Improving Query Representations for Dense Retrieval with Pseudo Relevance Feedback: A Reproducibility Study

Hang Li[1], Shengyao Zhuang[1], Ahmed Mourad[1], Xueguang Ma[2], Jimmy Lin[2], and Guido Zuccon[1]

[1] The University of Queensland, St. Lucia, Australia
{hang.li,s.zhuang,a.mourad,g.zuccon}@uq.edu.au
[2] University of Waterloo, Waterloo, Canada
{x93ma,jimmylin}@uwaterloo.ca

Abstract. Pseudo-Relevance Feedback (PRF) utilises the relevance signals from the top-k passages from the first round of retrieval to perform a second round of retrieval aiming to improve search effectiveness. A recent research direction has been the study and development of PRF methods for deep language model based rankers, and in particular in the context of dense retrievers. Dense retrievers provide a trade off between effectiveness, which is often reduced compared to more complex neural rankers, and query latency, which also is reduced making the retrieval pipeline more efficient. The introduction of PRF methods for dense retrievers has been motivated as an attempt to further improve their effectiveness.
In this paper, we reproduce and study a recent method for PRF with dense retrievers, called ANCE-PRF. This method concatenates the query text and that of the top-k feedback passages to form a new query input, which is then encoded into a dense representation using a newly trained query encoder based on the original dense retriever used for the first round of retrieval. While the method can potentially be applied to any of the existing dense retrievers, prior work has studied it only in the context of the ANCE dense retriever.
We study the reproducibility of ANCE-PRF in terms of both its training (encoding of the PRF signal) and inference (ranking) steps. We further extend the empirical analysis provided in the original work to investigate the effect of the hyper-parameters that govern the training process and the robustness of the method across these different settings. Finally, we contribute a study of the generalisability of the ANCE-PRF method when dense retrievers other than ANCE are used for the first round of retrieval and for encoding the PRF signal.

Keywords: Pseudo Relevance Feedback · Dense Retrievers · Query Representations

1 Introduction

Pseudo-Relevance Feedback (PRF) is a retrieval technique which assumes that the top-k results from the first round of retrieval are relevant. PRF, therefore,

uses this signal to improve the query representation for a second round of retrieval (or re-ranking) in a bid to obtain higher search effectiveness. PRF has been extensively studied and applied to bag-of-word retrieval models; representative techniques are Rocchio [23], KL expansion [32, 18], RM3 [17] and other relevance models [9]. It is well accepted that PRF tends to improve search effectiveness, and strong bag-of-words baselines often rely on PRF (e.g., BM25+RM3 is a typical baseline combination).

Aside from its use with bag-of-words models, PRF has been recently studied in the context of Transformer[25]-based deep language models such as BERT [4] and RoBERTa [16]; examples of such Transformer-based rankers include cross-encoder architectures such as monoBERT [21]. These deep language models have been very effective for ranking although, compared to bag-of-words methods, they often require substantially more computational power and are characterised by high query latencies. Their effectiveness can be further improved by PRF – but this is at the cost of even higher query latencies, rendering the use of PRF on top of BERT-based rankers like monoBERT practically unfeasible [12].

Dense retrievers (DRs) have been proposed as alternatives to the expensive BERT-based rankers [33, 7, 28, 8]. DRs also rely on deep language models like BERT; however instead of training a cross-encoder to encode a query and document[3] pair at the same time, it relies on a bi-encoder architecture where queries and documents are encoded separately. This separation in the encoding allows us to pre-compute document representations (which is computationally expensive for large collections) at indexing time, thus leaving only the encoding of the query and the matching between the query and document representations to be performed at query time. Dense retrievers provide a trade off between effectiveness and efficiency: while they are often less effective than the cross-encoder methods, DRs are more efficient (lower query latency). PRF with DRs then becomes suddenly more interesting than when applied to cross-encoders: PRF could provide effectiveness boosts while the additional computational cost imposed by the feedback, infeasible when considering cross-encoders, may be feasible in the context of DRs. This research direction has therefore attracted increasing interest [12, 30, 27].

In this paper, we consider a specific method for PRF with DRs: the ANCE-PRF method [30]. This method uses the ANCE dense retriever [28] to perform a first round of retrieval for a given query. Then, the text of the original query is concatenated with that from the top-k documents retrieved by ANCE. The output is a new text query, which is encoded using the purposely trained ANCE-PRF encoder to obtain a new dense query representation that is in turn used for computing the match with the document dense representations to determine a ranking for the query. The ANCE-PRF encoder is trained using a straightforward training procedure with negative sampling strategy based on the original ANCE

[3] In this paper, we use 'document' and 'passage' interchangeably. Our experiments and the methods considered are in the context of the passage retrieval task. However, the methods can generalise to deal with documents, at the cost of the development of strategies for managing the often large size of documents compared to passages [30].

model, expect that the input to the ANCE-PRF encoder is the concatenation of the query and the relevance signal (top-k documents), rather than just the query (or just the document) as in ANCE.

Given the ANCE-PRF method, we aim to replicate the initial study by Yu et al. [30] in terms of both the training of the ANCE-PRF encoder and its use for retrieval. In addition, we also aim to further extend that work by considering the factors that affect the training of the ANCE-PRF encoder, i.e., the hyper-parameters of the model, and studying their effect on the model performance and therefore its robustness across hyper-parameters settings. We also study the generalisability of the strategy underlying ANCE-PRF to other DRs. In doing so, we develop and publicly release a codebase that implements Yu et al.'s method, along with trained checkpoints of the method for both ANCE and other DRs.

2 Related Work

Pseudo-Relevance Feedback (PRF) is a classic query expansion method that aims to mitigate the mismatch between query intent and query representation [1, 26], by modifying the original query with the top-k initially retrieved results. Typical PRF approaches such as Rocchio [23], query-regularized mixture model [24], KL expansion [18, 32], RM3 [17], relevance models [9], and relevance-feedback matrix factorization [31] are well studied. However, most of the existing studies of PRF methods are applied on top of bag-of-words retrieval models.

With the emergence of transformer-based [25] models, many researchers have been looking into how to integrate PRF with deep language models. Zheng et al. [34] presented a BERT-based [4] PRF model, BERT-QE, which splits the PRF documents into smaller chunks and utilises the BERT model to identify the most relevant PRF document chunks and uses these chunks as PRF signals. Li et al. [11] proposed a neural PRF approach that uses a feed-forward neural network model to aggregate the query and feedback document relevance scores and provide the target document's relevance score. Yu et al. [29] utilises graph transformers to capture the PRF signals from the initial retrieved results; and Wang et al. [26] proposed a clustering method to gather the relevance signals from PRF documents. These methods show remarkable improvements, but the efficiency is significantly affected, e.g., BERT-QE inference requires 11.01× more computations than BERT alone, making these models computationally infeasible for many practical applications.

Recently, *dense retrievers* [28, 15, 7, 8, 6] have been attracting a lot of attention from researchers. These models, which often utilise a BERT-based dual-encoder to encode queries and passages into a shared embedding space, have shown great effectiveness and efficiency in various tasks and datasets. However, most of the existing studies are focusing on different training methods, especially negative sampling techniques [28, 5, 10]. Most of these models encode either the query or the document to a single embedding vector [28, 15, 6], which fits perfectly to many vector-based PRF methods.

In recent research, because of the nature of dense retrievers that use embedding vectors to represent query and document, different methods have been studied to integrate pseudo relevance information into dense retrievers. Li et al. [12] investigated two simple approaches, Average and Rocchio, to utilise PRF information in dense retrievers (ANCE [28] and RepBERT [33]) without introducing new neural models or further training. According to the results, both models achieved superior effectiveness with these two simple approaches without hurting the efficiency significantly compared to the original models, which shows the viability of integrating PRF signals in deep language models. A more recent work attempts to utilise the pattern learning ability of transformer models to leverage the PRF signals. Yu et al. [30] replaced the query encoder in ANCE [28] model by training a new query encoder, which takes the original query text and the PRF documents text together as the new query, based on the original ANCE model as the initial training checkpoint, without changing the document encoder. However, it has several major limitations: 1) for each different PRF depths, it requires training a new query encoder; 2) the input length for the query encoder is limited, which means the PRF depth is limited; 3) the new query encoder is trained on top of the ANCE query encoder, which means for different datasets, different ANCE models need to be trained first, making this new approach hard to be generalised.

3 Improving Query Representations for Dense Retrievers with Pseudo Relevance Feedback

In this section, we briefly describe the ANCE-PRF method [30], which extends ANCE [28] to integrate the PRF signal from the top-k documents to be encoded in combination with the query to form a new query representation.

In ANCE, the score of a document d for a query q is computed by separately encoding q and d using the RoBERTa [16] pre-trained deep language model, and then calculating the inner product between the resulting dense representations:

$$f_{\texttt{ANCE}}(q, d) = \texttt{ANCE}^q(\langle s\rangle q\langle /s\rangle) \cdot \texttt{ANCE}^d(\langle s\rangle d\langle /s\rangle) \tag{1}$$

where $\texttt{ANCE}^q$ and $\texttt{ANCE}^d$ represent the query and the document encoders, respectively, and $\langle s\rangle$ and $\langle /s\rangle$ represent the `[CLS]` and `[SEP]` tokens in ANCE. Both encoders use the final layer of the $\langle s\rangle$ token embedding as the query and document dense representations. In ANCE, the document embeddings are precomputed offline and stored in an index, while the query embeddings are encoded at inference (query) time [28]. For fine-tuning Equation 1, ANCE adopts noisy contrastive estimation loss and employs a negative sampling strategy where negative samples are dynamically retrieved from an asynchronously updated ANCE document index [28].

ANCE-PRF uses a similar schema to score documents for retrieval:

$$f_{\texttt{ANCE-PRF}}(q, d) = \texttt{ANCE}^{prf}(\langle s\rangle q\langle /s\rangle d_1\langle /s\rangle ...d_k\langle /s\rangle) \cdot \texttt{ANCE}^d(\langle s\rangle d\langle /s\rangle) \tag{2}$$

where $\texttt{ANCE}^{prf}$ is the newly trained PRF query encoder and $\langle s\rangle q\langle /s\rangle d_1\langle /s\rangle ... d_k\langle /s\rangle$ is the text concatenation of the original query q with the feedback documents $d_1, d_2, ..., d_k$ (in addition to `[CLS]` and separator tokens). We denote q^{prf} as the query embedding generated through PRF by $\texttt{ANCE}^{prf}$.

For the training of the PRF query encoder ($\texttt{ANCE}^{prf}$), ANCE-PRF uses the standard noisy contrastive estimation loss:

$$\mathcal{L} = -log\frac{exp(q^{prf} \cdot d^{+})}{exp(q^{prf} \cdot d^{+}) + \Sigma_{d^{-} \in D^{-}} exp(q^{prf} \cdot d^{-})} \tag{3}$$

where d^{+} represents a relevant document for the query, d^{-} represents an irrelevant document (obtained from the negative sampling technique). During the training process, the ANCE-PRF model uses the document embeddings from the original ANCE model. Therefore, the document embeddings remain unchanged in the ANCE-PRF model: it is only the query embedding that changes into the PRF query embedding q^{prf}.

Intuitively, ANCE-PRF should provide increases in search effectiveness because the newly trained ANCE-PRF query encoder learns to extract relevant information for the query from the PRF documents using the Transformer attention mechanism [25]. After training, the ANCE-PRF query encoder would then pay more attention to the relevant tokens in the PRF documents, while ignoring the irrelevant tokens from this signal. Although Yu et al. [30] do not report the query latency of ANCE-PRF, this should be approximately twice that of the original ANCE model.

4 Experimental Settings

4.1 Datasets

The datasets used in the original work of Yu et al. [30] are TREC DL 2019 [2], TREC DL 2020 [3], DL Hard [19], and MS MARCO Passage Ranking V1 [20]. These datasets are based on the same corpus provided by MS MARCO Passage Ranking V1, which has ∼8.8M passages in total. Note that for TREC DL 2019/2020 queries, each query has multiple judgements on a relevance scale from 0 to 3, while MS MARCO Passage Ranking V1 only has an average of one judgement per query with binary relevance, either 0 or 1.

The original paper used the training split from MS MARCO Passage Ranking V1 for training ANCE-PRF, which includes ∼530K queries. The trained models are evaluated on TREC DL 2019 (43 judged queries), DL 2020 (54 judged queries), DL HARD, and MS MARCO Passage Ranking V1 Dev set (6,980 queries). For direct comparison with the ANCE-PRF model, we follow the same process except for evaluation on TREC DL HARD (the results on this dataset for other dense retrievers considered in this paper are not publicly available).

4.2 Models

The original work by Yu et al. [30] only considers ANCE as the initial dense retriever. To validate their hypothesis that their PRF method can be generalised to other dense retrievers, we consider two recently published dense retrievers that achieve higher performance than ANCE: TCT ColBERT V2 HN+ [15] and DistilBERT KD TASB [6]. These two dense retrievers are different from ANCE with respect to the training process, and the inference is slightly different from each other. We refer the reader to the original papers for further details. The output of these three dense retrievers are all embedding vectors that represent either the query or the document based on the input. The indexes for all three models are pre-computed and stored offline.

$$q^{prf}_{TCT} = \texttt{TCT}^{prf}(\texttt{[CLS]}\ [Q]\ q\texttt{[SEP]}d_1\texttt{[SEP]}...d_k\texttt{[MASK]} * 512) \tag{4}$$

TCT ColBERT V2 HN+ uses a BERT encoder, as shown in Equation 4, to encode queries and documents, where $\texttt{TCT}^{prf}$ represents the new PRF query encoder based on the TCT ColBERT V2 HN+ query encoder. The input requires a `[CLS]` token, as well as a `[Q]` in text as prepend to the actual query text, then the PRF document texts are separated by the `[SEP]` token, then use the `[MASK]` token to pad the gap if the input is smaller than the max input size of the model, which is 512 for BERT-based models [4].

$$f_{\texttt{TCT-PRF}}(q, d) = q^{prf}_{TCT} \cdot \texttt{TCT}^{d}(\texttt{[CLS]}\ [D]\ d) \tag{5}$$

For retrieval, TCT ColBERT V2 HN+ uses a scoring function, as shown in Equation 5, where $\texttt{TCT}^{d}$ represents the document encoder, and the input document text is prepended with the `[CLS]` token and `[D]` in the text.

$$q^{prf}_{DBERT} = \texttt{DBERT}^{prf}(\texttt{[CLS]}q\texttt{[SEP]}d_1\texttt{[SEP]}...d_k\texttt{[SEP]}) \tag{6}$$

$$f_{\texttt{DBERT-PRF}}(q, d) = q^{prf}_{DBERT} \cdot \texttt{DBERT}^{d}(\texttt{[CLS]}d\texttt{[SEP]}) \tag{7}$$

On the other hand, DistilBERT KD TASB uses a DistilBERT encoder, as shown in Equation 6, and a scoring functions for retrieval, as shown in Equation 7. Similar to TCT ColBERT V2 HN+, except the input is a standard BERT input with the `[CLS]` token as prepend and the `[SEP]` token as separators to separate the PRF documents for both PRF query encoding and retrieval.

4.3 Inference and Training

Inference To reproduce the ANCE-PRF results, the authors have provided us with a model checkpoint of PRF depth 3. Since there is no inference code available from the original authors, we utilise the open source IR toolkit Pyserini[4] [14], which has already implemented the ANCE dense retriever, by introducing a second round of ANCE retrieval with the ANCE-PRF model checkpoint. During

[4] https://github.com/castorini/pyserini

the inference time, the document index is the same for both the first round ANCE retrieval and the second round ANCE-PRF retrieval. The only difference in this process is that the initial retrieval uses the ANCE query encoder, while the second retrieval uses the ANCE-PRF query encoder.

Training The authors have not released the training code. To replicate the ANCE-PRF training process, we utilise the open source dense retriever training toolkit Tevatron[5]. According to the original paper [30], all hyperparameters used in ANCE-PRF training are the same as ANCE training, and the ANCE-PRF query encoder is initialised from ANCE FirstP model[6] [28]. Although some of the parameters are still not reported in the original paper, we managed to replicate the same model as ANCE-PRF with $k = 3$ by adjusting different training settings.

We also experimented with two more effective dense retrievers, TCT ColBERT V2 HN+ [15] and DistilBERT KD TASB [6] to investigate the generalisability of the ANCE-PRF model. Therefore, we adopted the same hyperparameters from these two models and trained with the same settings as ANCE-PRF.

All models in our experiments are trained on two Tesla V100 SMX2 32GB GPUs. In the original paper, the ANCE-PRF model is trained with per device batch size 4 and gradient accumulation step 8 for 450K steps, which is equivalent to per device batch size 32 for ∼56K steps, therefore, in our training experiments, we use 10 epochs, which is roughly ∼80K steps.

4.4 Evaluation Metrics

The official evaluation metric for MS MARCO Passage Ranking V1 dataset is MRR@10 [20], for TREC DL 2019 and 2020 are nDCG@10, Recall@1000 [2, 3]. For the Recall@1000 evaluation metric on TREC DL 2019 and 2020, the judgements are binarized at relevance point 2 according to the official guideline. Besides the official evaluation metrics, the authors in the original work [30] also use HOLE@10 as an additional evaluation metric to measure the unjudged fraction of top 10 retrieved documents [28], to reflect the coverage of the pooled labels on these dense retrieval systems. However, in our experiments, we opt to keep the official evaluation metrics only, for the sake of comparison with other models and baselines. Statistical significance differences between models results are measured using two-tailed paired t-tests.

4.5 Research Questions

In this work, we aim to address the following research questions along with the reproducibility and replication [7] of the original method from Yu et al. [30]:

[5] https://github.com/texttron/tevatron

[6] https://github.com/microsoft/ANCE

[7] We use the terminology of reproducibility and replication in compliance with the definitions provided by ACM: https://www.acm.org/publications/policies/artifact-review-badging

RQ1: What is the possibility of reproducing the inference results of ANCE-PRF given only a checkpoint of the trained model provided by the original authors?

RQ2: The training process is governed by a number of hyper-parameters and choices, importantly including learning rate, optimizer, and negative sampling technique settings. Given the insufficient details in the original study, it is reasonable to expect that researchers attempting to replicate the ANCE-PRF method may set these parameters to values different from those in the original study. We are then interested to study: what is the impact of ANCE-PRF training hyper-parameters on the effectiveness of the method, and in particular if this is robust to different hyper-parameter settings?

RQ3: The PRF strategy underlying ANCE-PRF can be adapted to other dense retrievers as observed by Yu et al. [30], but not empirically validated. The original ANCE-PRF model is only trained with ANCE [28] as the initial dense retriever. We are then interested to investigate: do the improvements observed for ANCE-PRF generalise to other dense retrievers, such as the two more effective models, TCT ColBERT V2 HP+ [15] and DistilBERT KD TASB [6]?

5 Results and Analysis

5.1 RQ1: Reproduce ANCE-PRF Inference

A benefit of the transformer-based neural models nowadays is its easy reproducibility; the results can be reproduced easily by using the model checkpoint. Therefore, with the PRF 3 checkpoint provided by the authors, we tried to reproduce the same results reported in the original paper; the outcomes are shown in Table 1. During the reproducibility process, we found that the ANCE-PRF model is sensitive to uppercase or lowercase letters. For the original queries used in all three datasets in this experiment, no uppercase letters existed, therefore this detail is omitted from the paper. But from our reproducibility experiments, uppercase letters exist in the corpus, and the token ids and their associated tokens embeddings are different with different cases of the same word. Therefore, for PRF queries, after concatenating the PRF documents to the original query text, the new PRF queries contain uppercase letters and leads to different tokens after tokenization, and resulting in different performance compared to what is reported in the original paper. On the other hand, if we set the tokenizer to do lowercase at inference time, then we can get the same results as the original paper. Hence, we successfully reproduced the ANCE-PRF model for inferencing by using the checkpoint provided by the authors.

To answer RQ1, we confirmed that it is possible to reproduce the same results with the model checkpoint, however one key detail that was missing in the paper is the lowercase process to the PRF query. We make our ANCE-PRF inference implementation publicly available in Pyserini toolkit [8] so that practitioners can

[8] https://github.com/castorini/pyserini/blob/master/docs/experiments-ance-prf.md

Table 1. The reproduced results with the provided ANCE-PRF 3 checkpoint after inference. **unhandled** represents results with the PRF query containing both uppercase and lowercase letters. **do lowercase** indicates the results with PRF query converted to lowercase when tokenized. **ANCE-PRF 3** shows results in original paper.

Datasets	MS MARCO			TREC DL 2019		TREC DL 2020	
	MRR@10	nDCG@10	R@1000	nDCG@10	R@1000	nDCG@10	R@1000
ANCE [28]	0.330	0.388	0.959	0.648	0.755	0.646	0.776
ANCE-PRF 3 [30]	0.344	0.401	0.959	0.681	0.791	0.695	0.815
unhandled	0.342	0.399	0.960	0.678	0.792	0.674	0.794
do lowercase	0.344	0.402	0.960	0.681	0.791	0.695	0.815

easily reproduce the same results in the original paper with the author provided model checkpoint.

5.2 RQ2: Replicate ANCE-PRF Training

In this section, we would like to see if we can reproduce the model by following the training settings provided in the original paper. However, some details were missing and we had to consult with the authors to identify their exact settings. After clarifying the training parameters, we used the same setting to train our own ANCE-PRF model; the results are shown in Table 2.

Table 2. The replicated results with the trained ANCE-PRF 3 checkpoint after inference. **ANCE-PRF 3** shows the results from the original paper. **ANCE** represents the results from the original ANCE model. **Replicated** is the results from our replicated ANCE-PRF model.

Datasets	MS MARCO			TREC DL 2019		TREC DL 2020	
	MRR@10	nDCG@10	R@1000	nDCG@10	R@1000	nDCG@10	R@1000
ANCE [28]	0.330	0.388	0.959	0.648	0.755	0.646	0.776
ANCE-PRF 3 [30]	0.344	0.401	0.959	0.681	0.791	0.695	0.815
Replicated	0.347	0.405	0.963	0.672	0.794	0.701	0.814

From the results, once their setting was replicated, we obtained results that are close to those reported and with similar trends, at times worse, other times better, but never statistically significantly different from the results reported in the original paper. The minor differences between the two results can be potentially explained by random neuron drop out during training and the random seed while sampling the hard negatives from the initially retrieved ANCE results.

In the original study, the authors reported that they were using all hyperparameters from ANCE [28] training, and all models are trained on two RTX

Table 3. Initial represents the results by re-initialising the linear head layer. **Inherit** represents the results by inheriting the linear head layer from ANCE. **In-Batch** represents the results by using in-batch negatives. **No In-Batch** represents the results by not using in-batch negatives. **1e-6** represents the results by using 1e-6 as the learning rate. **1e-5** represents the results by using 1e-5 as learning rate.

Datasets	MS MARCO	TREC DL 2019		TREC DL 2020	
	MRR@10	nDCG@10	R@1000	nDCG@10	R@1000
Inherit/No In-Batch/1e-5	0.347	0.672	0.794	0.701	0.814
Inherit/No In-Batch/1e-6	0.335	0.680	0.798	0.678	0.814
Inherit/In-Batch/1e-5	0.347	0.672	0.797	0.678	0.807
Initial/No In-Batch/1e-5	0.313	0.631	0.710	0.644	0.772

2080 Ti GPUs with per-GPU batch size 4 and gradient accumulation step 8 for 450K steps. However, some parameters are still unclear in ANCE training. We trained the ANCE-PRF model with two Tesla-V100 SMX2 32GB GPUs with per-GPU batch size 32, learning rate $1e-5$, no in-batch negatives or cross batch negatives, and no gradient accumulation steps for 10 epoches. The reason why we chose to remove the gradient accumulation step setting is because we are using GPUs with larger memory. In the original settings, 450K steps with gradient accumulation step 8 and per-GPU batch size 4 is the same as 56,250 steps with per-GPU batch size 32. Therefore, in our training process, we used 10 training epoches, which is equivalent to 83,240 steps in total, and it is already more than the steps used in the original settings.

The optimizer in the training process for the ANCE-PRF model reported in the original study is the LAMB optimizer, which we overlooked at first, instead we used the AdamW optimizer which might lead to unsuccessful replication.

A common practice for training new models based on an existing model is to re-initialise the linear head layer and train from scratch while keeping the model body. We followed this practice at first, but it appears that the ANCE-PRF model is trained with everything inherited from the the ANCE model, including the embedding head and normalisation (linear head layer). So without keeping the linear head layer from ANCE, our trained ANCE-PRF is significantly worse than the original ANCE-PRF model, as shown in Table 3.

Recent more effective models such as RocketQA [22], uniCOIL [13] show that in-batch negatives help the model learn and achieve better performance. However, in our experiments, in-batch negatives do not help to improve the model performance, as shown in Table 3, the difference between using and not using in-batch negatives is not statistically significant though.

Learning rate also plays an important part in the training process. We have experimented with two different learning rates, 1e-5 and 1e-6; the results are shown in Table 3. Using a larger learning rate tends to improve MRR@10 for the MS MARCO dataset, while a smaller learning rate tends to improve nDCG@10

Table 4. Results of training two more effective dense retrievers with the same training process as ANCE-PRF. For direct comparisons, we trained the models with PRF depth of value 3. † represents statically significant difference based on a two-tailed paired t-test.

Datasets	MS MARCO			TREC DL 2019		TREC DL 2020	
	MRR@10	nDCG@10	R@1000	nDCG@10	R@1000	nDCG@10	R@1000
ANCE [28]	0.330	0.388	0.959	0.648	0.755	0.646	0.776
ANCE-PRF 3 [30]	0.344	0.401	0.959	0.681	0.791	0.695	0.815
TCT ColBERT V2 HN+	0.359	0.420	0.970	0.720	0.826	0.688	0.843
TCT ColBERT V2 HN+ PRF 3	0.357	0.418	0.971†	0.741	0.852†	0.712†	0.840
DistilBERT KD TASB	0.344	0.407	0.977	0.721	0.841	0.685	0.873
DistilBERT KD TASB PRF 3	0.348	0.411†	0.974	0.736	0.857†	0.698†	0.866

and R@1000 in TREC DL 2019. However, only MRR@10 is statistically significantly different.

To answer RQ2, some hyperparameters, such as learning rate, number of negatives, type of negatives, and optimizer, are crucial for reproducing the model checkpoint.

5.3 RQ3: Generalisability of ANCE-PRF Beyond ANCE

After successfully reproducing the ANCE-PRF model inference and replicating the training, we investigated if integrating this PRF strategy with other popular and more effective dense retrievers will provide some improvements in effectiveness when compared to the dense retrievers results without PRF. However, this improvement is of a smaller magnitude than that observed for ANCE, which can be observed from Table 4. This may be due to: (1) the best hyper-parameter settings for ANCE-PRF may not be adequate to generalise to other dense retrievers, and different settings may lead other dense retrievers to obtain larger improvements; this speaks to the limited robustness of ANCE-PRF's training strategy. (2) The dense retrievers we consider, TCT ColBERT V2 HN+[15] and DistilBERT KD TASB [6] are more effective than ANCE. The limited improvement then may be due to the fact that it is easier to improve a weaker model (ANCE) than it is to improve a more effective one.

To answer RQ3, we find that applying the same training strategy as ANCE-PRF to other more effective dense retrievers only achieves a smaller magnitude of improvement. Hence, the ANCE-PRF method may not generalize to other dense retrievers or may require specific hyper-parameter tuning.

6 Conclusion

In this paper we considered the ANCE-PRF model proposed by Yu et al. [30]. This method is the first of its kind to integrate PRF signals directly into the query encoder, without changing the document encoder or document index.

There are three research questions related to reproducing and replicating ANCE-PRF. RQ1 is aimed to address the issues when reproducing the inference results by directly adopting the model checkpoint provided by the original authors. Our experiments show that ANCE-PRF is an uncased model; it can only handle lowercase letters for queries. If the query contains uppercase letters, the ANCE-PRF model performs differently and hurts performance.

RQ2 is aimed to replicate the training process of the ANCE-PRF model by using the settings provided in the original study. However, some details are missing, which leads to unsatisfying performance of the replicated model. After consulting with the original authors and using the exact same training settings, we were able to replicate the ANCE-PRF model with insignificant differences that might be caused by the random seed in negative sampling or model initialisation. We then investigate the effects of hyper-parameters in the ANCE-PRF model training. Since some details are left out in the original study, we replicate the model by using common practice. However, in our experiments, we found that some common practice might not work in this case. For example, using the linear head layer from the ANCE model to train ANCE-PRF is significantly better than to initialise the linear head layer. In-batch negatives have been proved to be useful for training in many superior models, but in our experiment, there is no significant difference between using in-batch negatives and no in-batch negatives.

RQ3 is aimed to test the generalisability of the training method of the ANCE-PRF model. We use the same parameter settings to train the PRF model on top of TCT ColBERT V2 HN+[15] and DistilBERT KD TASB [6], two more effective dense retrievers compared to ANCE. However, the results are mixed; the improvements with PRF are of a smaller magnitude than that observed for ANCE. This may be because the best hyper-parameter settings are not suitable for all dense retrievers; to achieve better performance one may need to adjust the parameters accordingly. Another reason may be because both newly added models are more effective than ANCE; the limited improvements may be because this training method is more suitable to improve a weaker model.

The code to reproduce the training of all the models in this work is made available at https://github.com/ielab/APR.

References

1. Clinchant, S., Gaussier, E.: A theoretical analysis of pseudo-relevance feedback models. In: Proceedings of the 2013 Conference on the Theory of Information Retrieval. pp. 6–13 (2013)
2. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2019 Deep Learning Track. In: Text REtrieval Conference, TREC (2020)
3. Craswell, N., Mitra, B., Yilmaz, E., Campos, D., Voorhees, E.M.: Overview of the TREC 2020 Deep Learning Track. In: Text REtrieval Conference, TREC (2021)
4. 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. pp. 4171–4186 (2019)

5. Gao, L., Dai, Z., Chen, T., Fan, Z., Durme, B.V., Callan, J.: Complement lexical retrieval model with semantic residual embeddings. In: European Conference on Information Retrieval. pp. 146–160. Springer (2021)
6. Hofstätter, S., Lin, S.C., Yang, J.H., Lin, J., Hanbury, A.: Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 113–122. ACM (2021)
7. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense Passage Retrieval for Open-Domain Question Answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781 (2020)
8. Khattab, O., Zaharia, M.: 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. pp. 39–48 (2020)
9. 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. pp. 120–127 (2001)
10. Lee, K., Chang, M.W., Toutanova, K.: Latent Retrieval for Weakly Supervised Open Domain Question Answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 6086–6096 (2019)
11. Li, C., Sun, Y., He, B., Wang, L., Hui, K., Yates, A., Sun, L., Xu, J.: NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4482–4491 (2018)
12. Li, H., Mourad, A., Zhuang, S., Koopman, B., Zuccon, G.: Pseudo Relevance Feedback with Deep Language Models and Dense Retrievers: Successes and Pitfalls. arXiv preprint arXiv:2108.11044 (2021)
13. Lin, J., Ma, X.: A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques. arXiv preprint arXiv:2106.14807 (2021)
14. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: An easy-to-use Python toolkit to support replicable IR research with sparse and dense representations. arXiv preprint arXiv:2102.10073 (2021)
15. Lin, S.C., Yang, J.H., Lin, J.: Distilling dense representations for ranking using tightly-coupled teachers. arXiv preprint arXiv:2010.11386 (2020)
16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692 (2019)
17. Lv, Y., Zhai, C.: A Comparative Study of Methods For Estimating Query Language Models with Pseudo Feedback. In: Proceedings of the 18th ACM ACM International Conference on Information and Knowledge Management. pp. 1895–1898 (2009)
18. Lv, Y., Zhai, C.: Revisiting the Divergence Minimization Feedback Model. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management. pp. 1863–1866 (2014)
19. Mackie, I., Dalton, J., Yates, A.: How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (2021)

20. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A Human Generated Machine Reading Comprehension Dataset. In: Workshop on Cognitive Computing at NIPS (2016)
21. Nogueira, R., Cho, K.: Passage Re-ranking with BERT. arXiv preprint arXiv:1901.04085 (2019)
22. 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 (2021)
23. Rocchio, J.: Relevance Feedback in Information Retrieval. In: The SMART Retrieval System - Experiments in Automatic Document Processing. pp. 313–323 (1971)
24. Tao, T., Zhai, C.: Regularized Estimation of Mixture Models for Robust Pseudo-Relevance Feedback. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 162–169. SIGIR '06, Association for Computing Machinery (2006)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is All You Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 6000–6010 (2017)
26. Wang, J., Pan, M., He, T., Huang, X., Wang, X., Tu, X.: A pseudo-relevance feedback framework combining relevance matching and semantic matching for information retrieval. Information Processing & Management **57**(6), 102342 (2020)
27. Wang, X., Macdonald, C., Tonellotto, N., Ounis, I.: Pseudo-Relevance Feedback for Multiple Representation Dense Retrieval. arXiv preprint arXiv:2106.11251 (2021)
28. Xiong, L., Xiong, C., Li, Y., Tang, K.F., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In: International Conference on Learning Representations (2020)
29. Yu, H., Dai, Z., Callan, J.: PGT: Pseudo Relevance Feedback Using a Graph-Based Transformer. In: European Conference on Information Retrieval (2021)
30. Yu, H., Xiong, C., Callan, J.: Improving Query Representations for Dense Retrieval with Pseudo Relevance Feedback. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management. ACM (2021)
31. Zamani, H., Dadashkarimi, J., Shakery, A., Croft, W.B.: Pseudo-Relevance Feedback Based on Matrix Factorization. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 1483–1492 (2016)
32. Zhai, C., Lafferty, J.: Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In: Proceedings of the 10th ACM International Conference on Information and Knowledge Management. pp. 403–410 (2001)
33. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: RepBERT: Contextualized text embeddings for first-stage retrieval. arXiv preprint arXiv:2006.15498 (2020)
34. Zheng, Z., Hui, K., He, B., Han, X., Sun, L., Yates, A.: BERT-QE: Contextualized Query Expansion for Document Re-ranking. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings. pp. 4718–4728 (2020)