Task-Aware Specialization for Efficient and Robust Dense Retrieval for Open-Domain Question Answering

Hao Cheng[♠] **Hao Fang**[♣] **Xiaodong Liu**[♠] **Jianfeng Gao**[♠]
[♠] Microsoft Research [♣] Microsoft Semantic Machines
{chehao,hafang,xiaodl,jfgao}@microsoft.com

Abstract

Given its effectiveness on knowledge-intensive natural language processing tasks, dense retrieval models have become increasingly popular. Specifically, the *de-facto* architecture for open-domain question answering uses two isomorphic encoders that are initialized from the same pretrained model but separately parameterized for questions and passages. This bi-encoder architecture is parameter-inefficient in that there is no parameter sharing between encoders. Further, recent studies show that such dense retrievers underperform BM25 in various settings. We thus propose a new architecture, **T**ask-**A**ware **S**pecialization for d**E**nse **R**etrieval (TASER), which enables parameter sharing by interleaving shared and specialized blocks in a single encoder. Our experiments on five question answering datasets show that TASER can achieve superior accuracy, surpassing BM25, while using about 60% of the parameters as bi-encoder dense retrievers. In out-of-domain evaluations, TASER is also empirically more robust than bi-encoder dense retrievers. Our code is available at https://github.com/microsoft/taser.

1 Introduction

Empowered by learnable neural representations built upon pretrained language models, the dense retrieval framework has become increasingly popular for fetching external knowledge in various natural language processing tasks (Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020). For open-domain question answering (ODQA), the *de-facto* dense retriever is the bi-encoder architecture (Lee et al., 2019; Karpukhin et al., 2020), consisting of a question encoder and a passage encoder. Typically, the two encoders are isomorphic but separately parameterized, as they are initialized from the same pretrained model and then fine-tuned on the task.

Despite of its popularity, this bi-encoder architecture with fully decoupled parameterization has some open issues. First, from the efficiency perspective, the bi-encoder parameterization apparently results in scaling bottleneck for both training and inference. Second, empirical results from recent studies show that such bi-encoder dense retrievers underperform its sparse counterpart BM25 (Robertson and Walker, 1994) in various settings. For example, both Lee et al. (2019) and Karpukhin et al. (2020) suggest the inferior performance on SQuAD (Rajpurkar et al., 2016) is partially due to the high lexical overlap between questions and passages, which gives BM25 a clear advantage. Sciavolino et al. (2021) also find that bi-encoder dense retrievers are more sensitive to distribution shift than BM25, resulting in poor generalization on questions with rare entities.

In this paper, we develop **T**ask-**A**ware **S**pecialization for d**E**nse **R**etrieval, TASER, as a more parameter-efficient and robust architecture. Instead of using two isomorphic and fully decoupled Transformer (Vaswani et al., 2017) encoders, TASER interleaves shared encoder blocks with specialized ones in a single encoder, motivated by recent success in using Mixture-of-Experts (MoE) to scale up Transformer (Fedus et al., 2021). For the shared encoder block, the entire network is used to encode both questions and passages. For the specialized encoder block, some sub-networks are task-specific and activated only for certain encoding tasks. To choose among task-specific sub-networks, TASER uses an input-dependent *routing mechanism*, *i.e.,* questions and passages are passed through separate dedicated sub-networks.

We carry out both in-domain and out-of-domain evaluation for TASER. For the in-domain evaluation, we use five popular ODQA datasets. Our best model outperforms BM25 and existing bi-encoder dense retrievers, while using much less parameters. It is worth noting that TASER can effectively close the performance gap on SQuAD between dense retrievers and BM25. One interest-

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1864–1875

ing finding from our experiments is that excluding SQuAD from the multi-set training is unnecessary, which was a suggestion made by Karpukhin et al. (2020) and adopted by most follow-up work. Our out-of-domain evaluation experiments use EntityQuestions (Sciavolino et al., 2021) and BEIR (Thakur et al., 2021). Consistent improvements over the doubly parameterized bi-encoder dense retriever are observed in these zero-shot evaluations as well. Our code is available at https://github.com/microsoft/taser.

2 Background

In this section, we provide necessary background about the bi-encoder architecture for dense passage retrieval which is widely used in ODQA (Lee et al., 2019; Karpukhin et al., 2020) and is the primary baseline model in our experiments.

The bi-encoder architecture consists of a question encoder and a passage encoder, both of which are usually Transformer encoders (Vaswani et al., 2017). A Transformer encoder is built up with a stack of Transformer blocks. Each block consists of a multi-head self-attention (MHA) sub-layer and a feed-forward network (FFN) sub-layer, with residual connections (He et al., 2016) and layer-normalization (Ba et al., 2016) applied to both sub-layers. Given an input vector $\mathbf{h} \in \mathbb{R}^d$, the FFN sub-layer produces an output vector as following

$$\texttt{FFN}(\mathbf{h}) = \mathbf{W}_2 \max\{0, \mathbf{W}_1\mathbf{h} + \mathbf{b}_1\} + \mathbf{b}_2, \quad (1)$$

where $\mathbf{W}_1 \in \mathbb{R}^{m\times d}$, $\mathbf{W}_2 \in \mathbb{R}^{d\times m}$, $\mathbf{b}_1 \in \mathbb{R}^m$, and $\mathbf{b}_2 \in \mathbb{R}^d$ are learnable parameters. For a sequence of N tokens, each Transformer block produces N corresponding vectors, together with a vector for the special prefix token `[CLS]` which can be used as the representation of the sequence. We refer readers to (Vaswani et al., 2017) for other details about Transformer. Typically the question encoder and passage encoder are initialized from a pretrained language model such as BERT (Devlin et al., 2019), but they are parameterized separately, *i.e.,* their parameters would differ after training.

The bi-encoder model independently encodes questions and passages into d-dimension vectors, using the final output vectors for `[CLS]` from the corresponding encoders, denoted as $\mathbf{q} \in \mathbb{R}^d$ and $\mathbf{p} \in \mathbb{R}^d$, respectively. The relevance between a question and a passage can then be measured in the vector space using dot product, *i.e.,* $\text{sim}(\mathbf{q}, \mathbf{p}) = \mathbf{q}^T\mathbf{p}$. During training, the model is optimized based on a contrastive learning objective,

$$L_{sim} = -\frac{\exp(\text{sim}(\mathbf{q}, \mathbf{p}^+))}{\sum_{\mathbf{p}'\in\mathcal{P}\cup\{\mathbf{p}^+\}} \exp(\text{sim}(\mathbf{q}, \mathbf{p}'))}, \quad (2)$$

where $\mathbf{p}^+$ is the relevant (positive) passage for the given question, and $\mathcal{P}$ is the set of irrelevant (negative) passages. During inference, all passages are pre-converted into vectors using the passage encoder. Then, each incoming question is encoded using the question encoder, and a top-K list of most relevant passages are retrieved based on their relevance scores with respect to the question.

Although the bi-encoder dense retrieval architecture has achieved impressive results in ODQA, few work has attempted to improve its parameter efficiency. Further, compared to the spare vector space model BM25 (Robertson and Walker, 1994), such bi-encoder dense retrievers sometimes suffer from inferior generalization performance, *e.g.,* when the training data is extremely biased (Lebret et al., 2016; Karpukhin et al., 2020) or when there is a distribution shift (Sciavolino et al., 2021). In this paper, we conjecture that the unstable generalization performance is partially related to the unnecessary number of learnable parameters in the model. Therefore, we develop a task-aware specialization architecture for dense retrieval with parameter sharing between the question and passage encoders, which turns out to improve both parameter efficiency and generalization performance.

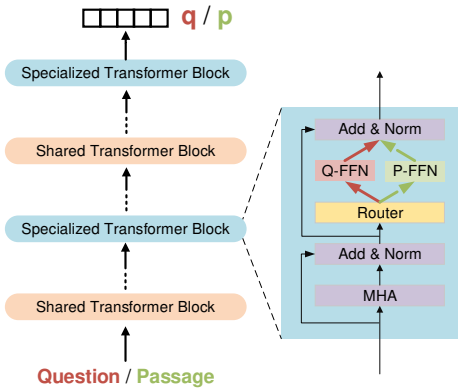

Figure 1: The architecture overview of TASER. The specialized transformer block consists of a Q-FFN for questions, a P-FFN for passages, and a router which deterministically chooses between these two expert FFN sub-layers based on input.

3 Proposed Model: TASER

As shown in Fig. 1, TASER interleaves shared Transformer blocks with specialized ones. The

shared Transformer block is identical to the Transformer block used in the bi-encoder architecture, but the entire block is shared for both questions and passages. In the specialized block, we apply MoE-style task-aware specialization to the FFN sub-layer, following (Fedus et al., 2021), where the router always routes the input to a single expert FFN sub-layer. In our experiments, we use a simple yet effective routing mechanism, which uses an expert sub-layer (Q-FFN) for questions and another (P-FFN) for passages. The router determines the expert FFN sub-layer based on whether the input is a question or a passage. Other routing mechanisms are discussed in Appendix A.

TASER uses one specialized Transformer block after every T shared Transformer blocks in the stack, starting with a shared one at the bottom. Our preliminary study indicates that the model performance is not sensitive to the choice of T, so we use $T = 2$ for experiments in this paper.

Similar to the bi-encoder architecture, TASER is trained using the contrastive learning objective L_{sim} defined in Equation 2. Specifically, the objective needs to use a set of negative passages $\mathcal{P}$ for each question. Following Xiong et al. (2020) and Qu et al. (2021), we construct $\mathcal{P}$ via hard negatives mining (Appendix B). Our experiments use the *multi-set* training paradigm, *i.e.,* the model is trained by combining data from multiple datasets to obtain a model that works well across the board.

4 Experiments

4.1 In-Domain Evaluation

We carry out in-domain evaluations on five ODQA datasets: NaturalQuestions (NQ; Kwiatkowski et al., 2019a), TriviaQA (TQ; Joshi et al., 2017), WebQuestions (WQ; Berant et al., 2013), CuratedTrec (CT; Baudiš and Šedivý, 2015), and SQuAD (Rajpurkar et al., 2016). All data splits and the Wikipedia collection for retrieval used in our experiments are the same as Karpukhin et al. (2020). The top-K retrieval accuracy (R@K) is used for evaluation, which evaluates whether any gold answer string is contained in the top K retrieved passages.

Besides BERT-base, coCondenser-Wiki (Gao and Callan, 2022) is also used to initialize TASER models. We further present results of hybrid models that linearly combine the dense retrieval scores with the BM25 scores. See Appendix D for details. Evaluation results are summarized in Table 1.[1] Note that the last five models Table 1 are trained with the same hard negatives mining.

	NQ	TQ	WQ	CT	SQuAD
BM25$^{(1)}$	62.9	76.4	62.4	80.7	71.1
Multi-Set Training (without SQuAD)					
DPR$^{(1)}$	79.5	78.9	75.0	88.8	52.0
DPR$_{\text{BM25}}{}^{(1)}$	82.6	82.6	77.3	90.1	75.1
xMoCo$^{(2)}$	82.5	80.1	78.2	89.4	55.9
SPAR$_{\text{Wiki}}{}^{(3)}$	83.0	82.6	76.0	89.9	73.0
SPAR$_{\text{PAQ}}{}^{(4)}$	82.7	82.5	76.3	90.3	72.9
Multi-Set Training (with SQuAD)					
DPR†	80.9	79.6	74.0	88.0	63.1
DPR$^{\diamond}$	82.5	81.8	77.8	91.2	67.0
DPR*	83.7	82.6	78.9	91.6	68.0
TASER$^{\diamond}$	83.6	82.0	77.9	91.1	69.7
TASER*	84.9	83.4	78.9	90.8	72.9
TASER$^{\star}_{\text{BM25}}$	**85.0**	**84.0**	**79.6**	**92.1**	**78.0**

Table 1: In-domain evaluation results (test set R@20). $^{(1)}$: (Ma et al., 2021). $^{(2)}$: (Karpukhin et al., 2020). $^{(3)}$: (Yang et al., 2021). $^{(4)}$: (Chen et al., 2022). BM25: combined with BM25 scores. †: initialized from BERT-base and without hard negatives mining. $^{\diamond}$: initialized from BERT-base. *: initialized from coCondenser-Wiki. The last five models are trained with the same hard negatives mining.

All prior work excludes SQuAD from the multi-set training, as suggested by Karpukhin et al. (2020). We instead train models using all five datasets. Specifically, we observe that this would not hurt the overall performance, and it actually significantly improves the performance on SQuAD, comparing DPR$^{(1)}$ with DPR†.

Comparing models initialized from BERT-base, TASER$^{\diamond}$ significantly outperforms xMoCo (Yang et al., 2018) and is slightly better than DPR$^{\diamond}$, using around 60% parameters. SPAR (Chen et al., 2022) is also initialized from BERT-base, but it augments DPR with another dense lexical model trained on either Wikipedia or PAQ (Lewis et al., 2021), which doubles the model sizes (Table A3). TASER$^{\diamond}$ is mostly on par with SPAR-Wiki and SPAR-PAQ, except on SQuAD, but its model size is about a quarter of SPAR.

Gao and Callan (2022) has shown the coCondenser model outperforms DPR models initialized from BERT-base in the single-set training setting. Here, we show that using coCondenser-Wiki for initialization is also beneficial for TASER under the multi-set setting, especially for SQuAD where

[1] We also report R@100 scores in Table A2 and corresponding model sizes in Table A3.

	R@20	nDCG@10			
	EQ	AA	DBP	FEV	HQA
BM25	71.2	31.5	31.3	75.3	60.3
$\text{DPR}_{\text{Multi}}$	56.7	17.5	26.3	56.2	39.1
TASER$^{\diamond}$	64.7	32.8	31.4	59.6	50.7
TASER*	66.7	30.5	31.6	58.8	54.5

Table 2: Out-of-domain evaluation results on EntityQuestions (R@20) and four BEIR datasets (nDCG@10). BM25 and $\text{DPR}_{\text{Multi}}$ results are from (Sciavolino et al., 2021) and (Thakur et al., 2021).

R@20 is improved by 3.2 points. Notably, SQuAD is the only dataset among the five where DPR underperforms BM25, due to its higher lexical overlap between questions and passages. Nevertheless, TASER* surpasses BM25 on all five datasets, and they are either on-par or better than state-of-the-art dense-only retriever models, demonstrating its superior parameter efficiency.

Consistent with previous work, combining BM25 with dense models can further boost the performance, particularly on SQuAD. However, the improvement is more pronounced on DPR as compared to TASER*, indicating that TASER* is able to capture more lexical overlap features. Finally, $\text{TASER}^{\star}_{\text{BM25}}$ sets new state-of-the-art performance on all five ODQA datasets.

We also compare the computation time needed for one epoch of training and validation. The baseline DRP model takes approximately 15 minutes, whereas TASER takes 11 minutes (26% improvement), both measured using 16 V100-32G GPUs.

4.2 Out-of-Domain Evaluation

We use two benchmarks to evaluate the out-of-domain generalization ability of TASER$^{\diamond}$ and TASER* from Table 1 . EntityQuestions (EQ; Sciavolino et al., 2021) is used to measure the model sensitivity to entity distributions, as DPR is found to perform poorly on entity-centric questions containing rare entities. BEIR (Thakur et al., 2021) is used to study the model generalization ability in other genres of information retrieval tasks. Specifically, we focus on four datasets from BEIR where DPR underperforms BM25, *i.e.,* ArguAna (AA; Wachsmuth et al., 2018), DBPedia (DBP; Hasibi et al., 2017), FEVER (FEV; Thorne et al., 2018), and HotpotQA (HQA; Yang et al., 2018). Results are summarized in Table 2. For EntityQuestions, we report R@20 scores following Sciavolino et al. (2021).[2] For BEIR datasets, nDCG@10 scores are used following Thakur et al. (2021).

On EntityQuestions, both TASER$^{\diamond}$ and TASER* outperform the doubly parameterized $\text{DPR}_{\text{Multi}}$ (Karpukhin et al., 2020), with TASER* being slightly better. Similar to the in-domain evaluation results, TASER can effectively reduce the performance gap between the dense retrievers and BM25. These results further support our hypothesis that more parameter sharing can improve the model robustness for dense retrievers.

On BEIR datasets, we also observe that TASER models consistently improve over $\text{DPR}_{\text{Multi}}$ across the board. Notably, TASER$^{\diamond}$ and TASER* can actually match the performance of BM25 on ArguAna and DBpedia. Interestingly, coCondenser pre-training has mixed results here, *i.e.,* TASER* is only better than TASER$^{\diamond}$ on HotpotQA and on par or worse on other datasets.

5 Related Work

Recent seminal work on dense retrieval demonstrates its effectiveness using Transformer-based bi-encoder models by either continual pre-training with an inverse cloze task (Lee et al., 2019) or careful fine-tuning (Karpukhin et al., 2020). One line of follow-up work improves dense retrieval models via various continual pre-training approaches (Guu et al., 2020; Chang et al., 2020; Izacard et al., 2021; Gao and Callan, 2022; Oğuz et al., 2021). Better contrastive learning objectives are also introduced (Xiong et al., 2020; Qu et al., 2021; Yang et al., 2021). Motivated by the success of augmenting dense models with sparse models, Chen et al. (2022) combine the dense retriever with a dense lexical model that mimics sparse retrievers. All above work focus on improving the accuracy of bi-encoder dense retrievers, whereas our work tackles the parameter efficiency issue.

Unlike most bi-encoder dense retrievers which measure the similarity between a question and a passage using their corresponding [CLS] vectors, ColBERT (Khattab and Zaharia, 2020) develops a late-interaction paradigm and measures the similarity via a MaxSim operator that computes the maximum similarity between a token in a sequence and all tokens in the other sequence. Such architecture has shown promising results in ODQA (Khattab et al., 2021) and the BEIR benchmark (Santhanam

[2]The R@20 scores are averaged over all relations. More evaluation metrics are reported in Table A4.

et al., 2022). Our work instead focus on the improvement on the underlying text encoders, and the MaxSim operator introduced by ColBERT can be applied on top of TASER.

Xiong et al. (2021) use the BERT-Siamese architecture for dense retrieval, where all Transformer blocks are shared. Compared with this architecture, TASER is a more effective and general way to increase the parameter efficiency, by interleaving specialized Transformer blocks with shared ones.

6 Conclusion

We propose a new parameterization framework, TASER, for improving the efficiency and robustness of dense retrieval for ODQA. It interleaves shared encoder blocks with specialized ones in a single encoder where some sub-networks are task-specific. As the specialized sub-networks are sparsely activated, TASER can provide better parameter efficiency with almost no additional computation cost. Experiments show that TASER substantially outperforms existing fully supervised bi-encoder dense retrievers on both in-domain and out-of-domain generalization.

7 Limitations

In this section, we point out several limitations in this work.

First, our in-domain evaluation experiments focus on passage retrieval for ODQA. While the dense retriever is mostly successful in ODQA, it can be also used in other types of retrieval tasks which may have different input and output format. For example, the KILT benchmark (Petroni et al., 2021) provides several knowledge-intensive tasks other than ODQA. The performance of TASER models trained on such retrieval tasks remain unknown.

Second, compared with traditional sparse vector models like TF-IDF and BM25, the cost of training is an inherent issue of dense retrievers. Although TASER significantly reduce the number of model parameters, the training cost is still high.

Third, in our experiments, we show that the learned routing does not outperform the deterministic routing. This may suggest a better architecture and/or training algorithms for learned routing is needed to fully unleash the power of MoE.

Last, as observed in §4.2, there is still a gap between TASER and BM25 in out-of-domain evaluation. Therefore, how to close this gap will remain a critical topic for future work on dense retrievers.

References

Jimmy Lei Ba, Jami Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv:1607.06459 [stat.ML]*.

Petr Baudiš and Jan Šedivý. 2015. Modeling of the question answering task in the yodaqa system. In *Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction - Volume 9283*, CLEF'15, page 222–228, Berlin, Heidelberg. Springer-Verlag.

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.

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 *Proc. International Conference on Learning Representations (ICLR)*.

Xilun Chen, Kushal Lakhotia, Barlas Oğuz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2022. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? *arXiv:2110.06918 [cs.CL]*.

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.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv:2101.03961 [cs.LG]*.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1265–1268, Shinjuku, Tokyo, Japan.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv: 2112.09118 [cs.IR]*.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv:1611.01144 [stat.ML]*.

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, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

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 '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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. 2019a. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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. 2019b. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213, Austin, Texas. 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. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Xuegang Ma, Kai Sun, Ronak Pradeep, and Jimmy Lin. 2021. A replication study of dense passage retriever. *arXiv:2104.05740 [cs.CL]*.

Barlas Oğuz, Kushal Lakhotia, Anchit Gupta, Patrick Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, and Yashar Mehdad. 2021. Domain-matched pre-training tasks for dense retrieval. *arXiv:2107.13602 [cs.CL]*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online.

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.

Stephen E. Robertson and Stephen Walker. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proc. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 232–241, Dublin, Ireland.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia.

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.

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 *Proc. International Conference on Learning Representations (ICLR)*.

Nan Yang, Furu Wei, Binxing Jiao, Daxing Jiang, and Linjun Yang. 2021. xMoCo: Cross momentum contrastive learning 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 6120–6129, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium.

A More Routing Mechanisms

In the paper, only input-dependent routing is considered. Here, we provide a more comprehensive study of routing mechanisms. In particular, we introduce three routing mechanisms: the deterministic routing (`Det-R`) which is used in our main experiments, the sequence-based routing (`Seq-R`), and the token-based routing (`Tok-R`). Both `Seq-R` and `Tok-R` are learned jointly with the task-specific objective.

Specifically, `Det-R` is the input-dependent routing studied in the main paper where two expert FFN sub-layers are needed for ODQA retrieval, one for questions and one for passages. In this case, the router determines the expert FFN sub-layer based on whether the input is a question or a passage.

For `Seq-R` and `Tok-R`, the router uses a parameterized routing function

$$\texttt{R}(\mathbf{u}) = \texttt{GumbelSoftmax}(\mathbf{A}\mathbf{u} + \mathbf{c}), \quad (3)$$

where `GumbelSoftmax` (Jang et al., 2016) outputs a I-dimensional one-hot vector based on the linear projection parameterized by $\mathbf{A} \in \mathbb{R}^{d \times I}$ and $\mathbf{c} \in \mathbb{R}^I$, I is the number of expert FFN sub-layers in the specialized Transformer block, and $\mathbf{u} \in \mathbb{R}^d$ is the input of the routing function. Here, the routing function is jointly learned with all other parameters using the discrete reparameterization trick. For `Seq-R`, routing is performed at the sequence level, and all tokens in a sequence share the same $\mathbf{u}$, which is the FFN input vector $\mathbf{h}_{\texttt{[CLS]}}$ representing the special prefix token `[CLS]`. For `Tok-R`, the router independently routes each token, *i.e.,* for the j-th token in the sequence, $\mathbf{u}$ is set to the corresponding FFN input vector $\mathbf{h}_j$.

For `Seq-R` and `Tok-R`, to avoid routing all inputs to the same expert FFN sub-layer, we further apply the entropic regularization

$$L_{ent} = -\sum_{i=1}^{I} P(i) \log P(i). \quad (4)$$

where $P(i) = \texttt{Softmax}(\mathbf{A}\mathbf{h}+\mathbf{c})_i$ is the probability of the i-th expert FFN sub-layer being selected. Hence, the joint training objective is

$$L_{joint} = L_{sim} + \beta L_{ent}, \quad (5)$$

where β is a scalar hyperparameter. In our work, we fix $\beta = 0.01$.

Also, all specialized Transformer blocks use the same number of expert FFN sub-layers for simplicity.

Model	I	# Params	Dev	Test
DPR	-	218M	-	78.4
$\texttt{TASER}_{\texttt{Shared}}$	1	109M	78.2	79.3
$\texttt{TASER}_{\texttt{Det-R}}$	2	128M	**79.2**	**80.7**
$\texttt{TASER}_{\texttt{Seq-R}}$	2	128M	**79.2**	80.6
$\texttt{TASER}_{\texttt{Seq-R}}$	4	166M	78.4	80.1
$\texttt{TASER}_{\texttt{ToK-R}}$	2	128M	78.5	79.8
$\texttt{TASER}_{\texttt{ToK-R}}$	4	166M	78.5	79.8
DPR†	-	218M	-	81.3
$\texttt{TASER}_{\texttt{Det-R}}$†	2	128M	**82.4**	**83.7**

Table A1: R@20 on NQ dev and test sets under the single-set training setting. I is the number of expert FFNs. The # params column shows the number of parameters in the model. † means the model is trained with hard negatives mining described in §B. The results for DPR and DPR† are reported in (Karpukhin et al., 2020) and `https://tinyurl.com/yckar3f6`, respectively.

B Hard Negative Mining

Recall that in Equation 2 the objective L_{sim} needs to use a set of negative passages $\mathcal{P}$ for each question. There are several ways to construct $\mathcal{P}$. In (Karpukhin et al., 2020), the best setting uses two negative passages per question: one is the top passage retrieved by BM25 which does not contain the answer but match most question tokens, and the other is chosen from the gold positive passages for other questions in the same mini-batch. Recent work shows that mining harder negative examples with iterative training can lead to better performance (Xiong et al., 2020; Qu et al., 2021). Hence, in this paper, we also train `TASER` with hard negatives mining. Specifically, we first train a `TASER` model with negative passages $\mathcal{P}_1$ same as Karpukhin et al. (2020). Then, we use this model to construct $\mathcal{P}_2$ by retrieving top-100 ranked passages for each question excluding the gold passage. In the single-set training, we combine $\mathcal{P}_1$ and $\mathcal{P}_2$ to train the final model. In the multi-set training, only use $\mathcal{P}_2$ is used to train the final model for efficiency consideration.

C Comparing `TASER` Variants

In this part, we compare different `TASER` variants discussed in §A by evaluating their performance on NQ under the single-set training setting. We use the bi-encoder dense passage retriever (DPR) from (Karpukhin et al., 2020) as our baseline. All models

Model	NQ		TriviaQA		WebQ		TREC		SQuAD	
	@20	@100	@20	@100	@20	@100	@20	@100	@20	@100
BM25[(1)]	62.9	78.3	76.4	83.2	62.4	75.5	80.7	89.9	71.1	81.8
Single-Set Training										
DPR[(2)]	78.4	85.4	79.4	85.0	73.2	81.4	79.8	89.1	63.2	77.2
DPR-PAQ[(3)]	84.7	**89.2**	-	-	-	-	-	-	-	-
coCondenser[(4)]	84.3	89.0	83.2	87.3	-	-	-	-	-	-
Multi-Set Training (without SQuAD)										
DPR[(1)]	79.5	86.1	78.9	84.8	75.0	83.0	88.8	93.4	52.0	67.7
$\text{DPR}^{(1)}_{\text{BM25}}$	82.6	88.6	82.6	86.5	77.3	84.7	90.1	95.0	75.1	84.4
xMoCo[(5)]	82.5	86.3	80.1	85.7	78.2	84.8	89.4	94.1	55.9	70.1
SPAR-Wiki[(6)]	83.0	88.8	82.6	86.7	76.0	84.4	89.9	95.2	73.0	83.6
SPAR-PAQ[(6)]	82.7	88.6	82.5	86.9	76.3	85.2	90.3	95.4	72.9	83.7
Multi-Set Training (with SQuAD)										
$\text{DPR}^{\dagger}$	80.9	86.8	79.6	85.0	74.0	83.4	88.0	94.1	63.1	77.2
$\text{DPR}^{\diamond}$	82.5	88.0	81.8	86.4	77.8	84.7	91.2	95.5	67.1	79.8
$\text{DPR}^{\star}$	83.7	88.7	82.6	86.7	78.9	85.3	91.6	95.1	68.0	80.2
$\texttt{TASER}^{\diamond}$	83.6	88.6	82.0	86.6	77.9	85.4	91.1	95.7	69.7	81.2
$\texttt{TASER}^{\diamond}_{\text{BM25}}$	83.8	88.6	83.3	87.1	78.7	85.7	91.6	95.8	77.2	86.0
$\texttt{TASER}^{\star}$	84.9	**89.2**	83.4	87.1	78.9	85.4	90.8	**96.0**	72.9	83.4
$\texttt{TASER}^{\star}_{\text{BM25}}$	**85.0**	**89.2**	**84.0**	**87.5**	**79.6**	**85.8**	**92.1**	**96.0**	**78.0**	**87.0**

Table A2: In-domain evaluation results. Test set R@20 and R100 are reported. [(1)]: (Ma et al., 2021). [(2)]: (Karpukhin et al., 2020). [(3)]: (Oğuz et al., 2021) [(4)]: (Gao and Callan, 2022). [(5)]: (Yang et al., 2021). [(6)]: (Chen et al., 2022). $_{\text{BM25}}$: combined with BM25 scores. †: initialized from BERT-base and without hard negatives mining. $^{\diamond}$: initialized from BERT-base. *: initialized from coCondenser-Wiki. The last five models are trained with the same hard negatives mining.

Model	Num. Parameters
DPR	218M
coCodenser	218M
xMoCo	218M
SPAR-Wiki; SPAR-PAQ	436M
DPR-PAQ	710M
$\texttt{TASER}^{\diamond}$; $\texttt{TASER}^{\star}$	128M

Table A3: Number of parameters for models reported in Table A2.

including DPR are initialized from the BERT-base (Devlin et al., 2019).[3] All TASER models are fine-tuned up to 40 epochs with Adam (Kingma and Ba, 2014) using a learning rate chosen from $\{3e-5, 5e-5\}$. Model selection is performed on the development set following (Karpukhin et al., 2020).

[3]Without further specification, we only consider the uncased version throughout the paper.

Results are summarized in Table A1.

$\texttt{TASER}_{\text{Shared}}$ is a variant without any task-aware specialization, *i.e.,* there is a single expert FFN sub-layer in the specialized Transformer block and the router is a no-op. As shown in Table A1, it outperforms DPR while using only 50% parameters.

Task-aware specialization brings extra improvements, with little increase in model size. Comparing the two learned routing mechanisms, Seq-R achieves slightly better results than Tok-R, indicating specializing FFNs based on sequence-level features such as sequence types is more effective for ODQA dense retrieval. This is consistent with the positive results for Det-R, which consists of two expert FFNs specialized for questions and passages, respectively. We also find that adding more expert FFNs does not necessarily bring extra gains, and $I = 2$ is sufficient for NQ. Consistent with the results on DPR, the hard negatives mining described in §B can further boost $\texttt{TASER}_{\texttt{Det-R}}$ per-

	Macro R@20	Micro R@20	Micro R@100
BM25	71.2	70.8	79.2
$\text{DPR}_{\text{Multi}}$	56.7	56.6	70.1
TASER$^{\diamond}$	64.7	64.3	76.2
TASER*	66.7	66.2	77.9

Table A4: Out-of-domain evaluation results on EntityQuestions. We report macro R@20 scores which are used in (Sciavolino et al., 2021) as well as micro R@20 and R@100 scores which are used in (Chen et al., 2022). Results for BM25 and $\text{DPR}_{\text{Multi}}$ are from (Sciavolino et al., 2021) and (Chen et al., 2022).

formance by 3.0 points in test set R@20. Since Det-R achieves the best R@20, our subsequent experiments focus on this simple and effective specialization strategy. In the remainder of the paper, we drop the subscript and simply use TASER to denote models using Det-R.

D Details about In-Domain Evaluations

All TASER models are fine-tuned up to 40 epochs with Adam (Kingma and Ba, 2014) using a learning rate chosen from $\{3e-5, 5e-5\}$. In our experiments, hard negatives are mined from NQ, TriviaQA and WebQ. We combine NQ and TriviaQA development sets for model selection.

We also present results of hybrid models that linearly combine the dense retrieval scores with the BM25 scores,

$$\text{sim}(\mathbf{q}, \mathbf{p}) + \alpha \cdot \text{BM25}(\mathbf{q}, \mathbf{p}). \quad (6)$$

We search the weight α in the range $[0.5, 2.0]$ with an interval of 0.1 based on the combined development set mentioned above. Unlike (Ma et al., 2021), we use a single α for all five datasets instead of dataset-specified weights so that the resulting hybrid retriever still complies with the multi-set setting in a strict sense. The same normalization techniques described in (Ma et al., 2021) is used. Similar to (Karpukhin et al., 2020; Ma et al., 2021), we separately retrieve K' candidates from TASER and BM25, and then retain the top K based on the hybrid scores, though we use a smaller $K' = 100$.

We used 16 V100-32GB GPUs and it took 9 hours to train our models.

E Dataset Licenses and Intended Use

All datasets used in our experiments are English datasets. The datasets used in this paper are released under the following licenses.

- NaturalQuestions (Kwiatkowski et al., 2019b): CC-BY-SA 3.0 License
- TriviaQA (Joshi et al., 2017): non-commercial research purposes only
- WebQuestions (Berant et al., 2013): CC-BY 4.0 License.
- SQuAD (Rajpurkar et al., 2016): CC-BY-SA 4.0 License
- EntityQuestions (Sciavolino et al., 2021): MIT License
- ArguAna (Wachsmuth et al., 2018): not specified
- DBPedia (Hasibi et al., 2017): not specified
- FEVER (Thorne et al., 2018): license terms described on the applicable Wikipedia article pages, and CC BY-SA 3.0 License
- HotpotQA (Yang et al., 2018): CC BY-SA 4.0

Our use is consistent with their intended use.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 7

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.

☑ B1. Did you cite the creators of artifacts you used?
Section 4.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix E.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix E.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
These are widely used datasets for benchmarking.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix E.

☒ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We followed the exact same setting as Karpukhin et al., (2020) and explicitly mentioned this in the paper.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Table A1 and Table A3. Appendix D.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix D.

☒ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Our results are from a single run and it is transparent from the description.

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
We will release the model and code.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.