Understanding the Multi-vector Dense Retrieval Models

Qi Liu
liuqi_67@ruc.edu.cn
Renmin University of China
Beijing, China

Jiaxin Mao*
maojiaxin@gmail.com
Renmin University of China
Beijing, China

ABSTRACT

While dense retrieval has become a promising alternative to the traditional text retrieval models, such as BM25, some recent studies show that multi-vector dense retrieval models are more effective than the single-vector method in retrieval tasks. However, due to a lack of interpretability, why the multi-vector method outperforms its single-vector counterpart has not been fully studied. To fill this research gap, in this work, we investigate and compare the behaviors of single-vector and multi-vector models in retrieval. Specifically, we analyze the vocabulary distribution of dense representations by mapping them back to the sparse, vocabulary space. Our empirical findings show that the multi-vector representation has more lexical overlaps between queries and passages. Additionally, we show that this feature of multi-vector representation can enhance its ranking performance when a given passage can fulfill different information needs and thus can be retrieved by different queries. These results shed light on the internal mechanisms of multi-vector representation and may provide new perspectives for future research.

CCS CONCEPTS

• **Information systems → Language models**.

KEYWORDS

document retrieval, dense retrieval, explainability

ACM Reference Format:
Qi Liu and Jiaxin Mao. 2023. Understanding the Multi-vector Dense Retrieval Models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615282

1 INTRODUCTION

With the recent rapid development of pre-trained language models (PLMs), such as BERT [1], dense retrieval has become popular in the information retrieval community and achieved state-of-the-art ranking performance on multiple benchmarks [5, 6, 15]. Typically, these models leverage PLMs to encode queries and passages into one or more low-dimensional, dense vectors and use the vector similarity between the query and document vectors to measure the semantic relevance. Previous studies have demonstrated that these dense retrieval models substantially outperform traditional retrieval techniques such as BM25 in ranking effectiveness. Because the vector representations of passages and queries can be computed independently, efficient retrieval can be achieved by encoding the passage collection offline and using approximate nearest neighbor (ANN) search techniques to reduce the online retrieval latency.

One commonly used technique for dense retrieval is utilizing a *single vector*, usually the embedding of the [CLS] token or the average-pooled embeddings of all tokens, to represent a passage or query [5]. However, Luan et al. [8] argued that single-vector representations are information bottlenecks. Therefore, more sophisticated training strategies are needed to improve the expressive power of vectors and remedy the information bottleneck [2, 15]. Other researchers have proposed methods to use *multiple vectors* to represent queries and passages. For example, the ColBERT model represents each token in the query and passage as a vector and adopts the sum-of-max function to compute the ranking score [6]. Other approaches, such as MEBERT [8] and MVR [16], represent the query as a single vector, and the passage as a fixed number of vectors. Both of these methods can improve the ranking performance at the expense of some additional computational cost.

However, as the multi-vector approaches are usually complex and harder to interpret than single-vector methods, the behaviors and mechanisms of multi-vector representation have not been fully studied. Specifically, we still do not know *why and how the multi-vector representation outperforms the single-vector representation*. In brief, we aim to investigate the following research questions:

- **RQ1:** What are the differences in expressiveness between multi-vector and single-vector representations?
- **RQ2:** When can multi-vector representations perform better than single-vector representations?

To address these questions, we propose a working hypothesis: there exists a mapping between dense, semantic representations and the sparse, lexical space, and multi-vector representations are more capable to capture the lexical features of queries, originated from different informational needs, after being projected into the vocabulary space.

Based on this hypothesis, for **RQ1**, we conduct a series of experiments to assess the expressiveness of the single-vector and multi-vector representation by mapping the corresponding dense vectors to the lexical space and investigating the exact matching and weight distribution of vocabulary [11]. Our findings demonstrate that the lexical overlap in the vocabulary projection between passages and queries is more evident in multi-vector representation in comparison to single-vector representation. For **RQ2**, we analyze

[*] Corresponding author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615282

the ranking performance of different methods on the test collections in which a passage is relevant to multiple queries. We find that multi-vector representation can achieve better performance than single-vector representation when a given passage is required to serve different information needs and be retrieved by different queries. Our findings help to understand the expressiveness of multi-vector representation, and it may also offer new perspectives for future research on building multi-vector dense retrieval models.

2 RELATED WORK

The interpretability of neural IR models has been widely studied. MacAvaney et al. [9] and Wallat et al. [13] analyzed neural retrieval models via probing tasks and study their characteristics. Ram et al. [11] employed the technique of vocabulary projection to demonstrate that the representations generated by the dense retrieval model can be intuitively reflected in the lexical space. However, the behaviors and mechanisms of multi-vector dense retrieval models have not been explored in the existing work. Some researchers argued that modeling passages that contain multiple meanings as multi-vector representations can make it easier for queries from different information needs to retrieve them [14, 16], but they have not investigated the differences between multi-vector representations and single-vector representations in detail, nor have they analyzed whether multi-vector models do perform better in this special scenario.

In this work, we conduct a comprehensive analysis to investigate why and how multi-vector dense retrieval models outperform single-vector models and fill this research gap.

3 PRELIMINARY

In this section, we briefly introduce the passage retrieval task and the fine-tuning process of PLMs for retrieval. Given a query q and a large-scale passage collection $\mathcal{P}$, the task of passage retrieval is to retrieve a set of passages that are most relevant to the query q from $\mathcal{P}$. To achieve this, we first use fine-tuned PLMs to encode the query and each passage into low-dimensional dense vector representations:

$$\begin{aligned} \boldsymbol{e}_q &= \mathrm{Enc}_Q(q) \in \mathbb{R}^d \\ \{\boldsymbol{e}_p^{(i)}\}_{i=1}^m &= \mathrm{Enc}_P(p) \in \mathbb{R}^d. \end{aligned} \tag{1}$$

Note that the query q is usually represented as a single dense vector as it can be implemented efficiently with standard ANN search, while the passage p can be encoded to one or more vectors concerning different models [5, 8, 16]. For single-vector representation, the embedding of the [CLS] token is commonly used as the representation and $m = 1$, while for multi-vector representations, there are often heuristic methods employed to select m embeddings. Then the similarity score $s(q, p)$ is calculated as follows:

$$s(q,p) = \max_{i=1}^{m} \boldsymbol{e}_q^T \boldsymbol{e}_p^{(i)}. \tag{2}$$

To adapt the PLMs to downstream retrieval tasks, the model must be fine-tuned on large labeled retrieval datasets, and the following loss as negative log likelihood of the positive passage is commonly used to optimize the model [5]:

$$\mathcal{L} = -\log \frac{\exp(s(q,p^+))}{\exp(s(q,p^+)) + \sum_{p^- \in \mathcal{P}^-} \exp(s(q,p^-))}, \tag{3}$$

where p^+ is the positive passage that is relevant to the query, and $\mathcal{P}^-$ is the set of irrelevant negative passages sampled from the passage collection.

4 METHODOLOGY

We describe the experimental setup in Section 4.1 and the overall performance of different models in Section 4.2. Then we elaborate on the analysis methods and results in Section 4.3 and 4.4.

4.1 Experimental Setup

Dataset. We conduct all experiments in three widely used open-domain question-answering datasets: Nature Questions (NQ) [7], TriviaQA [4], and SQuAD [10]. These three datasets all employ the Wikipedia corpus as their passage collection. Following Karpukhin et al. [5], we use the preprocessed collection where each article is split into multiple, distinct paragraphs. The preprocessed collection includes about 21 million passages in all. We evaluate the retrieval performance of different models on the official test set of the three datasets and utilize the development sets for analysis.

Models. We compare and analyze four retrieval models, namely BM25, DPR, MEBERT, and MVR. BM25 [12] is a traditional sparse retrieval model using exact match algorithms. DPR [5] is a representative single-vector dense retrieval model. MEBERT [8] and MVR [16] are multi-vector dense retrieval models. In MEBERT, the entire passage is represented by the embeddings of the first m tokens. And MVR employs the addition of multiple special tokens to represent the passage, utilizing the embeddings of m different special tokens. In both cases, a single vector is utilized to represent the query.

Implementation Details. To make a fair comparison, we re-fine-tuned all models. Our code is based on the Tevatron toolkit [3]. We initialize all models with the weights of pre-trained BERT and fine-tune them following the training settings and hyperparameters of DPR [5]. Different from the original DPR using individual query encoder and passage encoder, for all models in our experiments, the dual encoder shares its weights. The m in MEBERT and MVR is set to 4 as a good compromise between efficiency and effectiveness. Notice that we do not adopt mined hard negatives and warm-up pre-training strategies which are widely used in recent works [2, 15, 16].

4.2 Overall Performance

Table 1 reports the retrieval performance of four different models on the three datasets from our reproduction. As we expected, both the single-vector and the multi-vector dense retrieval model significantly outperform traditional BM25 in NQ and TriviaQA, indicating the effectiveness of neural dense retrieval models.

The results on the SQuAD dataset present an exceptional case that does not align with our initial expectations. As discussed in [5], the relatively low performance of dense retrieval models is caused

Table 1: Retrieval performance of different models on the test set of Natural Questions, TriviaQA, and SQuAD.

Model	NQ			TriviaQA			SQuAD		
	R@5	R@20	R@100	R@5	R@20	R@100	R@5	R@20	R@100
BM25 [12]	-	59.1	73.7	-	66.9	76.7	-	**68.8**	**80.0**
DPR [5]	66.7	79.2	86.3	70.7	79.7	85.3	33.9	52.4	71.5
MEBERT [8]	**68.0**	**79.5**	**86.5**	71.7	**79.8**	**85.4**	34.3	52.8	71.5
MVR [16]	67.7	79.3	**86.5**	**71.9**	**79.8**	85.3	**34.4**	52.9	71.3

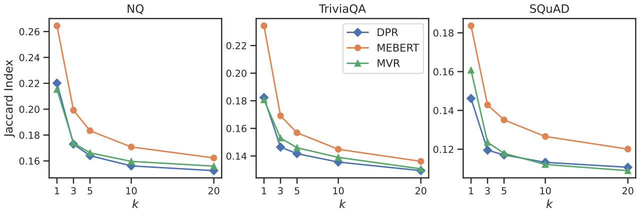

Figure 1: Jaccard index of different models on three datasets.

by a significant lexical overlap between the questions and paragraphs when constructing the dataset. Therefore, the BM25 algorithm based on exact matching may perform better. [1]

Meanwhile, compared with the single-vector retrieval model, the multi-vector retrieval model achieves better performance on all three datasets. This result is consistent with the findings of previous works [8, 16]. We believe that the multi-vector retrieval model can better capture the semantic information of the passage and the query, which is beneficial to retrieval, and we will discuss this in more detail later in Section 4.3 and 4.4.

4.3 Vocabulary Projection

The first analysis we conduct is to project the dense, semantic representations of queries and paragraphs onto the sparse, lexical space, using the masked language modeling (MLM) head of PLMs:

$$
\begin{aligned}
Q &= \text{MLM-Head}(\boldsymbol{e}_q) \in \mathbb{R}^{|V|} \\
P^{(i)} &= \text{MLM-Head}(\boldsymbol{e}_p^{(i)}) \in \mathbb{R}^{|V|},
\end{aligned} \tag{4}
$$

where $|V|$ is the size of the vocabulary.

Intuitively, we assume there exists a mapping between dense representations and the lexical space, so projecting the dense vectors onto the lexical space can help us analyze the semantic information learned by the dense representation and the difference between single-vector representations and multi-vector representations. To be more specific, as the MLM head is typically used in the MLM task to predict the masked tokens during pre-training, we can assume it had learned the mapping from dense representation to the lexical space. Therefore, the $|V|$-dimension output of the MLM head can be regarded as the probability distribution over the vocabulary.

Lexical Overlap. We analyze the lexical overlap between the query and the passage. Specifically, we select the tokens with top-k probability in Q and $P^{(i)}$, denoted as Q_k and $P_k^{(i)}$ respectively, and compute the Jaccard index to measure the lexical overlap between these two sets. For multi-vector representation, since we determine the similarity between a query vector and multiple passage vectors by computing the maximum similarity score, we also utilize the maximum value as a measure in this case. The Jaccard index is defined as:

$$
\text{Jaccard}(q, p, k) = \max_{i}^{m} \frac{|Q_k \cap P_k^{(i)}|}{|Q_k \cup P_k^{(i)}|}. \tag{5}
$$

The results on the development sets are shown in Figure 1. We can see that the Jaccard index of the multi-vector retrieval model is higher than that of the single-vector retrieval model on all three datasets. This indicates that the multi-vector retrieval model can better capture the lexical overlap between the query and the passage, which is beneficial to retrieval.

Table 2: The occurrence of a passage as a positive appears in 1, 2, or ≥3 queries. Because only the first positive passage of one query will be selected during training though there may be multiple positives, the statistic of the training set follows the same rule. But for the dev set, we select all positive passages since all positives that include answers will be a concern while evaluating.

	Train			Dev		
	1	2	≥3	1	2	≥3
NQ	32030	4944	3526	42290	2398	470
TriviaQA	43325	5306	1782	73161	3602	418
SQuAD	8473	6064	11772	37761	1848	240

4.4 Passages Retrieved by Multiple Queries

To further investigate in what situation the multi-vector dense retrieval model outperforms the single-vector model, we create a specialized dataset based on the development set, in which each paragraph is relevant to multiple queries. We analyze the retrieval performance of different models on this dataset.

We first analyze the relationship between the number of passages and the number of queries in the datasets. To be more specific, we focus on the number of queries that can retrieve the same passage. Table 2 presents the number of queries corresponding to each passage in the training and development sets across different datasets. It is evident from Table 2 that each of the three datasets presents an issue where many passages are repeated as positive passages for multiple queries from different information needs. Moreover, we

[1] We can't reproduce the original results of DPR. The paper by Wu et al. [14] also reports similar results and our reproduced results align with theirs. For a fair comparison, we only report the results based on our reproduction.

Table 3: Retrieval performances on the conducted test set. The absolute difference between the multi-vector dense retrieval model and DPR is indicated in parentheses.

Model	NQ			TriviaQA			SQuAD		
	MRR@1	MRR@20	R@20	MRR@1	MRR@20	R@20	MRR@1	MRR@20	R@20
DPR	13.1	20.3	40.3	6.1	12.7	38.8	30.6	41.3	68.2
MEBERT	13.0 (-0.1)	20.2 (-0.1)	40.6 (+0.3)	6.1 (+0.0)	12.8 (+0.1)	38.7 (-0.1)	31.0 (+0.4)	41.3 (+0.0)	68.1 (-0.1)
MVR	13.4 (+0.3)	20.4 (+0.1)	40.8 (+0.5)	6.7 (+0.6)	13.6 (+0.9)	39.8 (+1.0)	31.0 (+0.4)	41.6 (+0.3)	69.5 (+1.3)

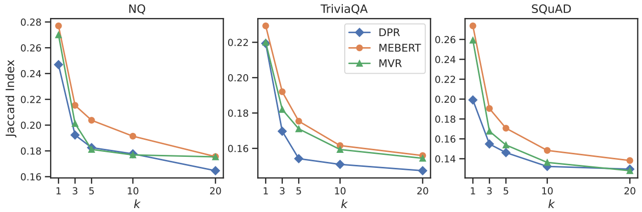

Figure 2: Jaccard index on the conducted test set.

believe that the single-vector retrieval model may have limitations when retrieving these passages. To verify our assumption, we select the passages that have more than or equal to 3 queries in the development set to conduct a new subset and use it for evaluation.

The results are shown in Table 3. In order to demonstrate the effectiveness of multi-vector retrieval in this scenario in terms of positioning relevant documents at the forefront, we also report the MRR as a reference. We can observe that the two multi-vector dense retrieval models exhibit different performances. MEBERT demonstrates only a slight performance improvement even worse compared to the single-vector model DPR, while MVR's performance is better than DPR across all different datasets. Based on these experimental results, we conjecture that MEBERT's multi-vector representation is constructed by selecting the embeddings of the top *m* tokens, which, while overcoming the information bottleneck through multiple vector representations, fails to capture the information of multiple semantics. Therefore, it still performs poorly when faced with different queries. In contrast, MVR encodes multiple vectors by introducing multiple special tokens of equal status, so MVR can better learn to model different semantic aspects of a single passage with multi-vector representations.

To further analyze the reason for the performance difference between MEBERT and MVR, We also analyze the lexical overlap between the query and the passage on this new dataset. The results are shown in Figure 2. We can observe that the Jaccard index of the multi-vector representation is still higher than that of the DPR. Moreover, in the new test set with multiple queries, the Jaccard index of the MVR model has significantly increased compared to the original development set, while the performance of MEBERT remains relatively stable. This confirms that MVR is indeed more adept at handling scenarios with multiple queries.

5 DISCUSSION

In this section, we discuss the results presented in Section 4 and respond to the research questions we posed in the introduction.

For **RQ1**, the retrieval performance of multi-vector dense retrieval models can be explained through the method of vocabulary projection. Our findings demonstrate that multi-vector representation has higher lexical overlap than single-vector after projection.

However, the higher lexical overlap between queries and passages in different multi-vector dense retrieval models may stem from different reasons. Concerning MEBERT, we posit that it is essentially an enhancement of the embedding of [CLS] token, resulting in increased lexical overlap and improved retrieval performance. As for the other vectors, we believe that their roles are more evident in training rather than retrieval, which is proven in our prior experiments. But for MVR, we believe that its higher lexical overlap stems from modeling different semantic aspects of passages across various vectors since these vectors all have equal status and tend to become more diverse during training [16].

The difference also leads to various performances of these two multi-vector dense retrieval models in retrieval, which contributes to the understanding of **RQ2**. Our experimental results demonstrate that the advantage of a multi-vector dense retrieval model such as MVR, becomes more pronounced when a passage will be retrieved by different queries. This is because its diverse multi-vector representation can better capture various semantic information in the passage, enabling it to be retrieved by different queries. In contrast, MEBERT or a single-vector dense retrieval model may encounter difficulties in this situation.

Our results suggest the need for training dense retrieval models that can represent the diverse information of each passage and fulfill queries originated from different information needs. It is necessary to benchmark dense retrieval models in this more realistic and challenging scenario and build more stable and powerful multi-vector models to further improve the retrieval performance.

6 CONCLUSION

In this paper, we investigate the behaviors and mechanisms of multi-vector dense retrieval models. Our analysis results show that the multi-vector dense retrieval model has more lexical overlaps between queries and passages when the dense, semantic vectors are projected to lexical, sparse vocabulary space. Moreover, we find that the multi-vector model's ranking performance is better when a passage needs to fulfill different information needs. These results may provide new perspectives for future research.

ACKNOWLEDGMENTS

This research was supported by the Natural Science Foundation of China (61902209), the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (22XNKJ15).

REFERENCES

[1] 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-HLT*. 4171–4186.

[2] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 981–993.

[3] Luyu Gao, Xueguang Ma, Jimmy J. Lin, and Jamie Callan. 2022. Tevatron: An Efficient and Flexible Toolkit for Dense Retrieval. *ArXiv* abs/2203.05765 (2022).

[4] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1601–1611.

[5] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 6769–6781.

[6] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[7] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (2019), 452–466.

[8] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345.

[9] Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the Behavior of Neural IR Models. *Transactions of the Association for Computational Linguistics* 10 (2022), 224–239.

[10] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392.

[11] Ori Ram, Liat Bezalel, Adi Zicher, Yonatan Belinkov, Jonathan Berant, and Amir Globerson. 2022. What Are You Token About? Dense Retrieval as Distributions Over the Vocabulary. *arXiv preprint arXiv:2212.10380* (2022).

[12] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[13] Jonas Wallat, Fabian Beringer, Abhijit Anand, and Avishek Anand. 2023. Probing BERT for ranking abilities. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part II*. Springer, 255–273.

[14] Bohong Wu, Zhuosheng Zhang, Jinyuan Wang, and Hai Zhao. 2022. Sentence-aware contrastive learning for open-domain passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1062–1074.

[15] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[16] Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-View Document Representation Learning for Open-Domain Dense Retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 5990–6000.