

SOFTQE: Learned Representations of Queries Expanded by LLMs

Varad Pimpalkhute^{1,*}, John Heyer², Xusen Yin², and Sameer Gupta²

¹ University of Massachusetts Amherst

² Alexa AI, Amazon

pimpalkhutevarad@gmail.com, {heyjohn, yxusen, gupsam}@amazon.com

Abstract. We investigate the integration of Large Language Models (LLMs) into query encoders to improve dense retrieval without increasing latency and cost, by circumventing the dependency on LLMs at inference time. SOFTQE incorporates knowledge from LLMs by mapping embeddings of input queries to those of the LLM-expanded queries. While improvements over various strong baselines on in-domain MS-MARCO metrics are marginal, SOFTQE improves performance by 2.83 absolute percentage points on average on five out-of-domain BEIR tasks.

1 Introduction

Query expansion [15,22] methods aim to expand search queries with additional terms to improve downstream information retrieval (IR) performance. Expansion terms can come directly from highly ranked documents, as in pseudo relevance feedback based methods like RM3 [15,17], or from generative models as in methods like GAR [19]. While query expansion can mitigate the *token mismatch* problem that plagues sparse retrieval methods like BM25 [21], which depend on token overlap between queries and documents, dense retrieval methods [11,14] offer a natural solution by embedding queries and documents in a shared feature space wherein queries and documents with strong *semantic* overlap are close.

Recent methods [9,13,30] prompt Large Language Models (LLMs) [1,4,26] to expand queries with relevant terms or "pseudo-documents" that resemble real passages from the corpus. Perhaps surprisingly, *query2doc* (Q2D) [30] demonstrates improved performance of *dense* retrievers, indicating that LLM-based query expansion can facilitate learning the semantic overlap between underspecified queries and document corpora. However, adding an LLM to a real-time IR pipeline is often prohibitively expensive in terms of both cost and latency. Motivated by both the promise of LLM-based query expansion for dense retrieval *and* its impracticality, we propose *Soft Query Expansion* (SOFTQE), wherein we learn to estimate the representations of LLM expansions *offline* during training, thus circumventing the dependency on LLMs at runtime as shown in Figure 1. SOFTQE performs at least as well as baseline dense retrievers such as

* Work done as an intern at Amazon

DPR [14], and stronger alternatives combining large-scale pretraining and cross-encoder distillation such as SimLM [28] and E5 [29], on in-domain MS-MARCO [2], TREC DL 2019 and 2020 datasets [5,6]. Further, SOFTQE significantly improves upon these baselines for a majority of out-of-domain BEIR [25] tasks. Our findings corroborate those of Q2D, specifically that the increase in retrieval performance diminishes when combined with stronger encoders. However, we observe measurable improvements in the zero-shot setting, suggesting that information learned through the SOFTQE objective is complementary to other forms of distillation, such as distillation from a cross-encoder.

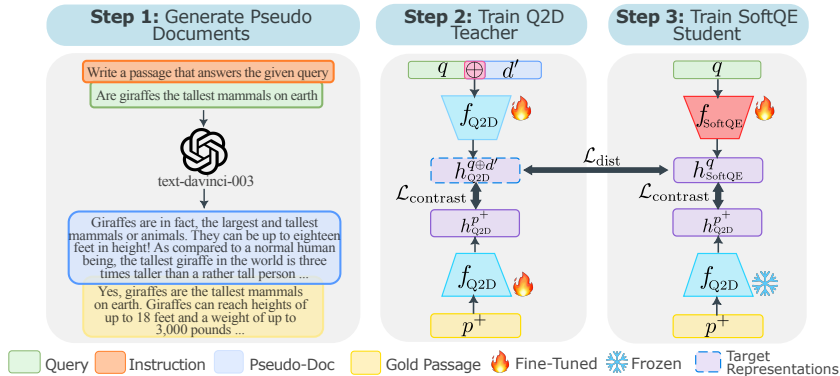


Fig. 1: Overview of the SOFTQE approach. **Step 1:** Given a query, prompt an LLM to generate a pseudo-document d' , as in Q2D [30]. **Step 2:** Train teacher encoder using the Q2D method and expanded queries from Step 1 ($q \oplus d'$). **Step 3:** Train SOFTQE encoder to align query representations with the expanded query representations from Step 2, in addition to the standard contrastive objective. h_y^x denotes the *representation* (e.g., the last hidden state of the CLS token) given an input x and encoder y .

2 Method

Expanded Queries. An expanded query q^+ is formed by appending a pseudo-document d' to the original query, q :

$$q^+ = q \oplus g_\phi(\mathcal{I}, q), \quad (1)$$

where, g_ϕ is an LLM that generates pseudo document (d') with prompt \mathcal{I} , employing techniques such few-shot, chain of thought [31], etc. We use the pseudo-documents released³ with Q2D [30], which were generated by *text-davinci-003* [1] using an instruction and examples of positive query/document pairs from MS MARCO [2]. An example pseudo document is shown in Figure 1.

³ Pseudo-documents generated using text-davinci-003 for MS MARCO queries are released by [30] here: https://huggingface.co/datasets/intfloat/query2doc_msmarco

Dual-Encoder Training. Dual encoders are typically trained by optimizing a contrastive objective [14]:

$$\mathcal{L}_{\text{cont}} = -\log\left(\frac{e^{h_q \cdot h_{p^+}}}{e^{h_q \cdot h_{p^+}} + \sum_{i=1}^N e^{h_q \cdot h_{p_i^-}}}\right), \quad (2)$$

where h_q and h_p represent query and passage embeddings, respectively, and N is the number of negative passages. In Q2D, embeddings of *expanded* query inputs (h_{q^+}) are learned, and BM25 hard negatives are used.

SOFTQE Objective. Driven by the superior performance of Q2D, we seek to align representations of queries with their expanded counterparts. We do so by introducing an additional distance component⁴, $\mathcal{L}_{\text{dist}}$, to the loss:

$$\mathcal{L}_{\text{SoftQE}} = \alpha \mathcal{L}_{\text{dist}}(f_\theta(q^+), f_\psi(q)) + (1 - \alpha) \mathcal{L}_{\text{cont}}, \quad (3)$$

where f_θ and f_ψ are transformer-based [27] encoders that map expanded queries and queries to vectors in the learned embedding space respectively, and α is a hyper parameter that controls the weight assigned to each component of the loss, as in knowledge distillation [12]. In other words, the expanded query representations produced by the Q2D encoder (teacher) serve as target query representations used to distill information into the SOFTQE query encoder (student). Importantly, the feature space is *pre-defined* by the Q2D dual-encoder, rather than updated during training. Accordingly, we only learn to embed *queries*, and reuse the Q2D encoder to produce passage embeddings as they are already well-aligned with the target query representations.

We additionally experiment with state-of-the-art dense retrievers [28,29] that are trained using KL divergence from cross-encoder scores [20]. We apply SOFTQE to distilled retrievers by simply combining the 3 objective terms with an additional weight controlled by β :

$$\mathcal{L}_{\text{SQE+KD}} = \alpha \mathcal{L}_{\text{dist}}(f_\theta(q^+), f_\psi(q)) + (1 - \alpha) [\beta \text{KL}(f_\theta, f_{\text{CE}}) + (1 - \beta) \mathcal{L}_{\text{cont}}], \quad (4)$$

as we find the information distilled through cross-encoder scores and expanded query representations to be complementary.

3 Experiments

Datasets, Metrics, and Baselines. For in-domain evaluation, we use the MS MARCO Passage Ranking [2], TREC DL 2019 [5] and TREC DL 2020 [6] datasets. Following Q2D [30], we evaluate zero-shot performance on five low-resource tasks from the BEIR benchmark [25], namely: SciFact, NFCorpus, Trec-Covid, DBPedia and Touche-2020. Evaluation metrics include MRR@10, R@50, R@1k, and nDCG@10. We benchmark SOFTQE against a DPR [14] dense retrieval baseline, and two state-of-the-art dense retrievers: SimLM [28], and E5 [29].

⁴ In practice, we find no significant difference between distance metrics, so we simply use mean squared error (MSE).

Hyperparameters. We follow the hyperparameter settings used in [30], with a few distinctions. We initialize our DPR models from BERT_{base} [7], and our SOFTQE variants of SimLM [28], and E5 [29] from their corresponding public checkpoints. When fine-tuning with cross-encoder distillation, β is set to 0.2, following SimLM [28]. We set α to 1.0 for 3 epochs in order to establish an initial alignment with the target expanded query embeddings, then relax α to 0.2 as well. This choice is further discussed in Section 4.

Table 1: Results on in-domain MS MARCO and TREC DL datasets, grouped by retrievers trained with and without distillation from cross-encoders. Underline: best result including Q2D, which requires an LLM at inference time; **Bold**: highest result among non-Q2D solutions; *: our reproduction; †: denotes statistical significance with a p-value less than 0.05 using a paired T-test.

Method	MS MARCO Dev Set			TREC DL 19	TREC DL 20
	MRR@10	R@50	R@1k	nDCG@10	nDCG@10
<i>Dual-encoder without distillation</i>					
DPR*	33.74	80.90	96.18	64.04	62.81
+ SOFTQE	33.87	81.24 †	96.25 †	65.22 †	63.80 †
+ Q2D*	<u>35.26</u>	<u>82.78</u>	<u>97.21</u>	<u>70.54</u>	<u>66.68</u>
<i>Dual-encoders distilled from cross-encoders</i>					
SimLM*	41.13	87.78	98.69	71.40	69.68
+ SOFTQE	41.15	87.93 †	98.61†	70.50†	70.10†
+ Q2D*	<u>41.45</u>	<u>88.43</u>	<u>98.82</u>	74.59	71.37
E5*	40.70	87.13	98.50	72.52	71.38
+ SOFTQE	40.30†	87.22	98.50	72.82 †	71.73 †
+ Q2D*	40.93	87.95	98.76	<u>75.03</u>	<u>73.27</u>

Results. We first evaluate the performance on in-domain datasets (Table 1). SOFTQE consistently improves upon DPR across all metrics on MS MARCO, TREC DL 19 and TREC DL 20 datasets. When evaluating the performance against dual-encoders distilled from cross-encoders, we notice that SOFTQE and SimLM perform closely with SOFTQE slightly underperforming in R@1k on MS MARCO and nDCG@10 on TREC DL2019. Similarly, SOFTQE results in marginal improvements over E5. This finding corroborates the claim in [30] that improvements diminish when encoders are distilled from strong cross-encoders.

Table 2: Results on out-of-domain BEIR benchmark datasets by nDCG@10. Underline: best result including Q2D, which requires an LLM at inference time; **Bold**: highest result among non-Q2D solutions; *: our reproduction⁵; †: denotes statistical significance.

Method	SciFact	NFCorpus	Trec-Covid	DBPedia	Touche-2020	Average
DPR*	51.85	25.72	44.81	30.99	19.91	34.65
+ SOFTQE	49.81†	25.73	60.02 †	31.82 †	20.59	37.59
SimLM*	61.42	32.38	52.90	35.06	19.21	40.19
+ SOFTQE	61.72	32.34	61.78	36.75	21.94	42.91
+ Q2D [30]	<u>59.50</u>	32.10	<u>59.90</u>	<u>38.30</u>	<u>25.60</u>	<u>43.08</u>

Table 2 highlights the zero-shot evaluation results on out-of-domain datasets from BEIR. SOFTQE considerably outperforms DPR and SimLM, by 2.94 and 2.72 absolute percentage points, respectively, averaged across tasks. SOFTQE yields marginal differences in performance on tasks where Q2D results in regressions (SciFact and NFCorpus), but substantial improvements on the remaining tasks when applied to either DPR or SimLM, indicating that SOFTQE is complementary to cross-encoder distillation.

4 Discussion

Is Fine-tuning on Expanded Queries Necessary? Traditional query expansion methods applied to lexical systems do not require modifications to the retrieval algorithm. Q2D [30], however, requires fine-tuning the dense retriever on expanded queries, as demonstrated by the the difference between the first 2 rows in Table 3. Simply passing expanded queries to an off-the-shelf DPR model actually deteriorates performance, which is somewhat surprising given the model’s ability to effectively embed queries and passages *independently*.

Table 3: MS Marco MRR@10 of DPR and Q2D with query (q) and expanded query (q^+) inputs. DPR has not been trained with expanded query inputs, while Q2D has.

Method	TREC DL19	TREC DL20
DPR(q^+)	61.65	59.45
+ Q2D(q^+)	70.54	66.68
DPR(q)	64.04	62.81
+ Q2D(q)	57.78	57.12

Table 4: TREC nDCG@10 across four variations of α in the training objective: ($\alpha\mathcal{L}_{\text{cont}}+(1-\alpha)\mathcal{L}_{\text{dist}}$). In "Warm up", we set alpha to 1 for the first 3 epochs, then to 0.2 for the remaining 3.

Method	α	TREC DL19	TREC DL20
$\mathcal{L}_{\text{dist}}$ Only	1	61.62	62.81
$\mathcal{L}_{\text{cont}}$ Only	0	64.66	63.42
Combined	0.2	63.33	63.03
Warm up	1→0.2	65.23	63.79

Combining L_{dist} and L_{cont} . In Table 4, we explore four variations of the training objective in Equation 3 to determine how to balance supervision from labeled passages vs. target representations. A perfect mapping ($L_{\text{dist}} = 0$) between query and expanded query representations would yield Q2D performance, but is not realistic, as evident by the subpar performance of " L_{dist} Only". Combining the two losses by setting α to 0.2 results in query embeddings that are no closer to the target embeddings produced when using only a contrastive loss, as shown by the MSE Loss plot in Figure 2 (right). To remedy this, we propose a step-wise "warm up" method, in which we set α to 1 (L_{dist} only) for 3 epochs to establish a strong alignment with the target representations, then relax α to 0.2 for the remaining 3 epochs. Figure 2 demonstrates that this reduces L_{dist} while negligibly impacting L_{cont} , resulting in the best performance in Table 4.

Should we also Fine-tune the Passage Encoder? We decided not to fine-tune the passage encoder during SOFTQE training, because it allowed us to

⁵ We could not reliably reproduce the E5 results on BEIR datasets, but Q2D did not yield significant improvements when applied to E5, so we assume the same for SOFTQE and omit E5 from zero-shot evaluation.

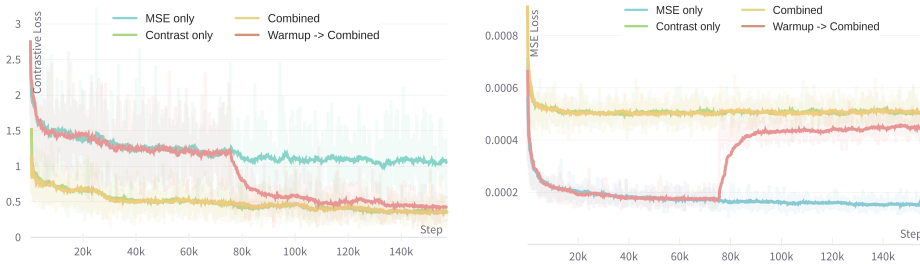


Fig. 2: Training curves of four settings of α shown in Table 4. **Left:** Contrastive Loss - *does it reject negative documents?* MSE-only performs the worst in terms of contrastive loss, while Warmup \rightarrow Combined converges to the same loss as Combined. **Right:** MSE Loss - *is it close to the teacher?* Contrast-only has the highest MSE loss, while Warmup \rightarrow Combined MSE loss increases after the warmup, but converges to a value noticeably lower than Combined.

re-use the Q2D passage-encoder. Intuitively, this means that the space in which passages and queries are embedded is the same as in Q2D. In Table 5 we show that, on average, fine-tuning the passage encoder results in reduced performance. This is assuring – if the passage representations were to change, our Q2D representation targets would be unfounded, as they would no longer be optimally aligned with the passages.

Table 5: Unfreezing the passage encoder during training results in a degradation of performance on TREC nDCG@10.

Freeze Encoder	Method	DL19	DL20
\times	SOFTQE	65.59	62.06
\checkmark	SOFTQE	65.22	63.80

Table 6: Comparing our method to traditional knowledge distillation (using only model predictions) on TREC nDCG@10.

Method	DL19	DL20
DPR	64.04	62.81
+ Traditional KD	61.12	62.14
+ SOFTQE	65.22	63.79

SOFTQE vs. Traditional Knowledge Distillation. Our method distills the high-dimensional *representation* of the teacher model, as opposed to teacher’s *predictions*, as in traditional knowledge distillation. In Table 6, we compare our method to a variant in which we compute MSE *only over the scalar-valued scores* produced by the Q2D teacher as our distillation loss. This results in reduced performance, as the score-only distillation model underperforms the DPR baseline, indicating that the teacher’s predictions alone do not provide sufficient supervision for estimating the nuanced information contained in the high-dimensional expanded query representations.

5 Related Work

Document Expansion. Doc2Query [10] attempts to resolve vocabulary mismatch by expanding *documents* with natural language queries whose answers

are likely to exist within the document. Document expansion is advantageous because it can be conducted entirely offline during indexing and combined with learned sparse retrieval methods [8,16,18] to leverage both neural supervision and efficient inverted index algorithms. However, document expansion techniques can significantly increase the size of the index, and must be applied to the entire corpus each time the expansion method is changed, which might be too costly for corpora containing billions of documents.

Knowledge Distillation. Knowledge distillation [12] methods use the predictions of large *teacher* models to improve the performance of smaller, more practical *student* models. Knowledge distillation is ubiquitous, and has been used to improve dense dual-encoder retrievers via distillation from a cross-encoder [20]. SOFTQE is a form of *indirect* knowledge distillation, wherein a student encoder targets the continuous representations of an architecturally-equivalent teacher model whose discrete, natural language *inputs* have been augmented by an LLM. Recent methods such as Alpaca [24] and Vicuna [3] use generations of superior LLMs to improve the performance of smaller, instruction-following models, but these methods imitate "teacher" LLMs by using their outputs as training data directly.

6 Conclusion

We present SOFTQE, a technique to align query-encoder representations with the representations of queries expanded by LLMs. Empirical evaluations demonstrate improvements across several retrieval benchmarks and models, and suggest that SOFTQE improves generalization to new domains, as made evident by zero-shot evaluations on BEIR datasets. Importantly, this improvement comes without increasing the cost or latency of dense retrieval at runtime compared to other single vector dual-encoder methods, because an LLM is not required at time of inference. Future work might consider improved prompting strategies, or applying LLM-based supervision to higher-capacity retrieval methods like ColBERTv2 [23]. To the best of our knowledge, SOFTQE is the first attempt to distill strong representations through natural language generation, and we hope that this will inspire efficient solutions to new tasks in the future.

A Additional BEIR Results

Table 7: nDCG@10 for the remaining BEIR tasks; *: our reproduction (not tested for significance). Q2D was not evaluated on these datasets.

Method	Signal 1m	Trec-News	Quora	NQ	Fifa	Arguana
DPR*	24.15	35.21	84.05	43.72	24.47	28.80
+ SOFTQE	22.18	36.43	84.32	44.19	25.07	31.11
Method	Scidocs	BioASQ	HotpotQA	Climate Fever	Fever	Avg.
DPR*	11.79	25.09	49.58	17.20	65.36	37.22
+ SOFTQE	11.35	27.05	48.36	18.40	67.38	37.80

References

1. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al.: Language models are few-shot learners. ArXiv **abs/2005.14165** (2020), <https://api.semanticscholar.org/CorpusID:218971783>
2. Campos, D.F., Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L., Mitra, B.: Ms marco: A human generated machine reading comprehension dataset. ArXiv **abs/1611.09268** (2016), <https://api.semanticscholar.org/CorpusID:1289517>
3. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (March 2023), <https://lmsys.org/blog/2023-03-30-vicuna/>
4. Chowdhery, A., Narang, S., Devlin, J., et al.: Palm: Scaling language modeling with pathways. ArXiv **abs/2204.02311** (2022), <https://api.semanticscholar.org/CorpusID:247951931>
5. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.F., Voorhees, E.M.: Overview of the trec 2019 deep learning track. ArXiv **abs/2003.07820** (2020), <https://api.semanticscholar.org/CorpusID:253234683>
6. Craswell, N., Mitra, B., Yilmaz, E., Campos, D.F., Voorhees, E.M.: Overview of the trec 2020 deep learning track. ArXiv **abs/2102.07662** (2021), <https://api.semanticscholar.org/CorpusID:212737158>
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). <https://doi.org/10.18653/v1/N19-1423>, <https://aclanthology.org/N19-1423>
8. Formal, T., Piwowarski, B., Clinchant, S.: Splade: Sparse lexical and expansion model for first stage ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 2288–2292. SIGIR '21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3404835.3463098>, <https://doi.org/10.1145/3404835.3463098>
9. Gao, L., Ma, X., Lin, J., Callan, J.: Precise zero-shot dense retrieval without relevance labels. ArXiv **abs/2212.10496** (2022), <https://api.semanticscholar.org/CorpusID:254877046>
10. Gospodinov, M., MacAvaney, S., Macdonald, C.: Doc2query-: When less is more. In: European Conference on Information Retrieval (2023), <https://api.semanticscholar.org/CorpusID:255545874>
11. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.W.: Realm: Retrieval-augmented language model pre-training. ArXiv **abs/2002.08909** (2020), <https://api.semanticscholar.org/CorpusID:211204736>
12. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. ArXiv **abs/1503.02531** (2015), <https://api.semanticscholar.org/CorpusID:7200347>
13. Jagerman, R., Zhuang, H., Qin, Z., Wang, X., Bendersky, M.: Query expansion by prompting large language models. ArXiv **abs/2305.03653** (2023), <https://arxiv.org/pdf/2305.03653.pdf>

14. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L.Y., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering. In: Conference on Empirical Methods in Natural Language Processing (2020), <https://api.semanticscholar.org/CorpusID:215737187>
15. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 120–127. SIGIR '01, Association for Computing Machinery, New York, NY, USA (2001). <https://doi.org/10.1145/383952.383972>, <https://doi.org/10.1145/383952.383972>
16. Lin, J., Ma, X.: A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques (2021)
17. Lv, Y., Zhai, C.: Positional language models for information retrieval. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 299–306. SIGIR '09, Association for Computing Machinery, New York, NY, USA (2009). <https://doi.org/10.1145/1571941.1571994>, <https://doi.org/10.1145/1571941.1571994>
18. Mallia, A., Khattab, O., Suel, T., Tonello, N.: Learning passage impacts for inverted indexes. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1723–1727. SIGIR '21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3404835.3463030>, <https://doi.org/10.1145/3404835.3463030>
19. Mao, Y., He, P., Liu, X., Shen, Y., Gao, J., Han, J., Chen, W.: Generation-augmented retrieval for open-domain question answering. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4089–4100. Association for Computational Linguistics, Online (Aug 2021). <https://doi.org/10.18653/v1/2021.acl-long.316>, <https://aclanthology.org/2021.acl-long.316>
20. Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W.X., Dong, D., Wu, H., Wang, H.: 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. pp. 5835–5847. Association for Computational Linguistics, Online (Jun 2021). <https://doi.org/10.18653/v1/2021.naacl-main.466>, <https://aclanthology.org/2021.naacl-main.466>
21. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.* **3**(4), 333–389 (apr 2009). <https://doi.org/10.1561/15000000019>, <https://doi.org/10.1561/15000000019>
22. Rocchio, J.J.: *Relevance Feedback in Information Retrieval*, p. 1. Prentice Hall, Englewood, Cliffs, New Jersey (1971), http://www.is.informatik.uni-duisburg.de/bib/docs/Rocchio_71.html
23. Santhanam, K., Khattab, O., Saad-Falcon, J., Potts, C., Zaharia, M.: Colbertv2: Effective and efficient retrieval via lightweight late interaction. *CoRR abs/2112.01488* (2021), <https://arxiv.org/abs/2112.01488>
24. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca (2023)
25. Thakur, N., Reimers, N., Ruckl'e, A., Srivastava, A., Gurevych, I.: Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. *ArXiv abs/2104.08663* (2021), <https://api.semanticscholar.org/CorpusID:233296016>

26. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. ArXiv **abs/2302.13971** (2023), <https://api.semanticscholar.org/CorpusID:257219404>
27. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6000–6010. NIPS’17, Curran Associates Inc., Red Hook, NY, USA (2017)
28. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., Wei, F.: Simlm: Pre-training with representation bottleneck for dense passage retrieval. ArXiv **abs/2207.02578** (2022), <https://api.semanticscholar.org/CorpusID:250311114>
29. Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., Wei, F.: Text embeddings by weakly-supervised contrastive pre-training. ArXiv **abs/2212.03533** (2022), <https://api.semanticscholar.org/CorpusID:254366618>
30. Wang, L., Yang, N., Wei, F.: Query2doc: Query expansion with large language models. ArXiv **abs/2303.07678** (2023), <https://api.semanticscholar.org/CorpusID:257505063>
31. Wei, J., Wang, X., Schuurmans, D., Bosma, M., hsin Chi, E.H., Xia, F., Le, Q., Zhou, D.: Chain of thought prompting elicits reasoning in large language models. ArXiv **abs/2201.11903** (2022), <https://api.semanticscholar.org/CorpusID:246411621>