An Encoder Attribution Analysis for Dense Passage Retriever in Open-Domain Question Answering

Anonymous ACL submission

Abstract

The bi-encoder design of dense passage retriever (DPR) is a key factor to its success in open-domain question answering (QA). However, it is unclear how DPR's question encoder and passage encoder individually contributes to the overall performance, which we refer to as the *encoder attribution* problem. The problem is important as it helps us isolate responsible factors for individual encoders to further improve overall performance. In this paper, we formulate our analysis under a probabilistic framework called *encoder marginalization*, where we quantify the contribution of a single encoder by marginalizing over other variables. We find that the passage encoder contributes more than the question encoder to the in-domain retrieval accuracy. We further use an example to demonstrate how to find the affecting factors for each encoder, where we train multiple DPR models with different amounts of data and use encoder marginalization to analyze the results. We find that the positive passage overlap and corpus coverage of training data have big impacts on the passage encoder, while the question encoder is mainly affected by training sample complexity under this setting. Based on this framework, we can devise data-efficient training regimes: for example, we manage to train a passage encoder on SQuAD using 60% less training data without loss of accuracy. These results illustrate the utility of our encoder attribution analysis.

1 Introduction

Attribution analysis, or credit assignment, concerns how individual components of a system contribute to its overall performance (Minsky, 1961). In this paper, we are interested in the *encoder attribution* problem of dense passage retrievers (DPR) (Karpukhin et al., 2020; Zhan et al., 2020b) for open-domain question answering (Voorhees and Tice, 2000; Chen et al., 2017). DPR leverages a bi-encoder structure that encodes questions and passages into low dimensional vectors separately. Follow-up work has proposed various methods to further improve and analyze DPR (Xiong et al., 2021; Luan et al., 2021; Mao et al., 2021; Gao and Callan, 2021). However, most of these methods only test the bi-encoder model in tandem, leaving two questions unanswered:

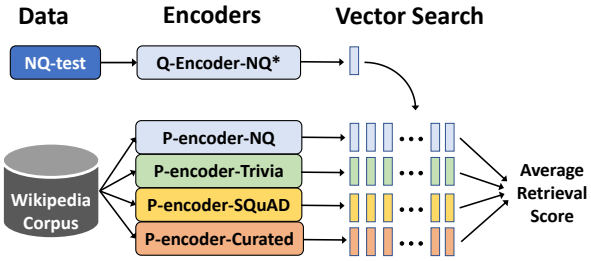


Figure 1: Encoder marginalization. "*": The target encoder we want to evaluate, where we use the Q-encoder of DPR trained on NQ as an example. The Q-encoder is evaluated on NQ-test data and paired with different P-encoders, and the final contribution is by averaging across the scores of different encoder pairings.

(1) *What are the individual contributions of each encoder of DPR?*

(2) *How to find the affecting factors for each encoder in different QA datasets?*

The first problem, which we refer to as *encoder attribution*, is important as it helps us understand which part of the DPR model might go wrong and identify possible sources of error in the data for the second problem. For example, if a DPR model fails to generalize to certain domains, it would be helpful to know whether the questions are out-of-distribution for the question encoder, or the passage encoding of the textual corpus is problematic. Therefore, it is important to separately inspect individual encoders of DPR.

In this paper, we perform an encoder attribution analysis of DPR under a probabilistic framework, where we model the evaluation function for

DPR's predictions as a Dirac delta distribution. The core component of our method is called *encoder marginalization*, where we target one encoder and marginalize over the other encoder variable in the Dirac delta distribution. We then use the expectation under the marginalized distribution as the encoder's contribution to the evaluation score. The marginalization can be approximated using Monte-Carlo as illustrated in Fig. 1, where we view the encoders trained from different domains as empirical samples from an encoder prior distribution which will be discussed in Section 6.

For question (1), we leverage encoder marginalization to compare the question encoder and passage encoder of the same DPR (Section 9). We find that in general, the passage encoder plays a more important role than the question encoder in terms of retrieval accuracy, as replacing the passage encoder causes a more significant performance drop.

For question (2), there are numerous affecting factors which we can not find them all in one paper. Therefore, we perform a case study where we analyze DPR's individual encoders under a data efficiency setting. We evaluate different DPR models trained with different amounts of data. Under this setting, we find that positive passage overlap and corpus coverage of the training data might be the affecting factors for the passage encoder, while the question encoder seems be affected by the sample complexity of training data. Based on the discovery of the affecting factors, we could develop a data-efficient training regime, where we manage to train a passage encoder on SQuAD using 60% less training data without loss of accuracy.

Our contributions in this paper are four-fold:

- To our knowledge, we formulate the first encoder attribution analysis for DPR under a probabilistic framework.
- We find that the passage encoder plays a more important role than the question encoder in terms of in-domain retrieval accuracy.
- Under a data efficiency setting, we identify that passage encoders are affected by positive passage overlap and corpus coverage of the training data, while question encoders are sensitive to the training sample complexity.
- Our framework enables the development of data-efficient training regimes where we are able to use up to 60% less training data without loss of accuracy.

2 Related Work

Attribution analysis It is also known as *credit assignment* and has long been discussed in various areas and applications. In reinforcement learning (Sutton and Barto, 1998), the accumulated reward from the environment needs to be distributed to the agent's historical decisions (Sutton, 1984; Harutyunyan et al., 2019; Arumugam et al., 2021). In investment (Binay, 2005), it is used to explain why a portfolio's performance differed from the benchmark. Attribution analysis has also been used in NLP (Mudrakarta et al., 2018; Jiang et al., 2021) and CV (Schulz et al., 2020) to interpret models' decisions. Therefore, attribution analysis is an important topic for understanding a system's behavior, especially for black-box models like deep neural networks (Goodfellow et al., 2016).

First-stage retrieval for QA The first-stage retrieval aims to efficiently find a set of candidate documents from a large corpus (Cai et al., 2021). Term-matching methods such as TF-IDF or BM25 (Robertson and Zaragoza, 2009; Lin et al., 2021) have established strong baselines in the first-stage retrieval of various QA tasks (Chen et al., 2017; Yang et al., 2019; Min et al., 2019). Recently, retrievers based on pre-trained language models (Devlin et al., 2019; Liu et al., 2019) also make great advancements (Seo et al., 2019; Lee et al., 2019; Guu et al., 2020; Khattab and Zaharia, 2020). Particularly, dense passage retrievers (DPR) (Karpukhin et al., 2020; Zhan et al., 2020b) set the milestone by encoding questions and passages separately with a bi-encoder design. Based on DPR, multiple works on compression (Yamada et al., 2021; Izacard et al., 2020; Ma et al., 2021), hard-negative mining (Xiong et al., 2021; Zhan et al., 2021), multi-vector encoding (Luan et al., 2021; Lee et al., 2021b), and QA pre-training (Lu et al., 2021; Gao and Callan, 2021) have further expanded the boundary of dense retrieval.

Other Analysis work of DPR BEIR investigates DPR's transferability over multiple retrieval tasks (Thakur et al., 2021), while Mr.TYDI evaluates DPR pre-trained on English corpus in a multilingual setting (Zhang et al., 2021). Lewis et al. (2021) finds that most of the test answers also occur somewhere in the training data for most QA datasets. Liu et al. (2021) observes that neural-retrievers fail to generalize to compositional questions and novel entities. Sciavolino et al. (2021)

also finds that dense models can only generalize to common question patterns.

3 Open-Domain Question Answering

Open-domain question-answering requires finding answers to given questions from a large collection of documents (Voorhees and Tice, 2000). For example, the question *"How many episodes in Season 2 Breaking Bad?"* is given and then the answer "13" will be either extracted from the retrieved passages or generated from a model. The goal of open-domain question answering is to learn a mapping from the questions to the answers, where the mapping could be a multi-stage pipeline that includes retrieval and extraction, or it could be a large language model that generate the answers directly given the questions. In this paper, we mainly discuss the retrieval component in the multi-stage system, which involves retrieving a set of candidate documents from a large text corpus. Based on the type of the corpus, we could further divide open-domain question answering into textual QA and knowledge base QA. Textual QA mines answers from unstructured text documents (e.g., Wikipedia) while the other one searches through a manually constructed knowledge base. We will mainly focus on textual QA in this paper.

4 Dense Passage Retrieval

Given a corpus of passages $\mathcal{C} = \{d_1, d_2, \cdots, d_n\}$ and a query q, DPR (Karpukhin et al., 2020) leverages two encoders η_Q and η_D to encode the question and documents separately. The similarity between the question q and document d is defined as the dot product of their vector output:

$$s = E_q^T E_d, \tag{1}$$

where $E_q = \eta_Q(q)$ and $E_d = \eta_D(d)$. The similarity score s will be used to rank the passages during retrieval. Both η_Q and η_D use the pre-trained BERT model (Devlin et al., 2019) for initialization and the `[CLS]` vector as the representation.

Training As pointed out by Karpukhin et al. (2020), training the encoders such that Eq. (1) becomes a good ranking function is essentially a metric learning problem (Kulis, 2012). Given a specific question q, let d^+ be the positive context that contains the answer a for q and $\{d_1^-, d_2^-, ...d_k^-\}$ be the negative contexts, the contrastive learning objective w.r.t. q, d^+, and $\{d_i^-\}_{i=1}^k$ is:

$$\begin{aligned}&\mathcal{L}(q, d^+, d_1^-, d_2^-, ...d_k^-)\\ &= -\log \frac{\exp(E_q^T E_{d^+})}{\exp(E_q^T E_{d^+}) + \sum_{i=1}^{k} \exp(E_q^T E_{d_i^-})}.\end{aligned} \tag{2}$$

The loss function in Eq. (2) encourages the representations of q and d^+ to be close and increases the distance between q and d^-.

Retrieval/Inference The bi-encoder design enables DPR to perform an approximate nearest neighbour search (ANN) using tools like FAISS (Johnson et al., 2017), where the representations of the corpus passages are indexed offline. It is typically used in first stage retrieval, where the goal is to retrieve all potentially relevant documents from the large corpus. Therefore, we consider the top-k accuracy as the evaluation metric in this paper following Karpukhin et al. (2020).

Let R be an evaluation function (e.g., top-k accuracy) for the first stage retrieval. Given a question-answer pair (q, a) and a corpus $\mathcal{C}$, we use η_Q and η_D to encode questions and retrieve passages separately. We define the evaluation score r_0 given the above inputs to be:

$$r_0 = R(q, a, \mathcal{C}, \eta_Q, \eta_D) \tag{3}$$

For simplicity's sake, in the rest of the paper, we will omit the answer a and corpus $\mathcal{C}$ as they are held fixed during evaluation.

5 Encoder Marginalization

In this section, we propose a simple probabilistic method to evaluate the contributions of encoders η_Q and η_D, as well as compare the same type of encoders across different datasets. The core component is called encoder marginalization, where marginalization simply means summing over the probability of possible values of a random variable.

Typically, the evaluation function R in Eq. (3) outputs a deterministic score r_0. However, we could also view r_0 as a specific value of a continuous random variable $r \in \mathbb{R}$ sampled from a Dirac delta distribution $p(r \mid q, \eta_Q, \eta_D)$:

$$\begin{aligned}p(r \mid q, \eta_Q, \eta_D) &\doteq \delta(r - r_0)\\ &= \begin{cases}+\infty, & r = r_0\\ 0, & r \neq r_0,\end{cases}\\ \text{s.t.,} \int_{-\infty}^{+\infty} &\delta(r - r_0)\mathrm{d}r = 1\end{aligned} \tag{4}$$

where $r_0 = R(q, a, \mathcal{C}, \eta_Q, \eta_D)$. Again, the answer a and corpus $\mathcal{C}$ are omitted for simplicity's sake.

The expectation of the evaluation score r under the Dirac delta distribution $\delta(r - r_0)$ is:

$$\begin{aligned}\mathbb{E}_{r\sim p(r|q,\eta_Q,\eta_D)}[r] &= \int_{-\infty}^{+\infty} r \cdot \delta(r - r_0)\mathrm{d}r \\ &= r_0 \end{aligned} \tag{5}$$

which is the score of the evaluation function in Eq. (3). This is also known as the *sifting property*[1] of the Dirac delta distribution (Mack, 2008), where the delta function is said to "sift out" the value at $r = r_0$. The reason for such a formalization is that now we could evaluate the contribution of a single encoder to the evaluation score r by marginalizing over the other random variables.

The contribution of an individual encoder η_Q or η_D to score r on a question q can be evaluated by marginalizing the other encoder of $p(r \mid q, \eta_Q, \eta_D)$ in Eq. (4). We assume that the question q is sampled from the training data distribution for learning η_Q and η_D. Let's take the question encoder η_Q as an example. The distribution of r after marginalizing over η_D is:

$$\begin{aligned} p(r \mid q, \eta_Q) &= \int_{\eta_D} p(r \mid q, \eta_Q, \eta_D)p(\eta_D)\mathrm{d}\eta_D \\ &\approx \frac{1}{K}\sum_{i=1}^{K} p(r \mid q, \eta_Q, \eta_D^{(i)}) \\ &= \frac{1}{K}\sum_{i=1}^{K} \delta(r - r_0^{(i)}) \end{aligned} \tag{6}$$

where the superscript (i) means the tagged random variables belong to the i^{th} out of K QA dataset (e.g., $\eta_D^{(i)}$ means the passage encoder trained on the i^{th} QA dataset). The second to the last step uses Monte-Carlo approximation, where we use $\eta_D^{(i)}$ sampled from a prior distribution $p(\eta_D)$ which will be discussed in Section 6.

The integration step in Eq. (6) assumes the independence between q, η_D, and η_Q. Although during training of DPR, η_D and η_Q are usually learned together, the two encoders do not necessarily need to be evaluated together during inference. For example, a question encoder trained on NQ could be paired with another passage encoder trained on Curated and tested on the Trivia QA dataset, without assuming any dependency among. Therefore, we here assume no prior knowledge about how η_D and η_Q are trained, but rather highlight their independence during evaluation to validate Eq. (6).

As for the contribution of η_Q, according to the expectation of Dirac delta distribution in Eq. (5), the expectation of r under the marginalized distribution in Eq. (6) is:

$$\begin{aligned}\mathbb{E}_{r\sim p(r|q,\eta_Q)}[r] &= \int_{-\infty}^{+\infty} r \cdot p(r \mid q, \eta_Q)\mathrm{d}r \\ &\approx \int_{-\infty}^{+\infty} r \cdot \frac{1}{K}\sum_{i=1}^{K} p(r \mid q, \eta_Q, \eta_D^{(i)})\mathrm{d}r \\ &= \frac{1}{K}\sum_{i=1}^{K}\int_{-\infty}^{+\infty} r \cdot \delta(r - r_0^{(i)})\mathrm{d}r \\ &= \frac{1}{K}\sum_{i=1}^{K} r_0^{(i)} \end{aligned} \tag{7}$$

which corresponds to the in-domain encoder marginalization in Fig. 1. In this way, we manage to calculate the contribution of a question encoder η_Q to the evaluation score r given a question q.

6 Encoder Prior Distribution, Sampling, and Approximation

In the previous section, we define the contribution of a single encoder for DPR using encoder marginalization. However, to approximate the expectation under the marginalized distribution in Eq. (6), we need to sample the encoder η_D from a prior distribution $p(\eta_D)$. In practice, we do not have access to $p(\eta_D)$ but instead we need to train η_D on specific datasets as empirical samples.

In addition, we can not consider every possible function for the encoder. Therefore, we need to put constraints on the encoder prior distribution, so that $p(\eta_D)$ becomes $p(\eta_D \mid \Phi)$ that implicitly conditions on some constraints Φ. In this paper, Φ could represent, for example, model structures, training schemes, optimizers, initialization, and so on. In this paper, the (sampled) encoders we run in the experiments are initialized with the same pre-trained language models (e.g., bert-base-uncased) and optimized with the same scheme (e.g., 40 epochs, Adam optimizers...), to ensure the constraints we put are consistent for different DPR models.

In practice, we use empirical samples such as DPRs pre-trained on different QA datasets for approximation in Eq. (7). Although the sample size is not big enough as it is very expensive to train DPR and encode a large textual corpus, the samples themselves are statistically meaningful as they are carefully fine-tuned at the domains we want to evaluate at, instead of using models with randomly initialized weights.

[1] This property requires the sifted function $g(r)$ (in this case, $g(r) = r$) to be Lipschitz continuous.

Datasets	Train	Dev	Test
Natural Questions	58,880	8,757	3,610
TriviaQA	60,413	8,837	11,313
WebQuestions	2,474	361	2,032
CuratedTREC	1,125	133	694
SQuAD	70,096	8,886	10,570

Table 1: Number of questions in each QA dataset from Karpukhin et al. (2020). The column of Train denotes the number of questions after filtering.

7 Experimental Setup

We follow the DPR paper (Karpukhin et al., 2020) to train and evaluate our dense retrievers. We reproduce their results on five benchmark datasets using Tevatron [2], an efficient toolkit for training dense retrievers with deep language models. Our reproduced results have only a maximum difference of $\sim$2% compared to their numbers. We report the top-20 and top-100 accuracy for evaluation.

Datasets We train individual DPR models on five standard benchmark QA tasks: Natural Questions (NQ) (Kwiatkowski et al., 2019), TriviaQA (Trivia) (Joshi et al., 2017), WebQuestions (WQ) (Berant et al., 2013), CuratedTREC (TREC) (Baudiš and Šedivỳ, 2015), SQuAD-1.1 (SQuAD) (Rajpurkar et al., 2016) as shown in Tbl. 1. We use the data provided in the DPR[3] repository to reproduce their results. We evaluate the retriever models on the test sets of the aforementioned datasets. For retrieval, we chunk the Wikipedia collections (Guu et al., 2020) into passages of 100 words as in Wang et al. (2019), which yields about 21 million samples in total. We follow Karpukhin et al. (2020) using BM25 (Robertson and Zaragoza, 2009; Lin et al., 2021) to select the positive and negative passages as the initial training data for DPR.

Models and Training During training, each question is paired with 1 positive passage, 1 hard negative retrieved by BM25, and $2 \times (B-1)$ in-batch negatives where B is the batch size. We optimize the objective in Eq. (2) with a learning rate of 1e-05 using Adam (Kingma and Ba, 2015) for 40 epochs. The rest of the hyperparameters remain the same as described in Karpukhin et al. (2020).

[2]https://github.com/texttron/tevatron
[3]https://github.com/facebookresearch/DPR

8 Generalization of Tandem Encoders

This section aims to show the generalization performance of DPR's bi-encoder evaluated in tandem. Tbl. 2 shows the zero-shot retrieval performance of different DPR models and BM25 on five benchmark QA datasets. Normally, the in-domain DPR model is expected to outperform the other DPR models trained from other domains, which is the situation that happens to most datasets such as NQ, Trivia, and SQuAD. However, for Curated, the DPR trained on NQ and Trivia has better zero-shot performance than the in-domain one. We suspect it is because NQ and Trivia have much larger training data than Curated as shown in Tbl. 1, which potentially covers some similar questions in Curated.

Moreover, BM25 outperforms all DPR models on SQuAD as SQuAD mainly contains entity-centred questions which is good for term-matching algorithms. Besides, the SQuAD dataset is mainly for machine reading comprehension and therefore a passage could be used to answer multiple questions, which could cause potential conflicts in representation learning (Wu et al., 2021).

In the following sections, we will perform encoder attribution analysis to examine DPR's each encoder individually.

9 In-Domain Encoder Marginalization

This section aims to answer the question (1) "*What are the individual contributions of each encoder of DPR?*" in Section 1. To analyze the contribution of a single encoder on a specific QA dataset, we compare the marginalized top-20 retrieval accuracy of the encoder using in-domain encoder marginalization shown in Fig. 1 and Eq. (7).

Fig. 2 shows the in-domain encoder marginalization results relative to the tandem DPR results. The blue bars show the question encoder's contributions where we target the question encoder and marginalize over the passage encoders, and vice versa for the orange bars (passage encoder) on five datasets. We further divide those results by the in-domain DPR performance which are normalized to 100% (the horizontal line in Fig. 2). We do not compare across different datasets, but rather compare the question encoder and passage encoder for each domain. We can see that in general, the passage encoder (orange bar) contributes more to the top-20 accuracy compared to the question encoder (blue bar) on all five datasets. Moreover, for the Curated dataset, marginalizing over the out-of-domain

Encoder \ Test set	NQ	Trivia	WQ	Curated	SQuAD	Average
BM25	62.9/78.3	62.4/75.5	76.4/83.2	80.7/89.9	**71.1/81.8**	70.7/81.7
DPR-NQ	**79.8/86.9**	73.2/81.7	68.8/79.3	86.7/92.7	54.5/70.2	**72.6/82.2**
DPR-Trivia	66.4/78.9	**80.2/85.5**	71.4/81.7	**87.3/93.9**	53.0/69.2	71.7/81.8
DPR-WQ	54.9/70.0	66.5/78.9	**76.0/82.9**	82.9/90.8	49.3/66.2	65.9/77.8
DPR-Curated	68.5/72.7	66.5/77.7	65.5/77.5	84.0/90.7	51.3/67.5	67.2/77.2
DPR-SQuAD	56.6/72.3	71.0/81.7	64.3/77.0	83.3/92.4	61.1/76.0	67.3/80.0

Table 2: Zero-shot evaluation of DPR's bi-encoder in tandem. Top-20/Top-100 retrieval accuracy (%) on five benchmark QA test sets is reported. Each score represents the percentage of top-20/100 retrieved passages that contain answers.

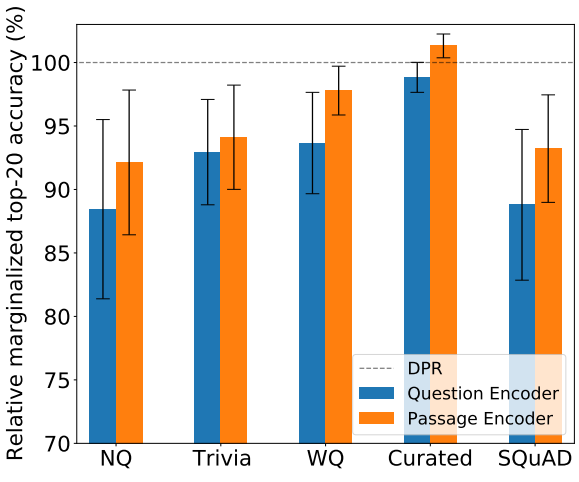


Figure 2: In-Domain marginalized top-20 accuracy (%) of each encoder relative to the in-domain DPR for each dataset using Eq. (7). Each in-domain DPR's top-20 accuracy is normalized to 100%.

question encoders even improves the marginalized performance of the passage encoder of Curated.

Overall, we could see that the passage encoder plays a more vital role compared to the question encoder in terms of in-domain retrieval accuracy, which makes sense as the passage encoder needs to encode the entire corpus (in our case, 21M passages), while the question sets are much smaller.

10 Affecting Factors for Encoders in QA Training Data

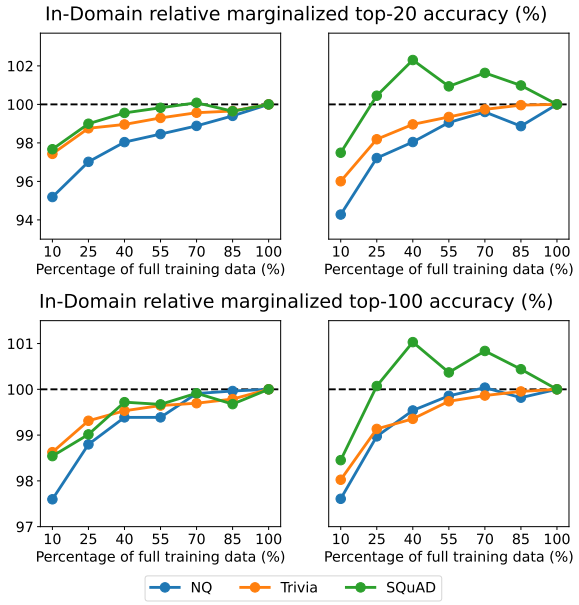
In this section, our goal is to answer the question (2) "*How to find the affecting factors for each encoder in different QA datasets?*" from Section 1. Obviously, there are too many affecting factors which we can not find them all in this paper. Therefore, we will use data efficiency test as an example and show how using encoder attribution in data efficiency test could help us locate possible affecting factors in the dataset. Specifically, we will train the DPR models with different amount of training data. The reason we choose to change the size of the training data is that data sizes often have major influences on a model's generalization performance, which could help reveal relevant affecting factors in the data.

10.1 In-Domain Data Efficiency Test

We train the DPR model with different amounts of data and test each encoder's in-domain marginalization performance w.r.t. the training data amount. Since it is extremely resource-consuming to train different DPR models and encode the entire Wikipedia corpus into dense vectors, in this section, we mainly focus on NQ, Trivia, and SQuAD due to their relatively large dataset sizes.

Fig. 3 shows the in-domain encoder marginalization results for both question encoder and passage encoder under a data efficiency setting, where we uniformly sample $10\%, 25\%, 40\%, 55\%, 70\%, 85\%$ of training data of each dataset to train DPR. We use in-domain encoder marginalization to evaluate each encoder's performance with different amounts of data. Specifically, to provide a fair comparison, we use 100% data trained DPR's encoders as the samples for all marginalization. For example, for the question encoder trained with 10% data, it will be paired with five passage encoders of DPR trained on five different domains with 100% data. This is to ensure the comparison between different question encoders is not affected by different ways of marginalization.

As we could see, the performance of the question encoder w.r.t. to different training data amounts (left column in Fig. 3) on three datasets improves



(a) Question encoder. (b) Passage Encoder.

Figure 3: In-Domain encoder marginalization results under a data efficiency setting. We train DPR on NQ, Trivia, and SQuAD with different amounts of training data. The marginalized top-20/100 accuracy (%) for each encoder is normalized. The y-axis is shared in each row. The horizontal line is the performance of an encoder trained with 100% data.

as the amount of training data increases. For the passage encoder (right column in Fig. 3), NQ's and Trivia's behave similarly to the question encoder (blue and orange lines of the right column in Fig. 3). However, the performance of SQuAD's passage encoder (green line of the right column in Fig. 3) shows a non-monotonic behaviour w.r.t. to the training data sizes at the $[40\%, 100\%]$ interval, where the performance first rises before 40% and drops afterwards. This means that besides the training sample complexity, there's more affecting factors that influence the performance of the passage encoder, which we will have further analysis in the following section.

10.2 Factor Analysis

Based on the results in the previous section, we now propose two possible affecting factors in the training data for the question encoder and passage encoder: *corpus coverage* and *positive passage overlap* defined as follows:

- **Corpus coverage**: Number of distinct positive passages in the training data (i.e., with different texts and titles in Wikipedia corpus).

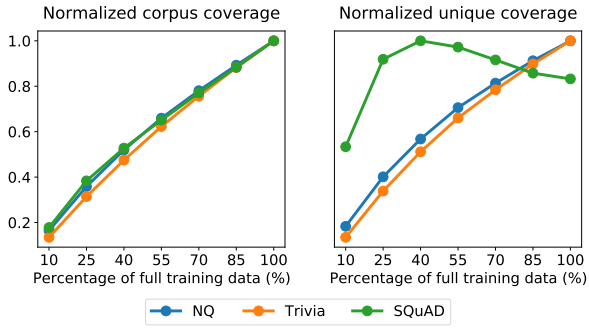


Figure 4: Dataset statistics for different amounts of data. Left: Normalized corpus coverage. Right: Normalized unique passage coverage. The y-axis is shared in each row.

Dataset	Coverage	Overlap	Unique
NQ	30,466	0.21	22,424
Trivia	**42,473**	0.14	**34,910**
SQuAD	3,247	**0.68**	738

Table 3: Corpus coverage and positive passage overlap, as well as the unique passage coverage which equals to corpus coverage $*$ $(1 -$ positive passage overlap$)^{1.3}$ for each dataset.

- **Positive passage overlap**: Ratio between the number of positive passages that can answer more than two training questions and the total number of distinct positive passages.

In this paper, each question only has one positive passage. We further define an intermediate statistics called *unique passage coverage*:

- **Unique passage coverage**: Corpus coverage $\times$ $(1 -$ positive passage overlap$)^{\alpha}$.

where α is an empirical value and is used to adjust the weights between the coverage and overlap.

Despite there being other statistics, we find these statistics above reasonable to reflect the features of each dataset, as well as the correlation with the cross-domain marginalization.

Tbl. 3 shows the corpus coverage and positive passage overlap we define on three QA datasets, where we collect the aforementioned statistics for the training data of each dataset. We can see that despite having the most training data, SQuAD also has the largest positive passage overlap.

Fig. 4 (right column) shows that the unique passage coverage of SQuAD (green line) also behaves similarly as the in-domain marginalization results of SQuAD's passage encoder (Fig. 3, right column),

P-encoder	NQ	Trivia	WQ	Curated	SQuAD	Average
SQuAD-100%	**63.3/77.1**	**73.5/82.4**	65.2/76.7	79.5/90.6	61.1/76.0	68.5/80.5
SQuAD-40%	62.8/76.4	72.8/82.3	**65.9/77.4**	**81.3/91.1**	**62.3/76.8**	**69.2/80.8**

Table 4: Top-20/100 (%) accuracy of the passage encoders trained on SQuAD and 40% of SQuAD, pairing with the question encoder trained on each domain and tested on each domain's test set. With only 40% of data, a better balance between the corpus coverage and positive passage overlap is achieved on SQuAD, and therefore this passage encoder is even better than the one trained with 100% SQuAD data.

which rises as the data amount increases and then drops after 40% of training data. We set $\alpha = 1.3$ for the unique corpus coverage in order to obtain the best curve in Fig. 4. For other α values in $[1, 2]$, the trend is similar but peaks at different percentages of the data.

To further verify the robustness of the passage encoder trained with only 40% training data of SQuAD, we test its passage encoder on five QA test sets and pair it with the in-domain question encoder trained with 100% data. Tbl. 4 shows the comparison between the passage encoders trained with full SQuAD and 40% of SQuAD, respectively. We can see that with only 40% of training data, the passage encoder manages to achieve similar and even higher performance compared with the one trained with full data. Therefore, we have enough reasons to believe that the unique passage coverage, which is related to the corpus coverage and positive passage overlap of the training data, indeed influences the passage encoder strongly.

11 Discussions about Passage Encoder

In the previous sections, we manage to identify the importance of the passage encoder and its affecting factors such as positive passage overlap and corpus coverage of the training data. We find that our discoveries are consistent with some previous work's conclusions. For example, Zhan et al. (2021, 2020a); Sciavolino et al. (2021) all find that it is sufficient to achieve satisfying retrieval performance by just fine-tuning the question encoder with a fixed passage encoder, which demonstrates the importance of a robust passage encoder in domain adaptation and hard-negative mining.

However, how to learn such a robust passage encoder is challenging as pre-training DPR on a single QA dataset will introduce biases. Multi-task dense retrieval (Maillard et al., 2021; Li et al., 2021; Metzler et al., 2021) uses multiple experts learned on different domains to solve this problem. These solutions are effective but not efficient as they build multiple indexes and perform searches for each expert, requiring a lot of resources and storage space.

Another solution is to build a question-agnostic passage encoder so that the model is not biased towards particular QA tasks. Densephrases (Lee et al., 2021a,b) pioneers in this direction by building indexes using phrases instead of chunks of passages for multi-granularity retrieval. By breaking passages into finer units, Densephrases indeed improves the generalization of dense retrieval in different domains with query-side fine-tuning. However, similar to multi-task learning, it is not efficient as the phrase index could be enormous for a corpus like Wikipedia, and techniques such as product quantization (Gray and Neuhoff, 1998) are applied to improve efficiency at the cost of effectiveness.

Overall, it is desirable to have a robust passage encoder for efficient dense retrieval according to previous work and our analysis, but challenges still remain in effectiveness-efficiency trade-off.

12 Conclusions

We propose a encoder attribution analysis of DPR using encoder marginalization to individually evaluate each encoder of DPR. We quantify the contribution of each encoder of DPR by marginalizing over the other random variables under a probabilistic framework. We find that the passage encoder plays a more important role compared to the question encoder in terms of top-k retrieval accuracy. We also perform a case study under the data efficiency setting to demonstrate how to find possible affecting factors in the QA datasets for individual encoders. We identify that passage encoders are affected by positive passage overlap and corpus coverage of the training data, while question encoders are sensitive to the training sample complexity. Our framework is also very general and can be applied to other bi-encoder-based methods for encoder attribution analysis, but one needs to pay attention to the choice of the encoder prior distribution to ensure the marginalization is appropriate.

References

Dilip Arumugam, Peter Henderson, and Pierre-Luc Bacon. 2021. An information-theoretic perspective on credit assignment in reinforcement learning. *arXiv preprint arXiv:2103.06224*.

Petr Baudiš and Jan Šedivỳ. 2015. Modeling of the question answering task in the YodaQA system. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 222–228. Springer.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Murat Binay. 2005. Performance attribution of us institutional investors. *Financial Management*, 34(2):127–152.

Yinqiong Cai, Yixing Fan, Jiafeng Guo, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2021. Semantic models for the first-stage retrieval: A comprehensive review. *arXiv preprint arXiv:2103.04831*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

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. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*.

Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. 2016. *Deep Learning*. Adaptive computation and machine learning. MIT Press.

R.M. Gray and D.L. Neuhoff. 1998. Quantization. *IEEE Transactions on Information Theory*, 44(6):2325–2383.

Kelvin Guu, Kenton Lee, Z. Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, and Rémi Munos. 2019. Hindsight credit assignment. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12467–12476.

Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Sebastian Riedel, and Edouard Grave. 2020. A memory efficient baseline for open domain question answering. *arXiv preprint arXiv:2012.15156*.

Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. 2021. How does BERT rerank passages? an attribution analysis with information bottlenecks. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 496–509, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

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 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)*, pages 6769–6781, Online. 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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Brian Kulis. 2012. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364.

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.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021a. Learning dense representations of phrases at scale. 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 6634–6647, Online. Association for Computational Linguistics.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021b. Phrase retrieval learns passage retrieval, too. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3661–3672, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Patrick S. H. Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1000–1008. Association for Computational Linguistics.

Minghan Li, Ming Li, Kun Xiong, and Jimmy Lin. 2021. Multi-task dense retrieval via model uncertainty fusion for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 274–287, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An easy-to-use Python toolkit to support replicable ir research with sparse and dense representations. *arXiv preprint arXiv:2102.10073*.

Linqing Liu, Patrick S. H. Lewis, Sebastian Riedel, and Pontus Stenetorp. 2021. Challenges in generalization in open domain question answering. *arXiv preprint arXiv:2109.01156*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Shuqi Lu, Chenyan Xiong, Di He, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is more: Pretraining a strong siamese encoder using a weak decoder. *arXiv preprint arXiv:2102.09206*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Trans. Assoc. Comput. Linguistics*, 9:329–345.

Xueguang Ma, Minghan Li, Kai Sun, Ji Xin, and Jimmy Lin. 2021. Simple and effective unsupervised redundancy elimination to compress dense vectors for passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2854–2859, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chris Mack. 2008. Fundamental principles of optical lithography. Appendix C: The Dirac Delta Function:495–500.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oguz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. 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 1098–1111, Online. Association for Computational Linguistics.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Generation-augmented retrieval 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4089–4100. Association for Computational Linguistics.

Donald Metzler, Yi Tay, Dara Bahri, and Marc Najork. 2021. Rethinking search: Making domain experts out of dilettantes. *SIGIR Forum*, 55(1).

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. 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 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Marvin Minsky. 1961. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne,*

Australia, July 15-20, 2018, Volume 1: Long Papers, pages 1896–1906. 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.

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

Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. 2020. Restricting the flow: Information bottlenecks for attribution. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. *arXiv preprint arXiv:2109.08535*.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Richard S. Sutton and Andrew G. Barto. 1998. Reinforcement learning: An introduction. *IEEE Trans. Neural Networks*, 9(5):1054–1054.

Richard Stuart Sutton. 1984. *Temporal credit assignment in reinforcement learning*. Ph.D. thesis, University of Massachusetts Amherst.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. 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 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Bohong Wu, Zhuosheng Zhang, Jinyuan Wang, and Hai Zhao. 2021. Representation decoupling for open-domain passage retrieval. *arXiv preprint arXiv:2110.07524*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing 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 2: Short Papers)*, pages 979–986, Online. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1503–1512. ACM.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020a. Learning to retrieve: How to train a dense retrieval model effectively and efficiently. *arXiv preprint arXiv:2010.10469*.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020b. Repbert: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498*.

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. TYDI: A multi-lingual benchmark for dense retrieval. *arXiv preprint arXiv:2108.08787*.