

Multi-View Document Representation Learning for Open-Domain Dense Retrieval

Shunyu Zhang^{1,*}, Yaobo Liang², Ming Gong³, Daxin Jiang³, Nan Duan²

¹ Intelligent Computing and Machine Learning Lab, School of ASEE, Beihang University

² Microsoft Research Asia ³ Microsoft STC Asia

¹zhangshunyu@buaa.edu.cn

^{2,3}{yalia, migon, djiang, nanduan}@microsoft.com

Abstract

Dense retrieval has achieved impressive advances in first-stage retrieval from a large-scale document collection, which is built on bi-encoder architecture to produce single vector representation of query and document. However, a document can usually answer multiple potential queries from different views. So the single vector representation of a document is hard to match with multi-view queries, and faces a semantic mismatch problem. This paper proposes a multi-view document representation learning framework, aiming to produce multi-view embeddings to represent documents and enforce them to align with different queries. First, we propose a simple yet effective method of generating multiple embeddings through viewers. Second, to prevent multi-view embeddings from collapsing to the same one, we further propose a global-local loss with annealed temperature to encourage the multiple viewers to better align with different potential queries. Experiments show our method outperforms recent works and achieves state-of-the-art results.

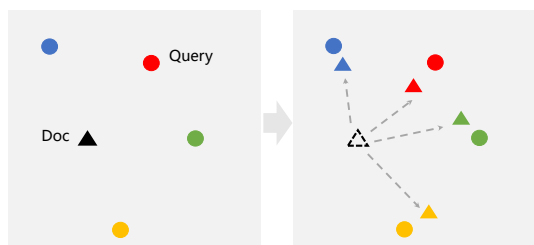
1 Introduction

Over the past few years, with the advancements in pre-trained language models (Devlin et al., 2019; Liu et al., 2019), dense retrieval has become an important and effective approach in open-domain text retrieval (Karpukhin et al., 2020; Lee et al., 2019; Qu et al., 2021; Xiong et al., 2020). A typical dense retriever usually adopts a bi-encoder (Huang et al., 2013; Reimers and Gurevych, 2019) architecture to encode input query and document into a single low-dimensional vector (usually CLS token), and computes the relevance scores between their representations. In real-world applications, the embedding vectors of all the documents are pre-

* Work done during internship at Microsoft Research Asia.

Title: iPod
Document: Beginning in mid-2007, four major airlines, United, Continental, Delta, and Emirates, reached agreements to install iPod seat connections. The free service will allow passengers to power and charge an iPod, and view video and music libraries on individual seat-back displays. Originally KLM and Air France were reported to be part of the deal with Apple, but they later released statements explaining that they were only contemplating the possibility of incorporating such systems. The iPod line can play several audio file formats including MP3 , AAC/M4A , Protected AAC , AIFF , WAV , Audible audiobook , and Apple Lossless . The iPod Photo introduced the ability to display JPEG, BMP, GIF, TIFF, and PNG image file formats.
Q1: Where can people using iPods on planes view the device's interface? A1: Individual seat-back displays.
Q2: What are two airlines that considered implementing iPod connections but did not join the 2007 agreement? A2: KLM and Air France.
Q3: What are some examples of audio formats supported by the iPod? A3: MP3, AAC/M4A, Protected AAC, AIFF, WAV, Audible audiobook, and Apple Lossless.
Q4: What is the name of an audio format developed by Apple? A4: Apple Lossless.

(a) An example from SQuAD Open Dataset.



(b) Our proposed MVR method.

Figure 1: The illustration of our multi-view document representation learning framework. The triangles and circles mean document and query vectors separately. Usually, one document can be asked to different potential queries from multiple views. Our method comes from this observation and generates multi-view representations for documents to better align with different potential queries.

-computed in advance, and the retrieval process can be efficiently boosted by the approximate nearest neighbor (ANN) technique (Johnson et al., 2019). To enhance bi-encoder's capacity, recent studies carefully design sophisticated methods to train it effectively, including constructing more challenging hard negatives (Zhan et al., 2021; Xiong et al., 2020; Qu et al., 2021) and continually pre-train the

language models (Gao and Callan, 2021a; Oğuz et al., 2021) for a better transfer.

However, being limited to the single vector representation, bi-encoder faces the upper boundary of representation capacity according to theoretical analysis in Luan et al. (2021). In the real example from SQuAD dev dataset, we also find that a single vector representation can't match well to multi-view queries, as shown in Figure.1. The document corresponds to four different questions from different views, and each of them matches to different sentences and answers. In the traditional bi-encoder, the document is represented to a single vector while it should be recalled by multiple diverse queries, which limits the capacity of the bi-encode.

As for the multi-vector models, cross-encoder architectures perform better by computing deeply-contextualized representations of query-document pairs, but are computationally expensive and impractical for first-stage large-scale retrieval (Reimers and Gurevych, 2019; Humeau et al., 2020). Some recent studies try to borrow from cross-encoder and extend bi-encoder by employing more delicate structures, which allow the multiple vector representations and dense interaction between query and document embeddings. including late interaction (Khattab and Zaharia, 2020) and attention-based aggregator (Humeau et al., 2020; Tang et al., 2021). However, most of them contain softmax or sum operators that can't be decomposed into max over inner products, and so fast ANN retrieval can't be directly applied.

Based on these observations, we propose *Multi-View document Representations learning* framework, *MVR* in short. MVR originates from our observation that a document commonly has several semantic units, and can answer multiple potential queries which contain individual semantic content. It is just like given a specified document, different askers raise different questions from diverse views. Therefore, we propose a simple yet effective method to generate multi-view representations through **viewers**, optimized by a global-local loss with annealed temperature to improve the representation space.

Prior work has found *[CLS]* token tends to aggregate the overall meaning of the whole input segment (Kovaleva et al., 2019; Clark et al., 2019), which is inconsistent with our goal of generating multi-view embeddings. So we first modify the

bi-encoder architecture, abandon *[CLS]* token and add multiple *[Viewer]* tokens to the document input. The representation of the viewers in the last layer is then used as the multi-view representations.

To encourage the multiple viewers to better align with different potential queries, we propose a global-local loss equipped with an annealed temperature. In the previous work, the contrastive loss between positive and negative samples is widely applied (Karpukhin et al., 2020). Apart from global contrastive loss, we formulate a local uniformity loss between multi-view document embeddings, to better keep the uniformity among multiple viewers and prevent them from collapsing into the same one. In addition, we adopt an annealed temperature which gradually sharpens the distribution of viewers, to help multiple viewers better match to different potential queries, which is also validated in our experiment.

The contributions of this paper are as follows:

- We propose a simple yet effective method to generate multi-view document representations through multiple viewers.
- To optimize the training of multiple viewers, we propose a global-local loss with annealed temperature to make multiple viewers to better align to different semantic views.
- Experimental results on open-domain retrieval datasets show that our approach achieves state-of-the-art retrieval performance. Further analysis proves the effectiveness of our method.

2 Background and Related Work

2.1 Retriever and Ranker Architecture

With the development of deep language model (Devlin et al., 2019), fine-tuned deep pre-trained BERT achieve advanced re-ranking performance (Dai and Callan, 2019; Nogueira and Cho, 2019). The initial approach is the cross-encoder based re-ranker, as shown in Figure.2(a). It feeds the concatenation of query and document text to BERT and outputs the *[CLS]* token's embeddings to produce a relevance score. Benefiting from deeply-contextualized representations of query-document pairs, the deep LM helps bridge both vocabulary mismatch and semantic mismatch. However, cross-encoder based rankers need computationally expensive cross-attention operations (Khattab and Zaharia, 2020; Gao and Callan, 2021a), so it is

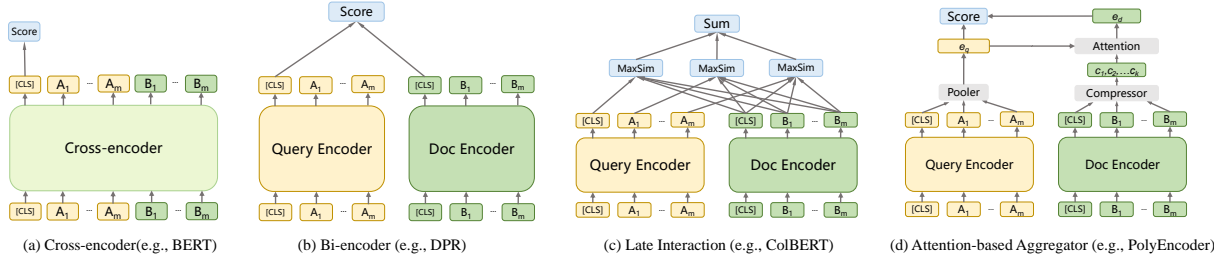


Figure 2: The comparison of different model architectures designed for retrieval/re-ranking.

impractical for large-scale first-stage retrieval and is usually deployed in second-stage re-ranking.

As for first-stage retrieval, bi-encoder is the most adopted architecture (Karpukhin et al., 2020) for it can be easily and efficiently employed with support from approximate nearest neighbor (ANN) (Johnson et al., 2019). As illustrated in Figure.2(b), it feeds the query and document to the individual encoders to generate single vector representations, and the relevance score is measured by the similarity of their embeddings. Equipped with deep LM, bi-encoder based retriever has achieved promising performance (Karpukhin et al., 2020). And later studies have further improved its performance through different carefully designed methods, which will be introduced in Sec.2.2

Beside the typical bi-encoder, there are some variants(Gao et al., 2020; Chen et al., 2020; Mehri and Eric, 2021) proposing to employ dense interactions based on Bi-encoder. As shown in Fig.2(c), ColBERT (Khattab and Zaharia, 2020) adopts the late interaction paradigm, which computes token-wise dot scores between all the terms’ vectors, sequentially followed by max-pooler and sum-pooler to produce a relevance score. Later on, Gao et al. (2021) improve it by scoring only on overlapping token vectors with inverted lists. Another variant is the attention-based aggregator, as shown in Fig.2(d). It utilizes the attention mechanism to compress the document embeddings to interact with the query vector for a final relevance score. There are several works (Humeau et al., 2020; Luan et al., 2021; Tang et al., 2021) built on this paradigm. Specifically, Poly-Encoder(learnt-k) (Humeau et al., 2020) sets k learnable codes to attend them over the document embeddings. DRPQ (Tang et al., 2021) achieve better results by iterative K-means clustering on the document vectors to generate multiple embeddings, followed by attention-based interaction with query. However, the dense interaction methods can’t be directly de-

ployed with ANN, because both the sum-pooler and attention operator can’t be decomposed into max over inner products, and the fast ANN search cannot be applied. So they usually first approximately recall a set of candidates then refine them by exhaustively re-ranking. While MVR can be directly applied in first-stage retrieval.

Another related method is ME-BERT(Luan et al., 2021), which adopts the first k document token embeddings as the document representation. However, only adopting the first-k embeddings may lose beneficial information in the latter part of the document, while our viewer tokens can extract from the whole document. In Sec.5.2, we also find the multiple embeddings in MEBERT will collapse to the same $[CLS]$, while our global-local loss can address this problem.

2.2 Effective Dense Retrieval

In addition to the aforementioned work focusing on the architecture design, there exist loads of work that proposes to improve the effectiveness of dense retrieval. Existing approaches of learning dense passage retriever can be divided into two categories: (1) pre-training for retrieval (Chang et al., 2020; Lee et al., 2019; Guu et al., 2020) and (2) fine-tuning pre-trained language models (PLMs) on labeled data (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2021).

In the first category, Lee et al. (2019) and Chang et al. (2020) propose different pre-training task and demonstrate the effectiveness of pre-training in dense retrievers. Recently, DPR-PAQ (Oğuz et al., 2021) proposes domain matched pre-training, while Condenser (Gao and Callan, 2021a,b) enforces the model to produce an information-rich CLS representation with continual pre-training.

As for the second class, recent work (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2021; Zhan et al., 2021) shows the key of fine-tuning an effective dense retriever revolves around hard nega-

tives. DPR (Karpukhin et al., 2020) adopts in-batch negatives and BM25 hard negatives. ANCE (Xiong et al., 2020) proposes to construct hard negatives dynamically during training. RocketQA (Qu et al., 2021; Ren et al., 2021b) shows the cross-encoder can filter and mine higher-quality hard negatives. Li et al. (2021) and Ren et al. (2021a) demonstrate that passage-centric and query-centric negatives can make the training more robust. It is worth mentioning that distilling the knowledge from cross-encoder-based re-ranker into bi-encoder-based retriever (Sachan et al., 2021; Izacard and Grave, 2021; Ren et al., 2021a,b; Zhang et al., 2021) can improve the bi-encoder’s performance. Most of these works are built upon bi-encoder and naturally inherit its limit of a single vector representation, while our work modified the bi-encoder to produce multi-view embeddings, and is also orthogonal to these strategies.

3 Methodology

3.1 Preliminary

We start with a bi-encoder using BERT as its backbone neural network, as shown in Figure 2(b). A typical Bi-encoder adopts dual encoder architecture which maps the query and document to single dimensional real-valued vectors separately.

Given a query q and a document collection $D = \{d_1, \dots, d_j, \dots, d_n\}$, dense retrievers leverage the same BERT encoder to get the representations of queries and documents. Then the similarity score $f(q, d)$ of query q and document d can be calculated with their dense representations:

$$f(q, d) = \text{sim}(E_Q(q), E_D(d)) \quad (1)$$

Where $\text{sim}(\cdot)$ is the similarity function to estimate the relevance between two embeddings, e.g., cosine distance, euclidean distance, etc. And the inner-product on the $[CLS]$ representations is a widely adopted setting (Karpukhin et al., 2020; Xiong et al., 2020).

A conventional contrastive-learning loss is widely applied for training query and passage encoders supervised by the target task’s training set. For a given query q , it computed negative log likelihood of a positive document d^+ against a set of negatives $\{d_1^-, d_2^-, \dots, d_l^-\}$.

$$\mathcal{L} = -\log \frac{e^{f(q, d^+)/\tau}}{e^{f(q, d^+)/\tau} + \sum_l e^{f(q, d_l^-)/\tau}} \quad (2)$$

Where τ is hyper-parameter of temperature-scaled factor, and an appropriate temperature can help in better optimization (Sachan et al., 2021; Li et al., 2021).

3.2 Multi-Viewer Based Architecture

Limited to single vector representation, the typical bi-encoder faces a challenge that a document contains multiple semantics and can be asked by different potential queries from multi-view. Though some previous studies incorporate dense interaction to allow multiple representations and somehow improve the effectiveness, they usually lead to more additional expensive computation and complicated structure. Therefore, we propose a simple yet effective method to produce multi-view representations by multiple viewers and we will describe it in detail.

As pre-trained BERT has benefited a wide scale of the downstream tasks including sentence-level ones, some work has found $[CLS]$ tend to aggregate the overall meaning of the whole sentence (Kovalova et al., 2019; Clark et al., 2019). However, our model tends to capture more fine-grained semantic units in a document, so we introduce **multiple viewers**. Rather than use the latent representation of the $[CLS]$ token, we adopt newly added multiple viewer tokens $[VIE]$ to replace $[CLS]$, which are randomly initialized. For documents input, we add different $[VIE_i]$ ($i=1,2,\dots, n$) at the beginning of sentence tokens. To avoid effect on the positional encoding of the original input sentences, we set all the position ids of $[VIE_i]$ to 0, and the document sentence tokens start from 1 as the original. Then We leverage the dual encoder to get the representations of queries and documents:

$$\begin{aligned} E(q) &= \text{Enc}_q([VIE] \circ q \circ [SEP]) \\ E(d) &= \text{Enc}_d([VIE_1] \cdots [VIE_n] \circ d \circ [SEP]) \end{aligned} \quad (3)$$

Where \circ is the concatenation operation. $[VIE]$ and $[SEP]$ are special tokens in BERT. Enc_q and Enc_d mean query and document encoder. We use the last layer hidden states as the query and document embeddings.

The representations of the $[VIE]$ tokens are used as representations of query q and document d , which are denoted as $E_0(q)$ and $E_i(d)$ ($i = 0, 1, \dots, k - 1$), respectively. As the query is much shorter than the document and usually represents one concrete meaning, we retain the typical setting to produce only one embedding for the query.

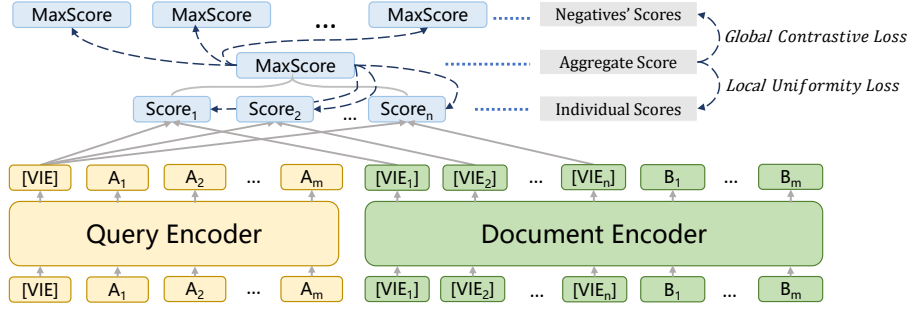


Figure 3: The general framework of multi-view representation learning with global-local loss. The gray blocks indicates the categories of scores in different layers.

Then the similarity score $f(q, d)$ of query q and document d can be calculated with their dense representations. As shown in Figure.3, we first compute the *Individual Scores* between the single query embedding and document’s multi-view embeddings, in which we adopt the inner-product. The resulted score corresponding to $[VIE_i]$ is denoted as $f_i(q, d)(i = 0, 1, \dots, k - 1)$. Then we adopt a max-pooler to aggregate individual score to the *Aggregate Score* $f(q, d)$ as the similarity score for the given query and document pairs:

$$\begin{aligned} f(q, d) &= \underset{i}{\text{Max}}\{f_i(q, d)\} \\ &= \underset{i}{\text{Max}}\{\text{sim}(E_0(q), E_i(d))\} \end{aligned} \quad (4)$$

3.3 Global-Local Loss

In order to encourage the multiple viewers to better align to different potential queries, we introduce a *Global-Local Loss* to optimize the training of multi-view architecture. It combines the global contrastive loss and the local uniformity loss.

$$\mathcal{L} = \mathcal{L}_{global} + \lambda\mathcal{L}_{local} \quad (5)$$

The global contrastive loss is inherited from the traditional bi-encoder. Given the query and a positive document d^+ against a set of negatives $\{d_1^-, d_2^-, \dots, d_l^-\}$, It is computed as follows:

$$\mathcal{L}_{global} = -\log \frac{e^{f(q, d^+)/\tau}}{e^{f(q, d^+)/\tau} + \sum_l e^{f(q, d_l^-)/\tau}} \quad (6)$$

To improve the uniformity of multi-view embedding space, we propose applying *Local Uniformity Loss* among the different viewers in Eq.7. For a specific query, one of the multi-view document representations will be matched by max-score in

Eq.4. The local uniformity loss enforces the selected viewer to more closely align with the query and distinguish from other viewers.

$$\mathcal{L}_{local} = -\log \frac{e^{f(q, d^+)/\tau}}{\sum_k e^{f_i(q, d^+)/\tau}} \quad (7)$$

To further encourage more different viewers to be activated, we adopt an annealed temperature in Eq.8, to gradually tune the sharpness of viewers’ softmax distribution. In the start stage of training with a high temperature, the softmax values tend to have a uniform distribution over the viewers, to make every viewer fair to be selected and get back gradient from train data. As the training process goes, the temperature decreases to make optimization more stable.

$$\tau = \max\{0.3, \exp(-\alpha t)\} \quad (8)$$

Where α is a hyper-parameter to control the annealing speed, t denotes the training epochs, and the temperature updates every epoch. To simplify the settings, we use the same annealed temperature in \mathcal{L}_{local} and \mathcal{L}_{global} and our experiments validate the annealed temperature works mainly in conjunction with \mathcal{L}_{local} through multiple viewers.

During inference, we build the index of all the reviewer embeddings of documents, and then our model directly retrieves from the built index leveraging approximate nearest neighbor (ANN) technique. However, both Poly-Encoder (Humeau et al., 2020) and DRPQ (Tang et al., 2021) adopt attention-based aggregator containing softmax or sum operator so that the fast ANN can’t be directly applied. Though DRPQ proposes to approximate softmax to max operation, it still needs to first recall a set of candidates then rerank them using the complex aggregator, leading to expensive computation and complicated procedure. In contrast, MVR

Method	SQuAD			Natural Question			Trivia QA		
	R@5	R@20	R@100	R@5	R@20	R@100	R@5	R@20	R@100
BM25 (Yang et al., 2017)	-	-	-	-	59.1	73.7	-	66.9	76.7
DPR (Karpukhin et al., 2020)	-	76.4	84.8	-	74.4	85.3	-	79.3	84.9
ANCE (Xiong et al., 2020)	-	-	-	-	81.9	87.5	-	80.3	85.3
RocketQA (Qu et al., 2021)	-	-	-	74.0	82.7	88.5	-	-	-
Condenser (Gao and Callan, 2021a)	-	-	-	-	83.2	88.4	-	81.9	86.2
DPR-PAQ (Oğuz et al., 2021)	-	-	-	74.5	83.7	88.6	-	-	-
DRPQ (Tang et al., 2021)	-	80.5	88.6	-	82.3	88.2	-	80.5	85.8
coCondenser (Gao and Callan, 2021b)	-	-	-	75.8	84.3	89.0	76.8	83.2	87.3
coCondenser(reproduced)	73.2	81.8	88.7	75.4	84.1	88.8	76.4	82.7	86.8
MVR	76.4	84.2	89.8	76.2	84.8	89.3	77.1	83.4	87.4

Table 1: Retrieval performance on SQuAD dev, Natural Question test and Trivia QA test. The best performing models are marked bold and the results unavailable are left blank.

can be directly applied in first-stage retrieval without post-computing process as them. Though the size of the index will grow by viewer number k , the time complexity can be sublinear in index size (Andoni et al., 2018) due to the efficiency of ANN technique (Johnson et al., 2019).

4 Experiments

4.1 Datasets

Natural Questions (Kwiatkowski et al., 2019) is a popular open-domain retrieval dataset, in which the questions are real Google search queries and answers were manually annotated from Wikipedia. **TriviaQA** (Joshi et al., 2017) contains a set of trivia questions with answers that were originally scraped from the Web.

SQuAD Open (Rajpurkar et al., 2016) contains the questions and answers originating from a reading comprehension dataset, and it has been used widely used for open-domain retrieval research.

We follow the same procedure in (Karpukhin et al., 2020) to preprocess and extract the passage candidate set from the English Wikipedia dump, resulting to about two million passages that are non-overlapping chunks of 100 words. Both NQ and TQA have about 60K training data after processing and SQuAD has 70k. Currently, the authors release all the datasets of NQ and TQ. For SQuAD, only the development set is available. So we conduct experiments on these three datasets, and evaluate the top5/20/100 accuracy on the SQuAD dev set and test set of NQ and TQ. We have counted how many queries correspond to one same document and the average number of queries of SQuAD, Natural Questions and Trivia QA are 2.7, 1.5 and 1.2, which indicates the multi-view problem is common

in open-domain retrieval.

4.2 Implementation Details

We train MVR model following the hyperparameter setting of DPR (Karpukhin et al., 2020). All models are trained for 40 epochs on 8 Tesla V100 32GB GPUs. We tune different viewer numbers on the SQuAD dataset and find the best is 8, then we adopt it on all the datasets. To make a fair comparison, we follow coCondenser (Gao and Callan, 2021b) to adopt mined hard negatives and warm-up pre-training strategies, which are also adopted in recent works (Oğuz et al., 2021; Gao and Callan, 2021a) and show promotion. Note that we only adopt these strategies when compared to them, while in the ablation studies our models are built only with the raw DPR model. During inference, we apply the passage encoder to encode all the passages and index them using the Faiss IndexFlatIP index (Johnson et al., 2019).

4.3 Retrieval Performance

We compare our MVR model with previous state-of-the-art methods. Among these methods, DRPQ (Tang et al., 2021) achieved the best results in multiple embeddings methods, which is the main compared baseline for our model. In addition, we also compare to the recent strong dense retriever, including ANCE (Xiong et al., 2020), RocketQA (Qu et al., 2021), Condenser (Gao and Callan, 2021a), DPR-PAQ (Oğuz et al., 2021) and coCondenser (Gao and Callan, 2021b). For coCondenser, we reproduced its results and find it a little lower than his reported one, maybe due to its private repo and tricks. Overall, these methods mainly focus on mining hard negative samples, knowledge distillation or pre-training strategies to

Models	R@5	R@20	R@100
DPR(k=1)	66.2	76.8	85.2
ME-BERT(k=4)	66.8	77.6	85.5
ME-BERT(k=8)	67.3	77.9	86.1
MVR(k=4)	68.5	78.5	85.8
MVR(k=6)	72.3	80.3	86.4
MVR(k=8)	75.5	83.2	87.9
MVR(k=12)	74.8	82.9	87.4

Table 2: Performance of different viewers’ number in MVR and compared models.

Models	R@5	R@20	R@100
(0) MVR ($\alpha = 0.1$)	75.5	83.2	87.9
(1) w/o LC loss	73.7	82.1	86.5
(2) w/o Annealed τ (Fixed=1)	74.3	81.9	86.8
(3) w/o LC loss + Annealed τ	72.8	81.0	85.8
(4) w/o Multiple Viewers	66.7	77.1	85.7
(5) Fixed $\tau = 10$	70.2	79.3	85.4
(6) Fixed $\tau = 0.3$	74.6	82.4	87.3
(7) Fixed $\tau = 0.1$	72.3	81.2	85.9
(8) Annealed $\tau(\alpha = 0.3)$	74.7	82.0	87.4
(9) Annealed $\tau(\alpha = 0.03)$	73.9	81.8	86.5

Table 3: Ablation study on Global-local Loss on SQuAD dev set.

improve dense retrieval. And our MVR framework is orthogonal to them and can be combined with them for better promotion.

As shown in Table 1, we can see that our proposed MVR performs better than other models. Compared to DRPQ which performs best in the previous multi-vector models, MVR can outperform it by a large margin, further confirming the superiority of our multi-view representation. It’s worth noting that our model improves more on the SQuAD dataset, maybe due to the dataset containing more documents that correspond to multiple queries as we state in Sec.4.1. It indicates that MVR can address the multi-view problem better than other models.

4.4 Ablation Study

Impact of Viewers’ Number: We conduct ablation studies on the development set of SQuAD open. For fair comparison, we build all the models mentioned in the following based on DPR toolkit, including MEBERT and MVR. The results are shown in Table 2, and the first block shows the results of DPR and MEBERT which adopt the first k token embeddings. Compared to DPR and MEBERT, our model shows strong performance, which indicates the multi-view representation is effective and useful. Then, we analyze how the different numbers of viewers ($k = 4, 6, 8, 12$) affect performance in

Method	Doc Encoding	Retrieval
DPR	2.5ms	10ms
ColBERT	2.5ms	320ms
MEBERT	2.5ms	25ms
DRPQ	5.3ms	45ms
MVR	2.5ms	25ms

Table 4: Time cost of online and offline computing in SQuAD retrieval task.

MVR. We find that the model achieves the best performance when $k = 8$. When k increase to $k = 12$ or larger, it leads little decrease in the performance, maybe due to there being not so many individual views in a document.

Analysis on Global-local Loss: In this part, we conduct more detailed ablation study and analysis of our proposed Global-local Loss. As shown in Table 3, we gradually reduce the strategies adopted in our model. We find not having either local uniformity loss (LC loss in table) or annealed temperature damages performance, and it decreases more without both of them. We also provide more experimental results to show the effectiveness of the annealed temperature. We first tune the fixed temperature, find it between 0.3 to 1 is beneficial. We adopt the annealed temperature annealed from 1 to 0.3 gradually as in Eq.8, finding a suitable speed($\alpha = 0.1$) can better help with optimization during training. Note that the model w/o Multiple Viewers can be seen as DPR with annealed τ , just little higher than raw DPR in Table 2, while annealed τ improves more when using multi-viewer. It indicates our annealed strategy plays a more important role in multi-view learning.

Efficiency Analysis: We test the efficiency of our model on 4 Nvidia Tesla V100 GPU for the SQuAD dev set, as shown in Table 4. We record the encoding time per document and retrieval time per query, and don’t include the query encoding time for it is equal for all the models. To compare our approach with other different models, we also record the retrieval time of other related models. We can see that our model spends the same encoding time as DPR, while DRPQ needs additional time to run K-means clustering. With the support of Faiss, the retrieval time MVR cost is near to DPR and less than ColBERT (Khattab and Zaharia, 2020) and DRPQ (Tang et al., 2021) which need additional post-computing as we state in Sec.2.1.

Models	R@5	R@20	R@100
DPR	66.2	76.8	85.2
MVR	75.5	83.2	87.9
Sentence-level	62.1	73.2	81.9
4-equal-splits	57.2	69.3	78.5
8-equal-splits	44.2	57.9	69.4

Table 5: Performance of different sentence-level retrieval models on SQuAD dev.

5 Further Analysis

5.1 Comparison to Sentence-level Retrieval

To analyze the difference between MVR and sentence-level retrieval which is another way to produce multiple embeddings, we design several models as shown in Table 5. Sentence-level means that we split all the passages into individual sentences with NLTK toolkit¹, resulting to an average of 5.5 sentences per passage. The new positives are the sentences containing answers in the original positives, and new negatives are all the split sentences of original negatives. K-equal-splits means the DPR’s original 100-words-long passages are split into k equal long sequences and training positives and negatives as Sentence-level’s methods. Compared to MVR, Sentence-level drops a lot even lower than DPR maybe for they lose contextual information in passages. It also indicates that the multi-view embeddings of MVR do not just correspond to sentence embeddings, but capture semantic meanings from different views. For even a single sentence may contain diverse information that can answer different potential queries (as in Fig.1). The k-equal-split methods perform worse much for it further lose the sentence structure and may contain more noise.

5.2 Analysis of Multi-View Embeddings

To further show the effectiveness of our proposed MVR framework, we evaluate the distribution of multi-view document representations. We conduct evaluations on the randomly sampled subset of SQuAD development set, which contains 1.5k query-doc pairs and each document has an average of 4.8 corresponding questions. We adopt two metrics *Local Variation* and *Perplexity* (Brown et al., 1992)(denoted as LV and PPL) to illustrate the effect of different methods. We first compute the normalized scores between the document’s multi-view embeddings and query embedding as in Eq.4,

¹www.nltk.org

Models	PPL	LV
ME-BERT	1.02	0.159
MVR	3.19	0.126
MVR w/o LC loss	3.23	0.052
MVR w/o Annealed τ	2.95	0.118

Table 6: Analysis of multi-view embeddings produced by different methods.

and record the scores $f_i(q, d)$ of all the viewers. Then Local Variation of a query-doc pair can be computed as follows, and then we average it on all the pairs.

$$LV = \text{Max}(f_i(q, d)) - \frac{\sum_k f_i(q, d) - \text{Max}(f_i(q, d))}{k - 1} \quad (9)$$

The Local Variation measures the distance of the max scores to the average of the others, which can curve the uniformity of different viewers. The higher it is, the more diversely the multi-view embeddings are distributed.

Then we collect the index of the viewer having the max score, and group the indexes of different queries corresponding to the same documents. Next, we can get the distributions of different indexes denoted as p_i . The Perplexity can be computed as follows:

$$PPL = \exp\left(-\sum_m p_i \log p_i\right) \quad (10)$$

If different viewers are matched to totally different queries, the p_i tends to be a uniform distribution and PPL goes up. The comparison results are shown in Table 6. When evaluating MEBERT, we find its multiple embeddings collapse into the same [CLS] embeddings rather than using the different token embeddings. So its PPL is near to one and Local Variation is too high. For MVR model, we find that without local uniformity loss (LC loss in short), the Local Variation drops rapidly, indicating our proposed LC loss can improve the uniformity of different viewers. In addition, MVR w/o annealed τ will damage the PPL, which also confirms it does help activate more viewers and align them better with different queries.

5.3 Qualitative Analysis

As shown in Table 7, there are two examples retrieved by DPR and MVR for qualitative analysis. The top scoring passages retrieved by DPR can’t give a clear answer for the queries, though they

Question	Passage received by DPR	Passage retrieved by MVR
What continent ranged over the majority of the southern hemisphere of earth in the Cambrian?	Title: Mesozoic ... and the Khingan Mountains in Manchuria. This orogeny was related to the opening of the Arctic Ocean and subduction of the North China and Siberian cratons under the Pacific Ocean. In contrast, the era featured the dramatic rifting of the supercontinent Pangaea, which gradually split into a northern continent, Laurasia, and a southern continent, Gondwana. This created the passive continental margin that characterizes most of the Atlantic coastline (such as along the U.S. East Coast) today. By the end of the era, the continents ...	Title: Geological history of Earth ... Laurentia, Baltica and Siberia remained independent continents following the break-up of the supercontinent of Pannotia . Gondwana started to drift toward the South Pole. Panthalassa covered most of the southern hemisphere, and minor oceans included the Proto-Tethys Ocean, Iapetus Ocean and Khanty Ocean. The Ordovician period started at a major extinction event called the Cambrian -Ordovician extinction event some time about 485.4 ± 1.9 Ma. During the Ordovician the southern continents were collected into a single continent called Gondwana. Gondwana started the period in ...
How long ago did the Ordovician period begin?	Title: Ordovician ... is a geologic period and system, the second of six periods of the Paleozoic Era. The Ordovician spans 41.2 million years from the end of the Cambrian Period million years ago (Mya) to the start of the Silurian Period Mya. The Ordovician, named after the Celtic tribe of the Ordovices, was defined by Charles Lapworth in 1879 to resolve a dispute between followers of Adam Sedgwick and Roderick Murchison, who were placing the same rock beds in northern Wales into the Cambrian and Silurian systems, respectively. ...	Title: Geological history of Earth ... Laurentia, Baltica and Siberia remained independent continents following the break-up of the supercontinent of Pannotia. Gondwana started to drift toward the South Pole. Panthalassa covered most of the southern hemisphere, and minor oceans included the Proto-Tethys Ocean, Iapetus Ocean and Khanty Ocean. The Ordovician period started at a major extinction event called the Cambrian-Ordovician extinction event some time about 485.4 ± 1.9 Ma . During the Ordovician the southern continents were collected into a single continent called Gondwana. Gondwana started the period in ...

Table 7: Examples of passages returned from DPR and MVR. Correct answers are written in **bold**.

seem to have a similar topic to the queries. In contrast, our MVR is able to return the correct answers when the passages contain rich information and diverse semantics. Take the second sample as an example, the passage retrieved by DPR is around *Ordovician* in the question but there are no more details answering the question. In comparison, MVR mines more fine-grained information in the passage and return the correct answer 485.4 ± 1.9 Ma (Ma means million years ago). It indicates that DPR can only capture the rough meaning of a passage from a general view, while our MVR is able to dive into the passage and capture more fine-grained semantic information from multiple views.

6 Conclusions

In this paper, we propose a novel Multi-View Representation Learning framework. Specifically, we present a simple yet effective method to generate multi-view document representations through multiple viewers. To optimize the training of multiple viewers, we propose a global-local loss with annealed temperature to enable multiple viewers to better align with different semantic views. We conduct experiments on three open-domain retrieval datasets, and achieve state-of-the-art retrieval performance. Our further analysis proves the effectiveness of different components in our method.

7 Acknowledgements

We thank Yuan Chai, Junhe Zhao, Yimin Fan, Junjie Huang and Hang Zhang for their discussions and suggestions on writing this paper.

References

- Alexandr Andoni, Piotr Indyk, and Ilya Razenshiteyn. 2018. Approximate nearest neighbor search in high dimensions. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3287–3318. World Scientific.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, and Robert L. Mercer. 1992. [An estimate of an upper bound for the entropy of English](#). *Computational Linguistics*, 18(1):31–40.
- Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. [Pre-training tasks for embedding-based large-scale retrieval](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Jiecao Chen, Liu Yang, Karthik Raman, Michael Bendersky, Jung-Jung Yeh, Yun Zhou, Marc Najork, Danyang Cai, and Ehsan Emadzadeh. 2020. [DiPair: Fast and accurate distillation for trillion-scale text matching and pair modeling](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2925–2937, Online. Association for Computational Linguistics.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? an analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Luyu Gao and Jamie Callan. 2021a. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 981–993.
- Luyu Gao and Jamie Callan. 2021b. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.
- Luyu Gao, Zhuyun Dai, and Jamie Callan. 2020. [Modularized transformer-based ranking framework](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4180–4190. Association for Computational Linguistics.
- Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. [COIL: Revisit exact lexical match in information retrieval with contextualized inverted list](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3030–3042, Online. Association for Computational Linguistics.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.
- Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. [Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Gautier Izacard and Edouard Grave. 2021. [Distilling knowledge from reader to retriever for question answering](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.
- Mandar Joshi, Eunsol Choi, Daniel 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)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. 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 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.
- 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.
- Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.
- Yizhi Li, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2021. More robust dense retrieval with contrastive dual learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 287–296.
- 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](#). *CoRR*, abs/1907.11692.
- 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*, 9:329–345.

- Shikib Mehri and Mihail Eric. 2021. [Example-driven intent prediction with observers](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2979–2992, Online. Association for Computational Linguistics.
- Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.
- Barlas Oğuz, Kushal Lakhota, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, et al. 2021. Domain-matched pre-training tasks for dense retrieval. *arXiv preprint arXiv:2107.13602*.
- Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. [RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.
- 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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021a. [PAIR: Leveraging passage-centric similarity relation for improving dense passage retrieval](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2173–2183, Online. Association for Computational Linguistics.
- Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021b. [Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835.
- Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. [End-to-end training of neural retrievers for open-domain question answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6648–6662, Online. Association for Computational Linguistics.
- Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. [Improving document representations by generating pseudo query embeddings for dense retrieval](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5054–5064, Online. Association for Computational Linguistics.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. [Approximate nearest neighbor negative contrastive learning for dense text retrieval](#). *CoRR*, abs/2007.00808.
- Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1253–1256.
- Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1503–1512.
- Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2021. Adversarial retriever-ranker for dense text retrieval. *arXiv preprint arXiv:2110.03611*.

A Scale Factor of Global-Local Loss

We have tuned the scale factor λ of the Global-local loss in Eq.5. The performances on SQuAD dev set are shown in Table 8. We find that a suitable scaling factor ($\lambda=0.01$) can improve more than others. Analysing other results, we infer that a large factor of local uniformity loss may lead to much impact on optimization of global loss, while a smaller one will degenerate into the form without local uniformity loss.

λ	R@5	R@20	R@100
0.5	72.4	80.4	85.9
0.05	74.7	82.5	87.3
0.01	75.5	83.2	87.9
0.001	72.9	82.2	85.7

Table 8: Performance on SQuAD dev set under different setting of scale factor.