Exploring Dual Encoder Architectures for Question Answering

Zhe Dong Jianmo Ni Dan Bikel Enrique Alfonseca
Yuan Wang Chen Qu Imed Zitouni
Google Inc
{zhedong, jianmon, dbikel, ealfonseca, yuawang, cqu, izitouni}@google.com

Abstract

Dual encoders have been used for question-answering (QA) and information retrieval (IR) tasks with good results. There are two major types of dual encoders, Siamese Dual Encoders (SDE), with parameters shared across two encoders, and Asymmetric Dual Encoder (ADE), with two distinctly parameterized encoders. In this work, we explore the dual encoder architectures for QA retrieval tasks. By evaluating on MS MARCO and the MultiReQA benchmark, we show that SDE performs significantly better than ADE. We further propose three different improved versions of ADEs. Based on the evaluation of QA retrieval tasks and direct analysis of the embeddings, we demonstrate that sharing parameters in projection layers would enable ADEs to perform competitively with SDEs.

1 Introduction

A dual encoder is an architecture consisting of two encoders, each of which encodes an input (such as a piece of text) into an embedding, and where the model is optimized based on similarity metrics in the embedding space. It has shown excellent performance in a wide range of information retrieval and question answering tasks (Gillick et al., 2018; Karpukhin et al., 2020). This approach is also easy to productionize because the embedding index of dual encoders can grow dynamically for newly discovered or updated documents and passages without retraining the encoders (Gillick et al., 2018). In contrast, generative neural networks used for question answering need to be retrained with new data. This advantage makes dual encoders more robust to freshness.

There are different valid designs for dual encoders. As shown in Table 1, the two major types are: Siamese Dual Encoder (SDE) and Asymmetric Dual Encoder (ADE). In a SDE the parameters are shared between the two encoders. In an ADE only some or no parameters are shared (Gillick et al., 2018). In practice, we often require certain asymmetry in the dual encoders, especially in the case where the inputs of the two towers are of different types. Though all of these models have achieved excellent results in different NLP applications, how these parameter-sharing design choices affect the model performance is largely unexplored.

Model	Architecture
DPR (Karpukhin et al., 2020)	Asymmetric
DensePhrases (Lee et al., 2021a)	Asymmetric
SBERT (Reimers and Gurevych, 2019)	Siamese
ST5 (Ni et al., 2021b)	Siamese

Table 1: Existing off-the-shelf dual encoders.

This paper explores the impact of parameter sharing in *different components* of dual encoders on question answering tasks, and whether the impact holds for dual encoders with different capacity. We conduct experiments across six well-established datasets and find that SDEs consistently outperforms ADEs on question answering tasks. We further propose to improve ADEs by sharing the projection layer between the two encoders, and show that this simple approach enables ADEs to achieve comparable or even better performance than SDEs.

2 Related work

Dual encoders have been widely studied in entity linking (Gillick et al., 2018), open-domain question answering (Karpukhin et al., 2020), and dense retrieval (Ni et al., 2021a), etc. This architecture consists of two encoders, where each encoder encodes arbitrary inputs that may differ in type or granularity, such as queries, images, answers, passages, or documents.

Open-domain question answering (ODQA) is a challenging task that searches for evidence across large-scale corpora and provides answers to user queries (Voorhees, 1999; Chen et al., 2017). One of the prevalent paradigms for ODQA is a two-step

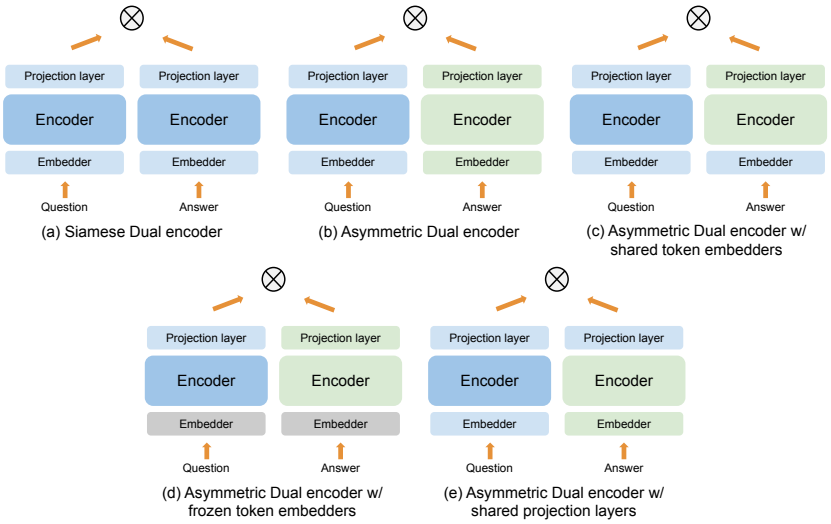

Figure 1: Architectures of dual encoders. We study whether parameter sharing in different dual encoder components (i.e. token embedder, transformer encoder, and projection layer) can lead to better representation learning. Different color within each figure represents distinctly parameterized components, and grey components are frozen during the fine-tuning.

approach, consisting of a retriever to find relevant evidence and a reader to synthesize answers. Alternative approaches are directly retrieving from large candidate corpus to provide sentence-level (Guo et al., 2021) or phrase-level (Lee et al., 2021b) answers; or directly generating answers or passage indices using an end-to-end generation approach (Tay et al., 2022). Lee et al. (2021a) compared the performance of SDEs and ADEs for phrase-level QA retrieval tasks. However, they only considered the two extreme cases that two towers have the parameters completely shared or distinct. In this work, we address the missing piece of previous work by exploring parameter sharing in different parts of the model.

3 Method

We follow a standard setup of QA retrieval: given a question q and a corpus of answer candidates $\mathcal{A}$, the goal is to retrieve k relevant answers $\mathcal{A}_k \in \mathcal{A}$ for q. The answer can be either a passage, a sentence, or a phrase.

We adopt a *dual encoder* architecture (Gillick et al., 2018; Reimers and Gurevych, 2019) as the model to match query and answers for retrieval. The model has two encoders, where each is a transformer that encodes a question or an answer. Each encoder first produces a fixed-length representation for its input and then applies a projection layer to generate the final embedding.

We train the dual encoder model by optimizing the contrastive loss with an in-batch sampled softmax (Henderson et al., 2017):

$$\mathcal{L} = \frac{e^{\text{sim}(q_i, a_i)/\tau}}{\sum_{j \in \mathcal{B}} e^{\text{sim}(q_i, a_j)/\tau}}, \quad (1)$$

where q_i is a question and a_* is a candidate answer. a_i is ground-truth answer, or a positive sample, for q_i. All other answers a_j in the same batch $\mathcal{B}$ are considered as negative samples during training. τ is the softmax temperature and $\mathbf{sim}$ is a similarity function to measure the relevance between the question and the answer. In this work, we use cosine distance as the similarity function.

We want to explore the architectures of dual encoder with different degrees of parameter sharing. In particular, we aim to evaluate the importance of parameter sharing in dual encoder training. To this end, we study five different variants of dual encoders as shown in Figure 1:

- Siamese Dual-Encoder (SDE),
- Asymmetric Dual-Encoder (ADE),
- ADE with shared token embedder (ADE-STE),
- ADE with frozen token embedder (ADE-FTE),
- ADE with shared projection layer (ADE-SPL).

4 Experiments and Analysis

We evaluate the proposed dual encoder architectures on six question-answering retrieval tasks from MS MARCO (Bajaj et al., 2016) and MultiReQA (Guo et al., 2021). In MS MARCO, we consider the relevant passages as answer candidates, while for the five QA datasets in MultiReQA the answer candidates are individual sentences.

Metric	Model	MSMARCO	NQ	SQuAD	TriviaQA	HotpotQA	SearchQA
P@1	SDE	**15.92**	48.87	**70.13**	36.55	**34.36**	36.40
	ADE	14.20	47.83	60.39	31.30	26.71	39.48
	ADE-STE	14.71	48.29	61.05	33.59	28.71	40.43
	ADE-FTE	14.23	49.38	62.86	35.11	29.07	**42.06**
	ADE-SPL	15.46	**50.06**	69.39	**38.17**	33.66	41.13
	BERT-DE	-	36.22	55.13	29.11	32.05	30.2
	USE-QA	-	38	66.83	32.58	31.71	31.45
MRR	SDE	**28.49**	61.15	**78.44**	49.29	45.58	54.26
	ADE	26.31	59.38	70.33	43.42	37.27	55.02
	ADE-STE	26.78	59.81	70.85	45.79	39.14	56.08
	ADE-FTE	26.64	61.23	72.18	46.95	39.72	57.44
	ADE-SPL	28.20	**61.92**	77.65	**50.3**	44.19	**57.48**
	BERT-DE	-	52.02	64.74	41.34	**46.21**	47.08
	USE-QA	-	52.27	75.86	42.39	43.77	50.7

Table 2: Precision at 1(P@1)(%) and Mean Reciprocal Rank (MRR)(%) on QA retrieval tasks. `SDE` and `ADE` stand for Siamese Dual-Encoder and Asymmetric Dual-Encoder, respectively. `ADE-STE`, `-FTE` and `-SPL` are the ADEs with shared token-embedders, frozen token-embedders, and shared projection-layers, respectively. `BERT-DE`, which stands BERT (Devlin et al., 2019) Dual-Encoder, and `USE-QA` (Yang et al., 2020) are the baselines reported in MultiReQA (Guo et al., 2021). The most performant models are marked in bold.

Metric	Model	Model Size		
		`small`	`base`	`large`
P@1	SDE	**14.50**	**15.92**	**16.53**
	ADE	13.31	14.20	14.17
	ADE-STE	12.99	14.71	15.14
	ADE-FTE	13.67	14.23	14.73
	ADE-SPL	14.31	15.46	16.42
MRR	SDE	**25.76**	**28.49**	29.63
	ADE	23.89	26.31	26.99
	ADE-STE	23.78	26.78	27.61
	ADE-FTE	24.51	26.64	27.40
	ADE-SPL	25.53	28.20	**29.70**

Table 3: Scaling effect. Precision at 1 (P@1)(%) and Mean Reciprocal Rank (MRR)(%) on MS MARCO (Bajaj et al., 2016) QA retrieval tasks, with dual encoders initialized from `t5.1.1-small`, `-base`, and `-large` checkpoints. The most performant models are marked in bold.

To initialize the parameters of dual encoders, we use pre-trained `t5.1.1` encoders (Raffel et al., 2020). Following Ni et al. (2021b), we take the average embeddings of the T5 encoder's outputs and send to a projection layer to get the final embeddings. The projection layers are randomly initialized, with variance scaling initialization with scale 1.0. For the retrieval, we use mean embeddings from the encoder towers. To make a fair comparison, the same hyper-parameters are applied across all the models for the fine-tuning with Adafactor optimizer (Shazeer and Stern, 2018), using learning rate 10^{-3} and batch size 512. The models are fine-tuned for $20,000$ steps, with linear decay of learning rate from 10^{-3} to 0 at the final steps. The fine-tuned models are benchmarked with precision at 1 (P@1) and mean reciprocal rank (MRR) on the QA retrieval tasks, in Table 2.

4.1 Comparing SDE and ADE

SDE and ADE in Figure 1 (a) and (b) are the two most distinct dual-encoders in terms of parameter sharing. Experiment results show that, on QA retrieval tasks, ADE performs consistently worse than SDE. To explain that, our **assumption** is that, at inference time, the two distinct encoders in ADE that do not share any parameters, map the questions and the answers into two parameter spaces that are not perfectly aligned. However, for SDE, parameter sharing enforces the embeddings from the two encoders to be in the same space. We verify this assumption in Section 4.3.

4.2 Improving the Asymmetric Dual Encoder

Although the dual encoders with maximal parameter sharing (SDEs) performs significantly better than the ones without parameter sharing (ADEs), we often require certain asymmetry in the dual encoders in practice. Therefore, trying to improve the performance of ADEs, we construct dual-encoders with parameters shared at different levels between the two encoders.

Shared and Frozen Token Embedders. Token embedders are the lowest layers close to the input text. In ADEs, token embedders are initialized from the same set of pre-trained parameters, but fine-tuned separately. A straightforward way to bring ADEs closer to SDEs is to share the token embedders between the two towers, or to an extreme, to simply freeze the token embedders during training.

Evaluated on MS MARCO and MultiReQA, the results in Table 2 show that both freezing (ADE-FTE) and sharing (ADE-STE) token embedders bring consistent, albeit marginal, improvements for ADEs. However, ADEs with common token embedders still leave a significant gap compared to SDEs on most tasks. These results suggest token embedders might not be the key to close this gap.

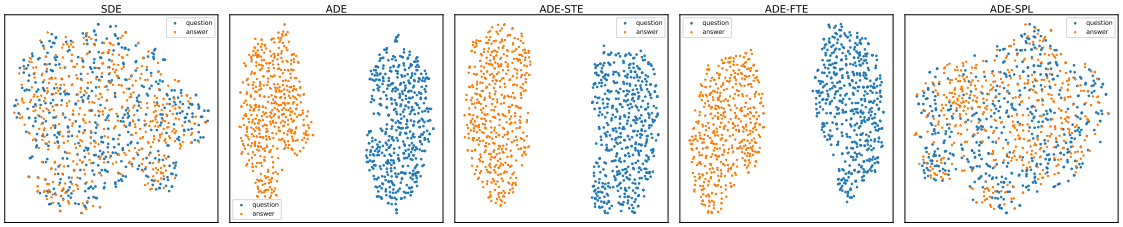

Figure 2: t-SNE clustering of the embeddings of the NaturalQuestions eval set generated by five dual encoders. The blue and orange points represent the embeddings of the questions and answers, respectively.

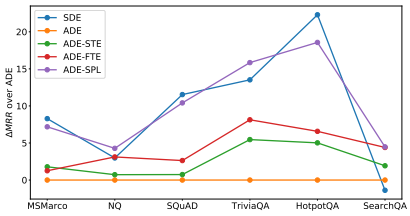

Figure 3: Relative performance improvements of different models relative to ADE on QA retrieval tasks. $\Delta\text{MRR} = (\text{MRR} - \text{MRR}_{ADE})/\text{MRR}_{ADE}) \times 100$.

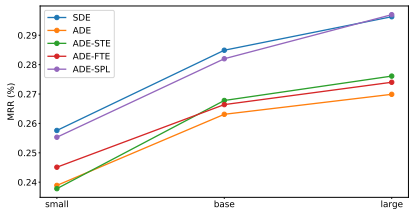

Figure 4: The impact of model size on the performance of different dual encoder architectures, measured by MRR on the eval set of MS MARCO.

Shared Projection Layers. Another way of improving retrieval quality of ADEs is to share the projection layers between the two encoders. Table 2 shows that sharing projection layers drastically improves the quality of ADEs. As in Figure 3, ADE-SPL (purple curve) performs on-par and, sometimes, even better than SDE (blue curve). This observation reveals that sharing projection layers is a valid approach to enhance the performance of ADEs. This technique would be vital if asymmetry is required by a modeling task. We further interpret this result in the next section.

4.3 Analysis on the Embeddings

The experiments corroborate our assumption that sharing the projection layer enforces the two encoders to produce embeddings in the same parameter space, which improves the retrieval quality.

To further substantiate our assumption, we first generate the question and answer embeddings from the NaturalQuestions eval set, and then use t-SNE (van der Maaten and Hinton, 2008) to project and cluster the embeddings into 2-dimensional space.[1] Figure 2 shows that, for ADE, ADE-STE and ADE-FTE that have separate projection layers, the question and answer embeddings are projected and clustered into two disjoint groups. In comparison, ADE-SPL that shares the projection layers, the embeddings of questions and answers are not separable by t-SNE, which is similar to the behavior of SDE. This verifies our assumption that the projection layer plays an important role in bringing together the representations of questions and answers, and is the key for retrieval performance.

4.4 Impact of Model Size.

To assess the impact of model size, we fine-tune and evaluate the dual-encoders initialized from `t5.1.1-small` ($\sim$ 77M parameters), `-base` ($\sim$ 250M), and `-large` ($\sim$ 800M) on the MS MARCO. Table 3 and Figure 4 show that, across different model sizes, sharing projection layers consistently improves the retrieval performance of ADE, and ADE-SPL performs competitively with SDE. This observation further validates our recommendation to share the projection layer in ADEs.

5 Conclusion and Future Work

Based on the experiments on six QA retrieval tasks with three different model sizes, we conclude that, although SDEs outperforms ADEs, sharing the projection layer between the two encoders enables ADEs to perform competitively with SDEs. By directly probing the embedding space, we demonstrate that the shared projection layers in SDE and ADE-SPL can map the embeddings of the two encoders into coinciding parameter spaces, which is crucial for improving the retrieval quality. Therefore, we recommend to share the projection layers between two encoders of ADEs in practice.

[1] For efficiently clustering with t-SNE, we randomly sampled questions and answers, 400 each, from the NQ eval set.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *ACL*.

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.

D. Gillick, A. Presta, and Gaurav Singh Tomar. 2018. End-to-end retrieval in continuous space. *ArXiv*, abs/1811.08008.

Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. 2021. MultiReQA: A cross-domain evaluation forRetrieval question answering models. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 94–104, Kyiv, Ukraine. Association for Computational Linguistics.

Matthew Henderson, Rami Al-Rfou, B. Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and R. Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *ArXiv*, abs/1705.00652.

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.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021a. Learning dense representations of phrases at scale. In *ACL/IJCNLP*.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021b. Learning dense representations of phrases at scale. In *Association for Computational Linguistics (ACL)*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hern'andez 'Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2021a. Large dual encoders are generalizable retrievers. *ArXiv*, abs/2112.07899.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021b. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *CoRR*, abs/2108.08877.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP/IJCNLP*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Yi Tay, Vinh Quang Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W. Cohen, and Donald Metzler. 2022. Transformer memory as a differentiable search index. *ArXiv*, abs/2202.06991.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Ellen M. Voorhees. 1999. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.