To Interpolate or not to Interpolate:
PRF, Dense and Sparse Retrievers

Hang Li*
The University of
Queensland
Brisbane, Australia
hang.li@uq.edu.au

Shuai Wang*
The University of
Queensland
Brisbane, Australia
shuai.wang2@uq.edu.au

Shengyao Zhuang
The University of
Queensland
Brisbane, Australia
s.zhuang@uq.edu.au

Ahmed Mourad
The University of
Queensland
Brisbane, Australia
a.mourad@uq.edu.au

Xueguang Ma
University of Waterloo
Waterloo, Canada
x93ma@uwaterloo.ca

Jimmy Lin
University of Waterloo
Waterloo, Canada
jimmylin@uwaterloo.ca

Guido Zuccon
The University of
Queensland
Brisbane, Australia
g.zuccon@uq.edu.au

ABSTRACT

Current pre-trained language model approaches to information retrieval can be broadly divided into two categories: sparse retrievers (to which belong also non-neural approaches such as bag-of-words methods, e.g., BM25) and dense retrievers. Each of these categories appears to capture different characteristics of relevance. Previous work has investigated how relevance signals from sparse retrievers could be combined with those from dense retrievers via interpolation. Such interpolation would generally lead to higher retrieval effectiveness.

In this paper we consider the problem of combining the relevance signals from sparse and dense retrievers in the context of Pseudo Relevance Feedback (PRF). This context poses two key challenges: (1) When should interpolation occur: before, after, or both before and after the PRF process? (2) Which sparse representation should be considered: a zero-shot bag-of-words model (BM25), or a learnt sparse representation? To answer these questions we perform a thorough empirical evaluation considering an effective and scalable neural PRF approach (Vector-PRF), three effective dense retrievers (ANCE, TCTv2, DistillBERT), and one state-of-the-art learnt sparse retriever (uniCOIL). The empirical findings from our experiments suggest that, regardless of sparse representation and dense retriever, interpolation both before and after PRF achieves the highest effectiveness across most datasets and metrics.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

---

*Both authors contributed equally to the paper.

KEYWORDS

Pseudo-Relevance Feedback, Dense and sparse retrieval interpolation, Neural IR

1 INTRODUCTION

Traditional unsupervised (bag-of-words – BOWs) sparse retrieval models, such as BM25, use exact term matching to retrieve relevant results from the collection. Recent studies have shown that these models are more likely to retrieve results that partially match the query, i.e., with low relevance labels [26]. Although unsupervised sparse models often fail to rank the most relevant results at the top, they often offer high recall. Combined with high efficiency, unsupervised bag-of-words sparse retrieval models like BM25 are still widely used within information retrieval pipelines, often as the initial retrieval stage of a more complex setup. To further enhance precision and push highly relevant results to the top, transformer-based dense retrievers (short for learned dense representations) strike a good balance between effectiveness and efficiency compared to traditional unsupervised sparse models and transformer-based deep language model re-rankers [10, 17, 18, 23, 24, 28, 30]. Dense retrievers utilise dual BERT-style encoders to encode queries and passages separately [16]; this allows the pre-encoding of passages into embeddings at indexing time and their offline storage. During query time, the query embeddings can be efficiently computed "on-the-fly" [32], and relevance estimations measured with a simple similarity calculation. Thus, it becomes feasible to perform retrieval over the entire collection using deep language models with efficiency comparable to traditional unsupervised sparse models, but with much higher effectiveness. While dense retrievers are very effective at encoding passages characterised by high relevance labels (i.e. highly relevant passages), they are less effective at identifying passages of lower relevance value [26]. On the other hand, learned

sparse models [2, 5, 6, 8, 9, 14, 19, 22, 31], also strike a good balance between effectiveness and efficiency compared to traditional unsupervised sparse models and transformer-based deep language model re-rankers [21]. They use transformer-based language models to learn term weights, and achieve comparable effectiveness to dense retrievers.

Interestingly, recent studies have found that interpolating sparse and dense retrieval [1] results can further enhance retrieval effectiveness [14, 17, 26], suggesting that both groups of retrievers tend to retrieve different relevant signals [14, 16]. An aspect that is still unclear in this context is what the contribution of PRF is with respect to the interpolation of sparse and dense models: this is the focus of our paper.

In this paper, we are interested in investigating the interpolation of dense and sparse retrieval results within the context of Pseudo-Relevance Feedback (PRF). Specifically, we adapt the interpolation approach on top of a recently proposed PRF method called Vector-PRF (VPRF) [12] that is designed for dense retrievers. This PRF method conducts the first round of dense retrieval to get the top retrieved passages' dense vectors and then uses these passage vectors to improve the original query's dense representation, which is then used to perform the second round of dense retrieval. In our experiments, we consider combining dense and sparse retrieval interpolation with VPRF in three different settings: **Before Vector-PRF (Pre-PRF), after Vector-PRF (Post-PRF)** and **both (Both-PRF)**. For Pre-PRF interpolation, we interpolate the sparse retriever with the dense retriever in the first round of retrieval, then apply Vector-PRF to generate the new query representation, and perform a second retrieval. For Post-PRF interpolation, we keep the two rounds of VPRF retrieval unchanged, but perform the sparse interpolation to the second round retrieval results. For Both-PRF, we apply interpolation in both the first round of retrieval and on the final results of VPRF. The research questions that we aim to address in this paper are:

RQ1: When is it better to do interpolation? **Before Vector-PRF (Pre-PRF), after Vector-PRF (Post-PRF) or both (Both-PRF)**?

RQ2: For sparse retrievers, which representation is more effective, unsupervised (BOWs) or learned? To address this research question, we consider BM25 and uniCOIL [14], respectively.

2 RELATED WORK

There are two lines of research that are related to our work. The first line of research investigates the integration of PRF with dense retrievers. Li et al. [12] proposed a simple PRF method called Vector-PRF which adapted the classic Rocchio PRF method [25] used on bag-of-words representations, to dense retrievers in a zero-shot manner. Vector-PRF has been shown to improve effectiveness, at additional minimal efficiency expense. We adopt this method in our paper. Wang et al. [27] proposed a more complex model that uses a clustering technique to model the PRF signals; this is in turn applied to the ColBERT dense retriever [11]. However, the improvements achieved by this method come at the cost of efficiency. Yu et al. [29], on the other hand, proposed the ANCE-PRF model that requires the training of a new query encoder based on the original ANCE [28]

---

query encoder. ANCE-PRF achieved significant improvements over ANCE. However, due to the input limit of the BERT-style model (512 tokens [7]), ANCE-PRF is limited in the amount of feedback it can consider: experimentally, $k = 5$ is the maximum PRF depth for MS MARCO.

The second line of research regards the interpolation of sparse retrieval models and deep language models to further boost effectiveness, especially in terms of recall. Wang et al. [26] investigated the interpolation of BM25 and dense retrievers. Their findings suggest that dense retrievers are highly effective in encoding strong relevance signals, but they are not as effective when dealing with weak relevance signals. The interpolation of BM25 and dense retrievers is able to make up for each other's weaknesses: this interpolation can significantly improve the effectiveness of dense retrievers. Furthermore, Lin and Ma [14], Lin et al. [18] and Arabzadeh et al. [1] also investigated different approaches to combine learned sparse retrieval results with dense retrieval results to improve retrieval effectiveness, and significant improvements are recorded from their experiments. Importantly, according to Wang et al. [26], dense retrievers are not so good at dealing with weak relevance signals. Therefore, PRF approaches based on dense retrievers might also inherit this limitation. To the best of our knowledge, there is no previous study that has examined the interpolation of sparse and dense retrievers within the PRF framework.

3 METHODS

Next, we introduce our method for interpolating sparse retriever and dense retrievers in the context of PRF, and in particular of the Vector-PRF approach [12]. In this paper, we adopt the same interpolation strategy used by Wang et al. [26]: a linear interpolation between sparse retriever scores and dense retriever scores, as shown in equation 1.

$$s(p) = \lambda \hat{s}_{Sparse}(p) + (1 - \lambda)s_{Dense}(p) \tag{1}$$

that is, the score of a passage is the linear interpolation of the sparse retriever score of the passage and the score of the same passage from the dense model, modulated by a parameter λ, which controls the contribution of the sparse retriever score to the final score of the passage. In our experiments, Sparse refers to BM25 or uniCOIL, Dense refers to any dense retrieval model we use. This interpolation mechanism is a simple yet effective approach to "help" the dense retrievers capture the passages' weak relevance signals.

As mentioned, for PRF, we used the Rocchio Vector-PRF approach of Li et al. [12]. The way we do interpolation with PRF can be categorized into three different types; we discuss each type with detail in the following subsections.

3.1 Pre-PRF Interpolation

The PRF process often involves two rounds of retrieval [12]. The first retrieval round is to generate the initial results for preparing PRF feedback candidates. After getting the initial retrieval results, top-k passages for each query from these results are selected as PRF feedback passages and are used to modify the original query representations. Therefore, the interpolation can be performed at either round of retrieval.

---

[1] Also known as hybrid models.

For Pre-PRF zero-shot interpolation, we perform the sparse retriever interpolation with the first round of dense retrieval results, then we apply PRF with the interpolated results to generate the new query representations for the second round of retrieval. After the interpolation, the ranking of the passages in the results are likely to be different, affecting the PRF's performance.

3.2 Post-PRF Interpolation

Other than applying interpolation before the PRF to the initially retrieved results, we also apply interpolation to the results after the second round of retrieval with PRF query representations. In this approach, the initial retrieval results are directly used for generating PRF query representations, then the PRF queries are used to perform a second round of retrieval. After the second round of retrieval, the results are then interpolated with the sparse retriever's results to obtain the final results list.

3.3 Both-PRF interpolation

Finally, Both-PRF performs interpolation before and after PRF. To perform Both-PRF, we firstly interpolate the sparse retriever's results with the dense retriever's results, then we apply PRF with the interpolated results to generate the new query representations for second round of retrieval. Then results from the second round of retrieval are again interpolated with the sparse retriever's results to generate the final results list.

4 EXPERIMENTAL SETUP

To investigate the interpolation of sparse retrievers with dense retriever PRF approaches, we devise a number of empirical experiments aimed at investigating: 1) the impact of interpolation on different dense retriever PRF approaches; 2) the impact of interpolating sparse retrievers before/after/both the PRF; 3) the impact of interpolating on different sparse retrievers, unsupervised (BOWs) or learned.

Datasets. For all of our experiments, we use the TREC Deep Learning Track passage retrieval task 2019 [3] (DL19) and 2020 [4] (DL20). DL19 contains 43 judged queries, while DL20 contains 54 judged queries. The relevance judgement levels for both datasets range from 0 (not relevant) to 3 (highly relevant). We treat passages with relevance label 1 as not relevant when we compute the binary relevance metrics (i.e., MAP, Recall). The passage collection in our experiments is the MS MARCO Passage Ranking Dataset [20], which is a benchmark English dataset for ad-hoc retrieval that contains ≈8.8 million passages. The average judgements per query for DL19 and DL20 are 215.3 and 210.9, whereas the MS MARCO Passage Ranking Dataset only has ≈1 judgement per query.

Baselines. We include:

- ANCE: First stage dense retriever [28]. We use the model implemented in Pyserini[2] [15] for inference;
- Vector-PRF (VPRF): A simple Rocchio PRF approach based on dense retrievers [12]. We use the model implemented in Pyserini[3] [15];

- TCT ColBERT V2 HN+ (TCTv2): A BERT-style distilled dense retriever learned from ColBERT with reduced query/passage embedding dimensions [18];
- TCT ColBERT V2 HN+ VPRF (TCTv2+VPRF): The application of the Rocchio VPRF from Li et al. [12] on top of TCT ColBERT V2 HN+ dense retriever. This model is also made available by the authors in Pyserini[3] [15];
- DistilBERT KD TASB (DBB): A DistilBERT-style dense retriever with balanced topic aware sampling training strategy [10]. We use the model implemented in Pyserini[4] [15] by the original authors;
- DistilBERT KD TASB + VPRF (DBB+VPRF): The application of the Rocchio VPRF from Li et al. [12] on top of DistilBERT KD TASB dense retriever. This model is implemented by Li et al. [12] and available to use in Pyserini[3] [15].

In our experiments, we use the parameters $\alpha = 0.4$, $\beta = 0.6$, and PRF depth = 3 for Rocchio VPRF, following the settings recommended by Li et al. [12]. In terms of the interpolation parameter λ, we use $\lambda = 0.5$ for all experiments. For generating the BM25 runs to be used for interpolation, we use the BM25 implementation provided by Pyserini [15] and we use the default parameter values for k_1 and b within Pyserini. For generating uniCOIL runs, we also use the pre-built uniCOIL index provided by Pyserini.

Evaluation Measures. We use the official evaluation metrics from DL19 and DL20: nDCG@10 and Recall@1000. We also report MAP as a complementary metric.

5 RESULTS

Next, we examine the results of our empirical investigation; for this we follow the research questions we put forward in Section 1. The main results are presented in Table 1. We note that all the models used in our experiments are provided by the original authors and we do not train a new model. The training of a new model may lead to variations in the effectiveness results because of e.g., the stochastic nature of the weight initialisation of the models. However, we observe that Li et al. [13] have shown that the differences observed empirically in terms of model effectiveness across re-training of the same model are minor and not statistically significant. The use of the original model checkpoint for each model then appears a fair choice in the context of our work, and the variability that the retraining of these models would cause seems a non-critical direction to investigate in this short paper.

5.1 RQ1: When is it better to do interpolation? Pre-PRF, Post-PRF or Both-PRF?

To answer the first research question, we perform Pre-PRF interpolation, Post-PRF interpolation, and Both-PRF interpolation. We do this across two sparse retrievers (one neural, uniCOIL, and one not, BM25), and three dense retrivers (ANCE, TCTV2 and DistillBERT).

We first discuss the results among each dense retriever with only PRF or only interpolation involved. In this case, the interpolation with uniCOIL always gives the highest effectiveness compared to using PRF or using the dense retriever along: this is regardless of the dense retriever of choice and dataset. However, when BM25

Dataset			DL19			DL20		
Sparse Model	Dense Model	PRF-Interpolation	MAP	nDCG@10	Recall@1000	MAP	nDCG@10	Recall@1000
	ANCE		0.3710	0.6452	0.7554	0.4076	0.6458	0.7764
	ANCE-VPRF		0.3831	0.6512	0.7611	0.4118	0.6479	0.7800
BM25	ANCE	No-PRF	0.4264	0.6888	**0.8607**	0.4067	0.6264	0.8643
		Pre-PRF	0.3868	0.6620	$0.7638^\wedge$	**0.4175**	**0.6548**	$0.7911^\wedge$
		Post-PRF	$0.4322^\wedge$	0.6885	$0.8604^\star$	0.4140	0.6353	$\mathbf{0.8666}^{\wedge\star}$
		Both-PRF	$\mathbf{0.4345}^\wedge$	**0.6895**	**0.8607**	$0.4100^\wedge$	0.6274	$0.8662^\wedge$
uniCOIL	ANCE	No-PRF	0.4587	0.6908	0.8459	0.4644	0.6984	0.8482
		Pre-PRF	$0.3857^\wedge$	$0.6602^\wedge$	$0.7584^\wedge$	$0.4154^\wedge$	$0.6545^\wedge$	$0.7892^\wedge$
		Post-PRF	$0.4617^\star$	$0.6910^\star$	$\mathbf{0.8495}^\star$	$0.4693^{\wedge\star}$	$\mathbf{0.7024}^\star$	$\mathbf{0.8500}^\star$
		Both-PRF	**0.4622**	**0.6917**	0.8458	$\mathbf{0.4699}^\wedge$	0.7012	$\mathbf{0.8500}^\wedge$
	TCTv2		0.4469	0.7204	0.8261	0.4754	0.6882	0.8429
	TCTv2-VPRF		0.4626	0.7219	0.8377	0.4863	0.6952	0.8462
BM25	TCTv2	No-PRF	0.4474	0.7067	0.8753	0.4468	0.6696	0.8872
		Pre-PRF	**0.4698**	**0.7268**	0.8436	$\mathbf{0.4879}^\wedge$	**0.6987**	$0.8455^\wedge$
		Post-PRF	$0.4547^\wedge$	0.7045	0.8788	$0.4511^{\wedge\star}$	0.6683	$\mathbf{0.8918}^{\wedge\star}$
		Both-PRF	$0.4574^\wedge$	0.7061	$\mathbf{0.8820}^\wedge$	0.4490	0.6659	$0.8902^\wedge$
uniCOIL	TCTv2	No-PRF	0.4771	0.7245	0.8557	0.4895	0.718	0.8683
		Pre-PRF	0.4659	0.7246	0.8415	0.4897	0.7083	$0.8484^\wedge$
		Post-PRF	$0.4826^{\wedge\star}$	**0.7347**	$0.8663^\wedge$	**0.4926**	0.7184	$0.8718^\star$
		Both-PRF	**0.4831**	0.7268	$\mathbf{0.8606}^\wedge$	$0.4920^\wedge$	**0.7190**	$\mathbf{0.8723}^\wedge$
	DBB		0.4590	0.7210	0.8406	0.4698	0.6854	0.8727
	DBB-VPRF		0.4667	0.7285	0.8479	0.4804	**0.7027**	0.8767
BM25	DBB	No-PRF	0.4584	0.6993	0.8622	0.4417	0.6491	0.8948
		Pre-PRF	**0.4711**	**0.7319**	0.8526	$\mathbf{0.4812}^\wedge$	$\mathbf{0.6968}^\wedge$	0.8755
		Post-PRF	$0.4652^\wedge$	0.7053	0.8711	$0.4509^\star$	$0.6522^\star$	0.8974
		Both-PRF	$0.4665^\wedge$	0.7019	**0.8720**	0.4442	0.6474	**0.8977**
uniCOIL	DBB	No-PRF	0.4779	0.7288	0.8542	0.4859	0.7041	0.8808
		Pre-PRF	0.4726	0.7255	0.8468	0.4815	0.7002	0.8760
		Post-PRF	0.4822	0.7298	0.8658	**0.4915**	**0.7097**	**0.8836**
		Both-PRF	$\mathbf{0.4858}^\wedge$	**0.7334**	**0.8669**	$0.4880^\wedge$	0.7062	0.8834

Table 1: The results of all baseline runs and No-PRF, Pre-PRF, Post-PRF and Both-PRF interpolation runs of all models with the Rocchio Vector PRF approach proposed by Li et al. [12]. Statistical significance tests are conducted between Pre- and Post-PRF models, significant difference are marked with ★. We also tested the statistical significance with Pre-PRF interpolation versus No-PRF interpolation, and Post-PRF interpolation versus No-PRF interpolation, and Both-PRF interpolation versus no-PRF interpolation, significant difference are marked with $\overline{\wedge}$. Best performance among each base sparse model is marked as Bold.

is used for interpolation, the interpolated results are lower than those achieved by PRF for MAP and nDCG (but not recall@1000), regardless of dense retriever and dataset.

We now consider interpolation with BM25. Unlike the findings obtained by comparing the use of the dense retriever with either only interpolation or only PRF, better effectiveness is achieved when both PRF and interpolation with BM25 are used. The effectiveness of the Post-PRF interpolation condition is very close to that of Both-PRF interpolation; however, Both-PRF interpolation achieves the highest effectiveness most of the times and across all dense retrievers and datasets. However, the difference between Post-PRF and Both-PRF interpolation is not significant, and Both-PRF interpolation requires some extra computations compared to Post-PRF. Pre-PRF interpolation tends to enhance early precision, while Post-PRF interpolation tends to enhance deep recall. Overall, however, Pre-PRF interpolation is the least effective of the three interpolation conditions, regardless of dense retriever, metric and dataset.

We now consider interpolation with uniCOIL. Here also we observe that PRF with uniCOIL interpolation can consistently achieve higher effectiveness than using the dense retrievers or sparse retrievers alone. Furthermore, we also found that using Post-PRF interpolation or Both-PRF interpolation outperforms results obtained with interpolation but not PRF, regardless of dense retriever and dataset. However, if Pre-PRF interpolation is used, then effectiveness is often similar or lower than when interpolation is used but no PRF. Indeed, Both-PRF can achieve the highest effectiveness most of the times.

Overall, the condition with interpolation both before and after PRF (Both-PRF) showcased often high effectiveness regardless of dense and sparse retriever. The condition with interpolation performed after PRF (Post-PRF) showcased high effectiveness only with the uniCOIL sparse retriever, but not with BM25.

5.2 RQ2: Which sparse retriever is more effective, unsupervised (BOWs) or learned?

In our experiments we considered BM25 and uniCOIL as sparse retriever. The uniCOIL method is a typical neural sparse retriever trained with contrastive loss that uses BERT to predict impact scores for both query tokens and document tokens. Before performing predictions, uniCOIL uses docTquery-T5 [22] to expand all passages in the collection by adding potential relevant tokens that are not in the original passages. BM25 instead is a traditional BOWs sparse retriever, where the representation is not learnt. We now consider which of these sparse representation is best to combine with the signal from the dense retrievers; we do this bot in the context of PRF (Pre-PRF, Post-PRF, and Both-PRF interpolation) and when not considering PRF (No-PRF).

The results show that the use of uniCOIL guarantees an increase in MAP and nDCG@10 compared to BM25, but a lower recall@1000. This result is valid across all dense retrievers, datasets and for No-PRF, Post-PRF and Both-PRF interpolation conditions.

For the Pre-PRF interpolation method, however, no general trend is found when comparing the two sparse models. On DL2019, uniCOIL consistently underperforms BM25 in the pre-PRF interpolation condition, with the only exception of MAP when DBB is used as dense retriever. On DL2020, uniCOIL always outperforms BM25 when the dense retrievers are TCTV2 or DBB, but underperforms BM25 when ANCE is used as dense retriever.

Overall, we found that when performing PRF and interpolating with the Post-PRF and Both-PRF conditions, an unsupervised BOWs sparse retriever leads generally to high recall, while a neural, trained sparse retriever achieves higher MAP and nDCG@10. However, the Pre-PRF condition shows no stable trends in terms of which sparse retriever to use.

6 CONCLUSION

Previous work has argued that sparse and dense retrievers encode different relevance characteristics of a document [16, 26]. Because of this, methods for the combination of these two signals have emerged; the simplest method being the interpolation of the scores originating from a sparse and a dense retriever [26].

In this paper, we conducted an extensive investigation to study the effect of interpolation between sparse and dense retrievers in the context of Pseudo Relevance Feedback for dense retrievers, and in particular for the scalable and effective Vector-PRF method [12]. With this respect, we studied applying BM25 and uniCOIL as sparse retrievers, along with three dense retrievers: ANCE, TCTV2 and DistillBERT. In terms of when to interpolate sparse and dense signals, we considered doing this before PRF, after PRF, and both before and after PRF.

The empirical results show that interpolation often can boost retrieval effectiveness, regardless of the choice of sparse and dense retrievers. Among the choices of when to interpolate, we found that interpolating both before and after the PRF process is the condition that most often lead to substantial gains.

REFERENCES

[1] Negar Arabzadeh, Xinyi Yan, and Charles LA Clarke. 2021. Predicting Efficiency/Effectiveness Trade-offs for Dense vs. Sparse Retrieval Strategy Selection. *arXiv preprint arXiv:2109.10739* (2021).

[2] Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. SparTerm: Learning term-based sparse representation for fast text retrieval. *arXiv preprint arXiv:2010.00768* (2020).

[3] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 Deep Learning Track. In *Proceedings of the Twenty-Ninth Text REtrieval Conference (NIST Special Publication)*. National Institute of Standards and Technology (NIST).

[4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2021. Overview of the TREC 2020 Deep Learning Track. In *Text REtrieval Conference, TREC*.

[5] Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval. *arXiv preprint arXiv:1910.10687* (2019).

[6] Zhuyun Dai and Jamie Callan. 2020. Context-aware document term weighting for ad-hoc search. In *Proceedings of The Web Conference 2020*. 1897–1907.

[7] 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*. 4171–4186.

[8] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086* (2021).

[9] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. 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*. 2288–2292.

[10] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 113–122.

[11] 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*. 39–48.

[12] Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2021. Pseudo Relevance Feedback with Deep Language Models and Dense Retrievers: Successes and Pitfalls. *arXiv preprint arXiv:2108.11044* (2021).

[13] Hang Li, Shengyao Zhuang, Ahemed Mourad, Xueguang Ma, Jimmy Lin, and Guido Zuccon. 2022. Improving Query Representations for Dense Retrieval with Pseudo Relevance Feedback: A Reproducibility Study. In *Proceedings of The 44th European Conference on Information Retrieval, ECIR 2022*.

[14] Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807* (2021).

[15] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. *Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations*. Association for Computing Machinery, New York, NY, USA, 2356–2362.

[16] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained transformers for text ranking: Bert and beyond. *arXiv preprint arXiv:2010.06467* (2020).

[17] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386* (2020).

[18] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. 163–173.

[19] A Mallia, O Khattab, T Suel, and N Tonellotto. 2021. Learning Passage Impacts for Inverted Indexes. In *44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2021*. Association for Computing Machinery, Inc, 1723–1727.

[20] 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. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (CEUR Workshop Proceedings, Vol. 1773)*. CEUR-WS.org.

[21] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[22] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery.

[23] 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*. 5835–5847.

[24] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A Joint Training Method for Dense Passage Retrieval and Passage Re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2825–2835.

[25] J.J. Rocchio. 1971. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*. 313–323.

[26] Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. BERT-based Dense Retrievers Require Interpolation with BM25 for Effective Passage Retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 317–324.

[27] Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2021. Pseudo-Relevance Feedback for Multiple Representation Dense Retrieval. *arXiv preprint arXiv:2106.11251* (2021).

[28] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations*.

[29] HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving Query Representations for Dense Retrieval with Pseudo Relevance Feedback. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM.

[30] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. RepBERT: Contextualized text embeddings for first-stage retrieval. *arXiv preprint arXiv:2006.15498* (2020).

[31] Shengyao Zhuang and Guido Zuccon. 2021. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513* (2021).

[32] Shengyao Zhuang and Guido Zuccon. 2021. TILDE: Term Independent Likelihood moDEl for Passage Re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery.