org/abs/2201.10005`
17. Reddy, C.K., Màrquez, L., Valero, F., Rao, N., Zaragoza, H., Bandyopadhyay, S., Biswas, A., Xing, A., Subbian, K.: Shopping queries dataset: A large-scale ESCI benchmark for improving product search. arXiv preprint arXiv:2206.06588 (2022)

18. Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., Zaharia, M.: ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 3715–3734. Association for Computational Linguistics, Seattle, United States (Jul 2022). https://doi.org/10.18653/v1/2022.naacl-main.272, https://aclanthology.org/2022.naacl-main.272
19. Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M.: Do NLP models know numbers? probing numeracy in embeddings. In: EMNLP-IJCNLP. pp. 5307–5315 (2019)
20. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., Wei, F.: Text embeddings by weakly-supervised contrastive pre-training (2024), https://arxiv.org/abs/2212.03533
21. Yang, H., Hu, Y., Kang, S., Lin, Z., Zhang, M.: Number cookbook: Number understanding of language models and how to improve it (2024), https://arxiv.org/abs/2411.03766