SamToNe: Improving Contrastive Loss for Dual Encoder Retrieval Models with Same Tower Negatives

Fedor Moiseev* **Gustavo Hernández Ábrego** **Peter Dornbach**
Imed Zitouni **Enrique Alfonseca** **Zhe Dong*†**
Google Inc.
{femoiseev, gustavoha, dornbach, izitouni, ealfonseca, zhedong}@google.com

Abstract

Dual encoders have been used for retrieval tasks and representation learning with good results. A standard way to train dual encoders is using a contrastive loss with in-batch negatives. In this work, we propose an improved contrastive learning objective by adding queries or documents from the same encoder towers to the negatives, for which we name it as "contrastive loss with SAMe TOwer NEgatives" (SamToNe). By evaluating on question answering retrieval benchmarks from MS MARCO and MultiReQA, and heterogenous zero-shot information retrieval benchmarks (BEIR), we demonstrate that SamToNe can effectively improve the retrieval quality for both symmetric and asymmetric dual encoders. By directly probing the embedding spaces of the two encoding towers via the t-SNE algorithm (van der Maaten and Hinton, 2008), we observe that SamToNe ensures the alignment between the embedding spaces from the two encoder towers. Based on the analysis of the embedding distance distributions of the top-1 retrieved results, we further explain the efficacy of the method from the perspective of regularisation.

1 Introduction

The dual encoder architecture applied to information retrieval has shown excellent performance in a wide range of tasks (Gillick et al., 2018; Karpukhin et al., 2020; Ni et al., 2021, 2022).

Recently, the Information Retrieval community has transitioned towards Deep Learning models that leverage large unsupervised corpus pretraining (Devlin et al., 2019; Raffel et al., 2020), which offers more powerful semantic and contextual representation for queries and documents. These models can be successfully applied to scoring tasks, e.g. Dehghani et al. (2017), or retrieval tasks, e.g. Gillick et al. (2018). In contrast, classic retrieval models, such as BM25 (Robertson and Zaragoza, 2009), rely on bag-of-words lexical overlap, term frequency heuristics, inverse document frequency and document length. This type of retrieval models does not require any training and can generalize reasonably well, but they fall short of finding documents that have low term overlap but high semantic similarity.

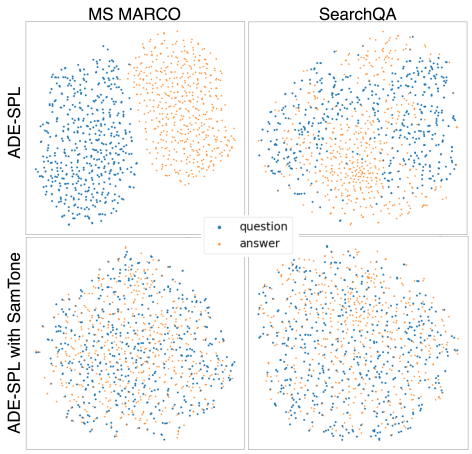

Figure 1: Embedding space analyses on MS MARCO and SearchQA show that sharing a projection layer in Asymmetric Dual Encoders (ADE-SPL) (Dong et al., 2022) may not guarantee that the embeddings from the two encoder towers are in coinciding parameter spaces. However SamToNe can effectively achieve that.

A dual encoder (Gillick et al., 2018; Yang et al., 2020; Karpukhin et al., 2020; Reimers and Gurevych, 2019) consists of two encoding towers that map queries and documents, respectively, into a shared low-dimensional dense representation, namely, the embedding space. The model is usually optimized by a contrastive loss (Chopra et al., 2005), which moves the embeddings of the queries and documents from the same positive examples closer to each other, and the embeddings from negative examples farther away. Training the dual encoder in batches allows to use, for each question, the passages that answer all the other questions within the batch as negatives (Gillick et al., 2018), namely "in-batch negatives". At indexing time, all

* These authors contributed equally.
† Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12028–12037

the documents in a corpus are encoded via bulk inference and indexed. To run retrieval, a query is encoded and its most relevant documents can be retrieved through Nearest Neighbours Search (Vanderkam et al., 2013; Johnson et al., 2021) over the embedding space using a measure of similarity, e.g. the dot-product or cosine distance of the embedding vectors.

Motivation. In this work, we consider two major types of dual encoder architectures: "Symmetric Dual Encoder" (SDE)[1], with parameters shared between two encoder towers, and "Asymmetric Dual Encoder" (ADE), with two distinctly parameterized encoder towers. Dong et al. (2022) demonstrated that sharing projection layers can significantly improve the performance of ADEs. They empirically explained the efficacy of SDE and ADE-SPL by claiming that the shared projection layers help mapping the embeddings of the two encoder towers into a coinciding parameter space.

By repeating this embedding space analysis on a variety tasks, we find that ADE-SPL may not be enough to ensure that the embedding spaces from two encoder towers are coinciding, as shown in Figure 1. This motivates us to further improve the dual encoder retrieval quality beyond the architectural change explored in Dong et al. (2022). Although the projection layers are shared, our analyses suggest that an extra mechanism, other than using the standard contrastive loss with in-batch negatives, is required to ensure the adjacency of the embeddings of a ground truth pair.

Contributions. In this paper, we propose an improved training objective for dual encoder models: *contrastive loss with Same Tower Negatives* (**SamToNe**). In Section 3, we demonstrate its usefulness on a variety of Information Retrieval tasks, including both tasks with in-task fine-tuning and a zero-shot benchmark suite. Across all the tasks explored, SamToNe performs competitively comparing to the traditional training setup, with a significant improvement on the metrics averaged across tasks. Finally, through an analysis of the produced embeddings, in Section 4, we further make evident the superiority of SamToNe from the perspective of regularisation.

[1]This kind of dual encoders have also been called "Siamese" or "Twin" dual encoders.

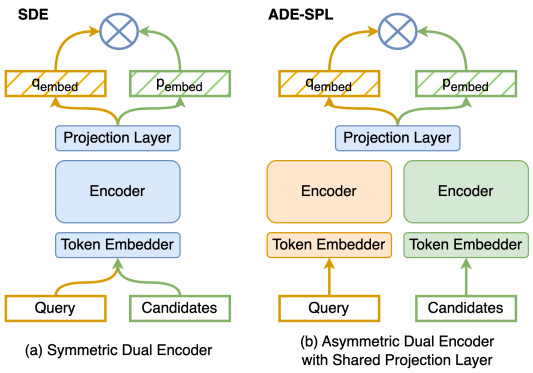

Figure 2: The dual encoder architectures, where the blue components are shared between two encoding paths.

2 Method

Dual Encoder Architecture. We follow the standard setup of information retrieval: given a query, q, and a corpus of retrieval candidates, $\mathcal{P}$, the goal is to retrieve k relevant candidates, $p_k \in \mathcal{P}$. The candidate can be a phrase, a sentence, a passage, or a document.

Recent research (Dong et al., 2022) demonstrated that sharing projection layers can significantly improve the performance of ADEs and we use this shared projection layer for ADEs (ADE-SPL) throughout our experiments. Figure 2 illustrates the SDE and ADE-SPL architectures we use in this work. Our dual encoders are initialized from pre-trained `t5.1.1` encoders (Raffel et al., 2020). Following Ni et al. (2022); Dong et al. (2022), we encode a query, q_i, or a candidate, p_i, by averaging the T5 encoder outputs and projecting them to the final embedding vector.

Contrastive Loss. A standard way to train a dual encoder model is optimizing an in-batch sampled softmax loss for contrastive learning (Henderson et al., 2017):

$$\mathcal{L}_c = \frac{\exp(\texttt{sim}(q_i, p_i)/\tau)}{\sum_{j \in \mathcal{B}} \exp(\texttt{sim}(q_i, p_j)/\tau)}, \quad (1)$$

where `sim` is cosine similarity, $\mathcal{B}$ is a mini-batch of examples, and τ is the softmax temperature. p_i is the ground-truth relevant passage for the query q_i in a batch of retrieval candidates p_*, where all the other passages p_k ($k \neq i$) are treated as the negative examples for contrastive learning.

Bi-directional in-batch sampled softmax loss is commonly applied to improve the embedding quality of both towers, where the contrastive loss is computed for both query to passage matching and passage to query matching (Yang et al., 2019). We use the bi-directional loss throughout this work.

Same Tower Negatives. The in-batch sampled softmax loss is a contrastive loss that only considers the contrastive estimation between the target example pair $\{q_i, p_i\}$, and the in-batch sampled negative pairs $\{q_i, p_j\}$ $(j \neq i)$.

One way to improve the quality of the retrieval is to improve the contrast among the embeddings of the queries. Therefore, we propose a novel contrastive loss using **Same** **To**wer **Ne**gatives, which we abbreviate as **SamToNe**:

$$\mathcal{L}_S = \frac{e^{\texttt{sim}(q_i,p_i)/\tau}}{\sum_{j\in\mathcal{B}} e^{\texttt{sim}(q_i,p_j)/\tau} + \sum_{j\in\mathcal{B},j\neq i} e^{\texttt{sim}(q_i,q_j)/\tau}}, \quad (2)$$

where the second term in the denominator is the contribution from the same tower negatives.

SamToNe can be interpreted as a regularized version of the in-batch sampled softmax loss, where the term $\sum_{j\in\mathcal{B},j\neq i} e^{\texttt{sim}(q_i,q_j)/\tau}$ is a regularizer. When query embeddings are not well distributed, $\max \texttt{sim}(q_i, q_j) \gg \max \texttt{sim}(q_i, p_j)$, and the second term in the denominator will dominate the contribution from the negative examples. Thus, it will drive the separation of the query embeddings in contrastive learning. In Section 4, we provide empirical evidence of the effects of SamToNe as a regularizer of the embedding space.

Ren et al. (2021) proposed an improved contrastive loss, PAIR, which is a hybrid loss $\mathcal{L}_{PAIR} = -(1-\alpha)\log \mathcal{L}_c - \alpha \log \mathcal{L}_P$, where

$$\mathcal{L}_P = \frac{e^{\texttt{sim}(q_i,p_i)/\tau}}{\sum_{j\in\mathcal{B},j\neq i} e^{\texttt{sim}(p_i,p_j)/\tau}} \quad (3)$$

penalizes the similarities between passages / documents. Despite both SamToNe and PAIR are penalizing the similarities among the same tower inputs, there are two significant differences. *Firstly*, SamToNe is hyper-parameter free, while PAIR introduces a new hyper-parameter α. This is because SamToNe introduces the new term from an embedding space regularization prospective (see Section 4 for detailed analysis). Therefore SamToNe can be easily applied to both query and document encoders (see Section 3.4), but PAIR needs to introduce yet another hyper-parameter to be applied to both. *Secondly*, Ren et al. (2021) mentioned it required a 2-stage training, with the first stage using the PAIR loss, and the second using regular in-batch softmax loss. Due to its self-balancing nature, SamToNe doesn't require multi-stage training. A thorough comparison against PAIR can be found in sections 3 and 4. No added hyper-parameters, single stage training and guaranteed improvement on embedding space quality, make SamToNe much easier to use.

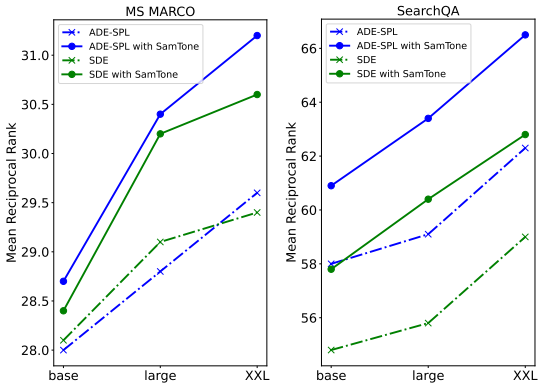

Figure 3: The impact of model sizes on the performance of different dual encoder architectures, measured by MRR on the eval set of MS MARCO (left) and SearchQA (right).

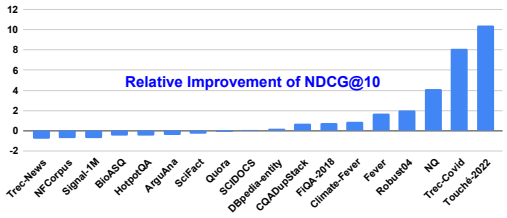

Figure 4: Relative improvement of NDCG@10 (%) on BEIR tasks, by applying SamToNe to SDE.

3 Experiments

3.1 Question-Answering Retrieval Tasks

We evaluate SamToNe on 5 question-answering (QA) retrieval tasks including MS MARCO (Nguyen et al., 2016) and MultiReQA (Guo et al., 2021). For MS MARCO, the retrieval candidates are relevant passages, and for the 4 tasks in MultiReQA, the retrieval candidates are answer sentences.

To make a fair comparison across the results of our experiments, the same fine-tuning hyper-parameters are applied to all our model variants. The models are optimized for $20,000$ steps using Adafactor optimizer (Shazeer and Stern, 2018), with softmax temperature $\tau = 0.01$, batch size 512, and a linearly decaying learning rate starting from 10^{-3} to 0 at the final step. To compare SamToNe and PAIR, we use the hyperparameter $\alpha = 0.1$ for PAIR as reported in Ren et al. (2021), and keep all the other experimental setups identical. SamToNe is applied only on the query side, as it is more robust across different datasets. For experiments and analysis on applying SamToNe on both encoder towers, please refer to Section 3.4. We benchmark

Model	Loss	MSMARCO		NQ		SQuAD		TriviaQA		SearchQA		Average	
		P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
ADE	Standard	14.1	26.8	53.5	65.2	64.3	74.0	37.9	50.4	41.5	57.2	42.3	54.7
	SamToNe	16.0	28.5	52.8	63.9	63.6	73.0	38.4	49.8	**49.2**	62.3	44.0	55.5
ADE-SPL	Standard	15.7	28.8	55.3	67.0	74.5	**82.1**	41.7	54.4	42.3	59.1	45.9	58.3
	SamToNe	**17.6**	**30.4**	**55.7**	**67.2**	73.8	81.7	44.0	55.9	48.5	**63.4**	**47.9**	**59.7**
	PAIR	16.9	29.6	55.7	67.0	74.4	82.0	**45.0**	**56.8**	44.1	60.4	47.2	59.2
SDE	Standard	16.1	29.1	54.4	66.6	74.1	81.9	41.4	54.2	37.6	55.8	44.7	57.5
	SamToNe	17.2	30.2	54.2	66.4	**74.6**	**82.0**	42.1	54.5	44.0	60.4	46.4	58.7
	PAIR	16.1	29.1	53.8	66.2	74.13	81.7	41.3	54.5	38.7	56.6	44.7	57.5

Table 1: Precision at 1 (P@1)(%) and Mean Reciprocal Rank (MRR)(%) on QA retrieval tasks. The best-performing models for each task and metric are highlighted in **bold**.

Task \ Model	SDE	SamToNe	BM25	GTR-XXL
ArguAna	40.2	39.8	31.5	**54**
BioASQ	40.2	39.7	**46.5**	32.4
Climate-Fever	31.1	**32**	21.3	26.7
CQADupStack	40.7	**41.4**	29.9	39.9
DBpedia-entity	45.7	**45.9**	31.3	40.8
Fever	68.3	70	**75.3**	74
FiQA-2018	41.8	42.6	23.6	**46.7**
HotpotQA	**66.9**	66.4	60.3	59.9
NFCorpus	**37.2**	36.5	32.5	34.2
NQ	42.9	47	29.9	**56.8**
Quora	88.8	88.7	78.9	**89.2**
Robust04	53.5	**55.5**	40.8	50.6
SCIDOCS	22.3	**22.4**	15.8	15.9
SciFact	**68**	67.7	66.5	66.2
Signal-1M	31.8	31.1	**33**	27.3
Trec-Covid	53.1	61.2	**65.6**	50.1
Trec-News	**49.2**	48.4	39.8	34.6
Touché-2022	22	**32.4**	36.7	25.6
Average	46.9	**48.3**	42.3	45.8

Table 2: NDCG@10 for zero-shot evaluation on the BEIR benchmark after fine-tuning on MSMarco. The best-performing models for each task are highlighted in **bold**, while the best scores between **SDE** and **SDE w/ SamToNe** are underscored.

the fine-tuned models using precision at 1 ($P@1$) and mean reciprocal rank (MRR).

As shown in Table 1, SamToNe greatly improves the retrieval performance of both SDE and ADE-SPL models. Using SamToNe, ADE-SPL models can outperform SDE ones, especially for TriviaQA and SearchQA, by a great margin. Relative to PAIR, SamToNe provides better performance across different datasets in both types of models.

3.2 Scaling the Model Size

To assess the impact of the model size, we evaluate the dual encoders initialized from `t5.1.1-base` ($\sim$ 250M parameters), `t5.1.1-large` ($\sim$ 800M parameters), and `t5.1.1-XXL` ($\sim$ 11B parameters). Figure 3 and Appendix Table 4 show that SamToNe consistently improves the performance of dual encoders across different model sizes.

3.3 BEIR Generalization Tasks

We further demonstrate the efficacy of the dual encoders trained with SamToNe on BEIR (Thakur et al., 2021), a heterogeneous benchmark for zero-shot evaluations.

BEIR has 18 information retrieval datasets[2] across 9 domains, including *Bio-Medical*, *Finance*, *News*, *Twitter*, *Wikipedia*, *StackExchange*, *Quora*, *Scientific*, and *Misc*. The majority of the datasets have binary query relevance labels. The other datasets have 3-level or 5-level relevance judgements.

As BEIR is evaluating generalization capabilities and SDEs are commonly used for general purpose retrieval (Ni et al., 2021), we focus on evaluating the impact of SamToNe on BEIR using the SDE architecture. In this evaluation, we reuse the model fine-tuned with MS MARCO, as described in Section 3.1.

Evaluated with the same setting as GTR (Ni et al., 2021), SamToNe demonstrates strong performance on BEIR, as shown in Table 2 and Figure 4. On average, SamToNe improves NDCG@10 by 1.4% for SDE with XXL size. SDE trained with SamToNe significantly outperform BM-25, a sparse retrieval method, and GTR, a dense retrieval method that shares the same architecture and the same model size as SDE but fine-tuned with different corpora.

3.4 Applying SamToNe to Both Towers

Just as with the query tower, SamToNe can be applied to the document tower which leads to better query-document alignment. However, it is common that the training data contains a large fraction of duplicated documents for a diverse set of queries.

[2]MS Marco is excluded from the zero-shot comparison as many baseline models use it as training data.

SamToNe	MSMARCO P@1	MSMARCO MRR	TriviaQA P@1	TriviaQA MRR
W/O SamToNe	15.7	28.8	41.7	54.4
uni-directional	17.6	30.4	**44.0**	**55.9**
bidirectional	**18.2**	**31.0**	41.7	53.3
% of unique documents	98%		17%	

Table 3: Precision at 1 (P@1)(%) and Mean Reciprocal Rank (MRR)(%) when comparing ADE-SPL (`t5.1.1-large` size) trained without SamToNe and with SamToNe applied to the query tower (*uni-directional*) or to both towers (*bidirectional*). The best-performing models for each task and metric are highlighted in **bold**.

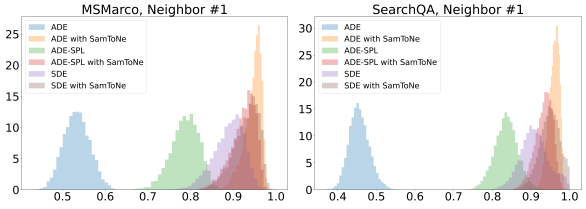

Figure 5: Distributions of cosine similarities between the embeddings of the queries and their *nearest* neighbour documents, for different models trained with or without SamToNe.

For example, only 17% of the documents in the train-split are unique for TriviaQA, but 98% for MSMARCO. For datasets with a low rate of unique documents, applying SamToNe on the document side will penalize $\texttt{sim}(p_i, p_j)$ with $p_i = p_j$ and may hinder the performance, as shown in Table 3.

4 Analysis

4.1 Embedding Space Analysis

As shown in the top row of Figure 1, for MS MARCO and SearchQA, ADE-SPL generates two connected but topologically separable embedding spaces. It requires an extra mechanism, beyond the shared projection layers, to ensure the adjacency of the embeddings from a ground truth pair.

SamToNe is proposed as the "force" drawing the embeddings of each ground truth training pair together. Its efficacy is illustrated in the bottom half of Figure 1.

4.2 SamToNe: an Embedding Distance Regularizer

To further understand SamToNe's role as a regularizer of embedding distances, we evaluate the distribution of the distances between the embeddings of the queries and their top-1 retrieval results in the test set of MS MARCO and SearchQA. The embedding distance is measured by cosine similarity, where 1.0 means perfect alignment with a range of $[-1.0, 1.0]$.

As shown in Figure 5, SamToNe drastically

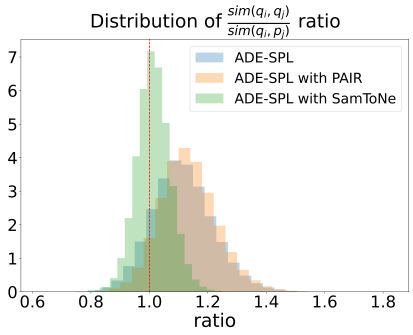

Figure 6: Distributions of query-query to query-document similarity ratios for different losses on SearchQA. SamToNe is applied to both query and document sides, and it pushes the ratio to be centered around 1.

shifts the distribution of the (query, top-1 retrieval result) pairs towards 1.0, demonstrating the regularizing effect of SamToNe over the embedding distances.

By placing the regularizing query-query similarity terms $e^{\texttt{sim}(q_i,q_j)/\tau}$ and the standard in-batch negative query-document similarity terms $e^{\texttt{sim}(q_i,p_j)/\tau}$ together in the denominator with same weight, SamToNe pushes the similarity ratio between query-query and query-documents, $\texttt{sim}(q_i, q_j)/\texttt{sim}(q_i, p_j)$, to be centered around 1.0. This is a *self-balancing* regularization effect. The query and document spaces are set to closely overlap each other and the embeddings of a positive pair are more likely to be located in the same region of the embedding space.

To empirically illustrate this effect, we plotted histograms of the $\frac{\texttt{sim}(q_i,q_j)}{\texttt{sim}(q_i,p_j)}$ ratios for randomly selected i and j in Figure 6. The regularization effect only shows when SamToNe is used, but not when PAIR (Ren et al., 2021) is. This is because the self-balancing effect does not exist in a hybrid loss such as PAIR.

5 Conclusions

Evaluating on QA retrieval tasks and zero-shot generalization benchmarks, we demonstrate that training with SamToNe can significantly improve the dual encoder retrieval quality. With t-SNE maps of query and document embeddings, we show that the embedding spaces from the two encoding towers of models trained with SamToNe are better aligned. Through the distributions of similarity distances between the embeddings of queries and their nearest neighbours, we empirically explain the efficacy of SamToNe from a regularisation prospective. In general, we recommend using SamToNe to train dual encoders for information retrieval tasks.

6 Limitations

Same tower negatives can be applied to other contrastive losses, e.g. triplet loss (Chechik et al., 2010). As we are focusing on improving the most popular method to train dual encoder models, i.e. the in-batch sampled softmax loss, we leave the application of same tower negatives to other types of contrastive loss as future work.

While SamToNe has proven to be effective to improve the training of dual encoders, its efficacy may depend on the diversity of the queries used as inputs. In dataset with a large portion of similar queries in the training set, one might need to use masking or other techniques to remove them from the negative computation. Such techniques can also improve the efficacy of SamToNe when applied to both the query and document towers, where SamToNe is currently known to hinder the performance on datasets with a low rate of unique documents, as discussed in Section 3.4.

We leave the in-depth exploration of aforementioned considerations for future works.

References

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(36):1109–1135.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural ranking models with weak supervision. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 65–74.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Zhe Dong, Jianmo Ni, Daniel M. Bikel, Enrique Alfonseca, Yuan Wang, Chen Qu, and Imed Zitouni. 2022. Exploring dual encoder architectures for question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 9414–9419. 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.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

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.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large dual encoders are generalizable retrievers.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21/140.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng

Wang, and Ji-Rong Wen. 2021. PAIR: Leveraging passage-centric similarity relation for improving dense passage retrieval. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2173–2183, Online. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

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.

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)*.

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

Dan Vanderkam, Rob Schonberger, Henry Rowley, and Sanjiv Kumar. 2013. Nearest neighbor search in google correlate. Technical report, Google.

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.

Yinfei Yang, Gustavo Hernandez Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5370–5378. International Joint Conferences on Artificial Intelligence Organization.

A Appendix

Model size	Architecture	SamToNe	MSMARCO		NQ		SQuAD		TriviaQA		SearchQA		Average	
			P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR	P@1	MRR
base	ADE	No	13.8	25.8	48.7	60.1	60.9	70.7	35	46.3	41.7	57.1	40	52
		Yes	15.1	27.1	46.1	57.	59	68.9	32.5	43.1	45.3	58.5	39.6	50.9
	ADE-SPL	No	15.4	28.	50.5	62.1	69.8	78.1	38.8	50.7	41.6	58.	43.2	55.4
		Yes	**16**	**28.7**	**50.9**	**62.3**	69.9	78.1	**40.4**	**51.7**	**45.8**	**60.9**	**44.6**	**56.3**
	SDE	No	15.7	28.1	49.3	61.4	70.2	**78.5**	37.7	50.4	36.9	54.8	42	54.6
		Yes	15.9	28.4	49.7	61.6	**70.4**	0.784	39.4	51.5	41.1	57.8	43.3	55.5
large	ADE	No	14.1	26.8	53.5	65.2	64.3	74	37.9	50.4	41.5	57.2	42.3	54.7
		Yes	16	28.5	52.8	63.9	63.6	73	38.4	49.8	49.2	62.3	44	55.5
	ADE-SPL	No	15.7	28.8	55.3	67	74.5	**82.1**	41.7	54.4	42.3	59.1	45.9	58.3
		Yes	**17.6**	**30.4**	**55.7**	**67.2**	0.738	0.817	**44**	**55.9**	**48.5**	**63.4**	**47.9**	**59.7**
	SDE	No	16.1	29.1	54.4	66.6	74.1	81.9	41.4	54.2	37.6	55.8	44.7	57.5
		Yes	17.2	30.2	54.2	66.4	**74.6**	82	42.1	54.5	44	60.4	46.4	58.7
XXL	ADE	No	14.9	27.9	57.2	69.2	68.7	77.8	46.1	58.7	47.4	62.7	46.9	59.3
		Yes	17	30	57.5	69	67.7	76.9	47	58.8	52.7	65.9	48.4	60.1
	ADE-SPL	No	16.2	29.6	58.7	70.6	**78.3**	**85.3**	**50.9**	**63**	45.7	62.3	50	62.2
		Yes	**17.7**	**31.2**	**59.8**	**71.4**	77.9	84.8	50.1	61.6	**51.9**	**66.5**	**51.5**	**63.1**
	SDE	No	15.8	29.4	58.2	70.6	79.2	86	46.9	60.3	40.6	59	48.1	61.1
		Yes	17.1	30.6	58.7	70.8	78.2	85.1	48.3	60.6	46.5	62.8	49.8	62
Dataset Size (train / test queries / test documents)			400776 / 6980 / 8841823		106521 / 4131 / 22118		87133 / 10485 / 10642		335659 / 7776 / 238339		629160 / 16476 / 454836			

Table 4: Precision at 1(P@1)(%) and Mean Reciprocal Rank (MRR)(%) on QA retrieval tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6

☐ A2. Did you discuss any potential risks of your work?
Not applicable. The paper uses public dataset and standard training recipes commonly used in existing papers.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and 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 2 and 3

☑ B1. Did you cite the creators of artifacts you used?
Section 2 and 3

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ 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)?
Not applicable. Left blank.

☐ 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?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ 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.
Appendix Table 3

C ☑ Did you run computational experiments?

Section 3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 2 and Section 3

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?
Section 2 and Section 3

☑ 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?
Section 3

☑ 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.)?
Section 2

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.