Effective Contrastive Weighting for Dense Query Expansion

Xiao Wang[‡], **Sean MacAvaney**[†], **Craig Macdonald**[†], **Iadh Ounis**[†],
School of Computing Science, University of Glasgow, UK
[‡]x.wang.8@research.gla.ac.uk
[†]{sean.macavaney, craig.macdonald, iadh.ounis}@glasgow.ac.uk

Abstract

Verbatim queries submitted to search engines often do not sufficiently describe the user's search intent. Pseudo-relevance feedback (PRF) techniques, which modify a query's representation using the top-ranked documents, have been shown to overcome such inadequacies and improve retrieval effectiveness for both lexical methods (e.g., BM25) and dense methods (e.g., ANCE, ColBERT). For instance, the recent ColBERT-PRF approach heuristically chooses new embeddings to add to the query representation using the inverse document frequency (IDF) of the underlying tokens. However, this heuristic potentially ignores the valuable context encoded by the embeddings. In this work, we present a contrastive solution that learns to select the most useful embeddings for expansion. More specifically, a deep language model-based contrastive weighting model, called CWPRF, is trained to learn to discriminate between relevant and non-relevant documents for semantic search. Our experimental results show that our contrastive weighting model can aid to select useful expansion embeddings and outperform various baselines. In particular, CWPRF can improve nDCG@10 by upto to 4.1% compared to an existing PRF approach for ColBERT while maintaining its efficiency.

1 Introduction

When using search engines, users frequently enter queries that insufficiently express their desired intent. For instance, a user who issues the query *georgia run off elections* may indeed be looking for details about a specific electoral procedure in the US state of Georgia. For search algorithms that rely on lexical matching, such as BM25, this can result in a *lexical gap*, since relevant documents may just as easily use different terms (e.g., GA and 2nd-round). Pseudo-Relevance Feedback (PRF) techniques are often employed to overcome such a lexical gap. Indeed, classical PRF techniques, such as RM3 (Abdul-Jaleel et al., 2004), have been widely used to enrich the user's query with terms selected from the initial retrieval top-ranked documents, i.e. the pseudo-relevance feedback set (Amati and Van Rijsbergen, 2002; Roy et al., 2016; Cao et al., 2008). This *expanded* query is capable of overcoming the lexical gap if the pseudo-relevance feedback documents are relevant and additional related terms can be identified (for instance adding GA to the query). However, there is a risk that the added terms *drift* the intent of the query (for instance, adding terms such as Tbilisi that relate to the *country* of Georgia rather than the US state).

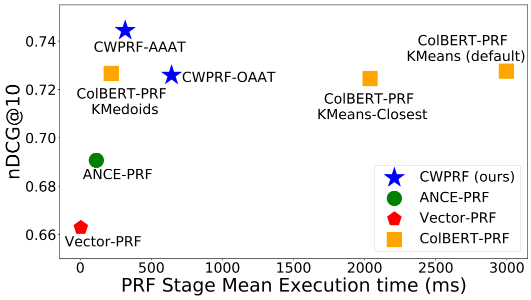


Figure 1: Effectiveness (nDCG@10) versus dense PRF stage mean execution time on the TREC 2019 query set.

An alternative approach for overcoming the lexical gap is to perform *semantic search* over learned embedded documents (*single representation*, e.g., ANCE (Karpukhin et al., 2020)) or tokens (*multiple representations*, e.g., ColBERT (Khattab and Zaharia, 2020)). Such *dense retrieval* approaches enable queries to retrieve documents that do not necessarily contain the query terms. However, the encoded query vectors might still not adequately express the user's desired intent. Indeed, several recent works have shown that implementing PRF techniques within the dense retrieval paradigm – such as ANCE-PRF (Yu et al., 2021), Vector-PRF (Li et al., 2023) and ColBERT-PRF (Wang et al., 2021, 2022b) – can further improve retrieval effectiveness. ColBERT-PRF has been shown to be

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12688–12704
July 9-14, 2023 ©2023 Association for Computational Linguistics

more effective than Vector-PRF and ANCE-PRF variants applied on various dense retrieval models.

A key limitation of ColBERT-PRF is that it relies on clustering and inverse document frequency (IDF) statistics for identifying the expansion embeddings — both of which are heuristics. This approach ignores valuable context present in the embeddings, e.g., for the *georgia run off elections* query, effectiveness might be improved by adding an embedding for 'US', however, this would not likely be selected due to its low IDF (indeed, 'us' is also a pronoun, and is often included in stopword lists). Moreover, there is no direct connection between the expansion embeddings selected by the heuristic and the semantic search algorithm itself.

To overcome these problems, we propose a contrastive weighting method, called CWPRF, to select and weight the usefulness of the feedback embeddings for dense expansion. More specifically, for each feedback token, we construct a contrastive objective, where, given positive and negative documents, CWPRF is trained to assign high weights to the tokens that are semantically closer to tokens occurring the positive document than to those in the negative document. Introducing the PRF passages into the training procedure of CWPRF enables the model to take the surrounding context into account when identifying the useful tokens from the PRF passages. Meanwhile, training CWPRF with the contrastive objective allows it to learn the effective weights for expansion embeddings that are tailored for the semantic ranking task.

Figure 1 presents the trade-off between the retrieval effectiveness and the mean PRF stage execution time for a variety of existing dense PRF techniques on the TREC 2019 Deep Learning track queries, including Vector-PRF (Li et al., 2023), ANCE-PRF (Yu et al., 2021), ColBERT-PRF variants (Wang et al., 2021, 2022b) and our proposed CWPRF method. As the figure shows, the default ColBERT-PRF implementation outperforms ANCE-PRF and Vector-PRF in terms of retrieval effectiveness but requires a longer execution time. Meanwhile, our proposed CWPRF achieves the highest nDCG@10 score without requiring high computational cost.

Overall, our contributions are summarised as follows: (1) We propose CWPRF, a contrastive weighting method for dense query expansion; (2) We construct the contrastive targets and train our CWPRF model to assign high expansion weights for tokens that can discriminate the relevant documents from the non-relevant documents. Based on the predicted weights, CWPRF helps to identify useful expansion embeddings for generating refined query representations; (3) We perform an extensive empirical evaluation and demonstrate how to effectively train our CWPRF in a supervised way; (4) Experiments show that our CWPRF can achieve significantly higher retrieval effectiveness but with less execution time than the default ColBERT-PRF.

2 Preliminaries

Given a query q and a document[1] d, we employ the pre-trained ColBERT (Khattab and Zaharia, 2020) query and document encoders to encode the query and document, respectively. The ColBERT query and document encoders share weights but are distinguished by the different prepended special tokens. The ColBERT model is defined as a linear layer upon the raw token embeddings obtained from a BERT model: $\text{ColBERT} = \text{Linear}(\text{BERT}(t_1, ..t_n), m)) \in \mathbb{R}^m$, where m is typically set to 128 (Khattab and Zaharia, 2020). In particular, the input query tokens are encoded as a list of query embeddings (each of dimension m), as follows: $\phi_q = \text{ColBERT}(\texttt{[CLS]}, \texttt{[Q]}, q_1, ..., q_{|q|}) \in \mathbb{R}^{32 \times m}$, where $m = 128$ and the '[MASK]' embeddings are used to pad the input query embeddings to 32. Similarly, for a document d, we encode it into a list of document embeddings, as follows: $\phi_d = \text{ColBERT}(\texttt{[CLS]}, \texttt{[D]}, d_1, ..., d_{|d|}) \in \mathbb{R}^{|d| \times m}$.

Based on the obtained query and document embeddings, the final similarity score between a query and a document, $s(q, d)$, is given by the summation of the highest cosine similarity among the document embeddings for each query embedding:

$$s(q,d) = \textstyle\sum_{i=1}^{|q|} \text{MaxSim}(\phi_{q_i}, \phi_d) = \sum_{i=1}^{|q|} \max_{j=1}^{|d|} \phi_{q_i}^T \phi_{d_j}. \quad (1)$$

3 Contrastive Weighting for Dense PRF

This section first provides an implementation overview of CWPRF for dense query expansion in Section 3.1. It then details the contrastive weighting method and the training procedure of CWPRF in Sections 3.2 & 3.3, respectively.

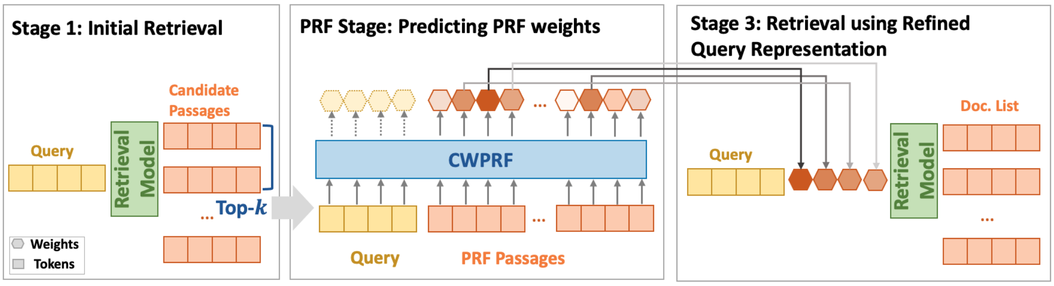

Stage 1: Initial Retrieval
Candidate Passages
Query
Retrieval Model
Top-k
Weights
Tokens
PRF Stage: Predicting PRF weights
CWPRF
Query
PRF Passages
Stage 3: Retrieval using Refined Query Representation
Query
Retrieval Model
Doc. List

Figure 2: Overview of CWPRF for dense query expansion.

3.1 CWPRF Implementation Overview

An overview of CWPRF in a multiple-representation dense expansion framework is illustrated in Figure 2, where three stages are presented: (1) initial retrieval, (2) predicting the PRF tokens weights and (3) retrieval with the refined query representation. We note that the first and the third stages of this framework are shared with ColBERT-PRF (Wang et al., 2021, 2022b).

In the initial retrieval stage, we obtain a result list in response to the original user's query q. The top f_p documents are employed as the pseudo-relevance feedback documents. Then, as input for our trained CWPRF model, we append the PRF passages to the query. The model outputs weights for each query token as well as for the feedback tokens. Finally, according to these produced weights, we identify f_e feedback tokens with high weights as our expansion tokens and append their corresponding expansion embeddings obtained from ColBERT's document encoder to the original query representation. Following conventional PRF models going back to Rocchio (Croft et al., 2010), the overall contribution of the expansion embeddings is further controlled by a hyper-parameter denoted by β. Finally, the refined query representation is reissued to the underlying dense retrieval model, i.e. ColBERT, so as to return the final document list.

The core challenge, which lies in the second PRF stage, is how to accurately predict the expansion weights for the refined query representation that can more effectively perform semantic search. We propose a novel contrastive weighting model that learns to weight each feedback token individually based on the extent it will increase the score of the relevant document w.r.t. the non-relevant one(s).

3.2 CWPRF Feedback Embedding Weighting

Building on ColBERT, and taking an initially retrieved set of pseudo-relevant feedback passages as input, the CWPRF model aims to predict the importance of each (token-level) feedback embedding in the feedback passages. This is achieved using a separate BERT model instance, which takes a list of input tokens and returns a scalar weight for each token: $\mathrm{CWPRF}(t_1...t_n) = \mathrm{Linear}(\mathrm{BERT}(t_1,..t_n),1)) \in \mathbb{R}^n$.

More specifically, given a document p in the pseudo-relevant set, which is tokenised into a sequence of PRF tokens $p_1, p_2, ..., p_{|p|}$, we employ the ColBERT encoder to obtain its embeddings: $\phi_p = \mathrm{ColBERT}(\texttt{[CLS]}, \texttt{[D]}, p_1, ..., p_{|p|}) \in \mathbb{R}^{|p| \times m}$. Then we obtain the feedback weight for each PRF token using CWPRF which takes the query representations as well as the PRF representations as input:

$$ws = \mathrm{CWPRF}\big(\overbrace{\texttt{[CLS]}, \texttt{[Q]}, q_1, q_{|q|}}^{\text{query tokens}} \overbrace{\texttt{[D]}, p_1, ..., p_{|p|}}^{\text{PRF tokens}}\big). \tag{2}$$

According to the returned importance score for each of the feedback embeddings in ϕ_p, we identify the highly important ranked embeddings as our expansion embeddings. The expansion embeddings are appended to the original query embeddings to refine the query representation. Note that the original query is included in the invocation of $\mathrm{CWPRF}(\cdot)$ – this is by design, to ensure that the CWPRF model considers the relation of the PRF tokens to the original query. However, we ignore the predicted weights of the original query; following ColBERT-PRF, the weights of the original embeddings are assumed to be unchanged. Furthermore, we apply a ReLU upon ws, to ensure that feedback weights non-negative. Finally, the score for a document can be calculated as the summation of the weighted MaxSims using the refined query representation:

$$s'(q, f_e, d) = s(q, d) + \beta \sum_{i=1}^{|f_e|} ws_i \cdot \mathrm{MaxSim}(f_{e_i}, \phi_d). \tag{3}$$

[1] We use 'document' and 'passage' interchangeably.

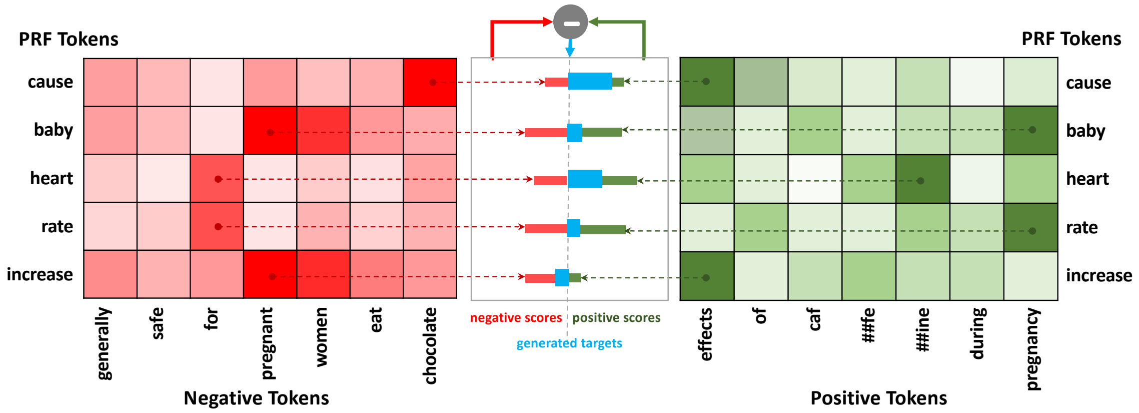


Figure 3: Target generation of CWPRF for the training query: "is a little caffeine ok during pregnancy". The target for a PRF token (blue bar) is generated by subtracting (a) the maximum negative similarity score of the PRF token interacting with the tokens from the negative passage (left-hand interaction plot) from (b) the maximum positive similarity of the PRF token interacting with the tokens from the positive passage (right-hand interaction plot).

3.3 Training CWPRF

To train $\text{CWPRF}(\cdot)$, we construct a contrastive target for each feedback token. In particular, we use a conventional training file containing triples of $\langle q, d^+, d^- \rangle$, and supplement it with PRF passages, i.e. the passages highly ranked for the original query q, which we assume to be relevant. The aim of our training objective, therefore, is to identify *which* tokens of a feedback passage p result in the positive passage being scored much higher than the negative passage, when the feedback passage is itself treated as the query. Therefore, for each feedback token, and given the positive and negative documents, CWPRF is trained to assign high weights to the tokens that are semantically closer to the tokens occurring in positive document than those in the negative document. Hence, the target for the i-th PRF token, p_i, is obtained as:

$$t(p_i) = \text{MaxSim}(p_i, d^+) - \text{MaxSim}(p_i, d^-), \tag{4}$$

where $\text{MaxSim}(.,.)$ measures the semantic similarity between representations, as per Equation (1).

The target generation process for CWPRF is illustrated in Figure 3. This figure presents the interaction matrices between a PRF document ("cause baby heart rate increase") obtained from the returned documents list in response to the query: "is a little caffeine ok during pregnancy" compared to the positive and negative document. The shading is indicative of the magnitude of dot product similarity between a PRF embedding and a document embedding, while the highest document embedding for each PRF embedding is indicated with a •. For each PRF embedding, we subtract the negative similarity from the positive similarity, resulting in an importance score for each PRF embedding. In this example, 'cause' and 'heart' are the most important tokens. These differences are used as targets for learning the CWPRF model. $p_{\text{AAAT}} = p_1^1, p_2^1, .., p^1_{|p^1|}, \texttt{[SEP]}, ..., p_1^k, p_2^k, .., p^k_{|p^k|}, \texttt{[SEP]}$. However, in common with all BERT models, $|p_{\text{AAAT}}|$ is limited to 512 tokens, so some tokens may be cut off for large feedback sets. Hence, in the OAAT training mode, each PRF document is regarded as an individual PRF sequence. The CWPRF training is then conducted for each feedback passage individually.

In-Batch Negative Sampling: In-Batch Negative (IBN) sampling is a technique that has been widely used for training effective dense retrieval models such as DPR (Karpukhin et al., 2020; Lin et al., 2020). However, it has not previously been applied for query expansion weighting. To promote the discriminative expansion embeddings and suppress the unimportant ones during our target generation, we adapt the idea of in-batch negative (IBN) sampling during the training of CWPRF. Thus, each training sample is equipped with one positive sample and $B-1$ negative samples, where B is the batch size used during training. As a consequence, the target for the i-th PRF token is obtained as:

$$t(p_i) = \text{MaxSim}(p_i, d^+) - \max_{j=1}^{|B-1|} \text{MaxSim}(p_i, d_j^-). \tag{5}$$

This ensures that the importance of each feedback embedding for ranking a positive passage is discounted by its presence in all negative passages of the batch. While IBN is commonly used for training ranking models on entire passages, our adaptation focuses instead on the token-level embedding importance.

Loss Functions: CWPRF is trained to assign weights from the target signal using the following objectives. For AAAT training, the loss is computed as follows:

$$\mathcal{L}_{\mathcal{AAAT}} = \frac{1}{N}\sum_{i=1}^{N}\left(t_{p_i} - ws_{p_i}\right)^2, \tag{6}$$

where N is the total number of tokens in the PRF sequence. For the OAAT training mode, we compute the loss for each PRF sequence and add them to obtain the total loss:

$$\mathcal{L}_{\mathcal{OAAT}} = \sum_{j=1}^{k}\left(\frac{1}{N}\sum_{i=1}^{N}(t_{p_i^j} - ws_{p_i^j})^2\right). \tag{7}$$

At the inference time, we apply CWPRF consistently with its training mode, i.e. AAAT or OAAT.

3.4 Discussion

Connection to ColBERT-PRF: Similar to ColBERT-PRF, CWPRF is implemented in the multiple representation late interaction dense retrieval paradigm. However, in contrast to ColBERT-PRF, CWPRF is a supervised approach, which is tailored for semantic search by selecting and learning the contrastive weights for the discriminate expansion embeddings. The Kendall's τ correlation between the contrastive weights learned by CWPRF and the IDF weights assigned by ColBERT-PRF is only 0.1, which indicates that CWPRF prioritises differently the feedback embeddings. Moreover, compared to ColBERT-PRF, CWPRF has advantages over ColBERT-PRF in that it can identify expansion embeddings that may have low IDF values. It can also avoid the expensive clustering and nearest neighbour lookups used by ColBERT-PRF.

Connection to Learned Sparse Models: In practice, the CWPRF model structure is similar to unexpanded learned sparse retrieval approaches (Dai and Callan, 2020; Mallia et al., 2021; Lin and Ma, 2021). Importantly, however, the learning objectives are different; learned sparse retrieval optimises for relevance directly, while CWPRF is optimised to identify and weight the most helpful query expansion embeddings.

4 Experimental Setup

Datasets: We conduct our experiments using the MS MARCO (Nguyen et al., 2016) passage ranking dataset. The corpus consists of 8.8M passages from web pages, along which are provided 0.5M training queries with sparse document relevance judgements. We employ the TREC Deep Learning track 2019 query set (43 queries with an average of 215 relevant documents per query) as our validation set and use TREC 2020 (54 queries with 211 relevance assessments per query) query set as our test set due to their dense judgements, which can provide more reliable evaluations (Carterette et al., 2006; Craswell et al., 2021). As pseudo-relevance feedback approaches are known not to show a benefit on sparsely judged documents (Amati et al., 2004), we omit the MS MARCO Dev queries. In addition, we also report the performance of CWPRF on four BEIR (Thakur et al., 2021) datasets in Appendix A.2.

We evaluate our method using the official metrics of TREC, such as nDCG@10, MAP@1000 and Recall@1000. We follow the standard practice of TREC (non-relevant = 0 or 1 and relevant = 2 or 3) for the binary-relevance based metrics (MAP and Recall). To investigate the extent that semantic matching, rather than exact token matches occurs when retrieving documents, we also report the semantic match proportion (SMP) (Wang et al., 2022a) for the ColBERT-based system. The calculation of SMP is detailed in Appendix B. For significance testing, we use the paired t-test ($p < 0.05$) and apply the Holm-Bonferroni multiple testing correction.

Experimental Implementation: Both the AAAT and OAAT training modes are trained using the MS MARCO "small" triples training set, i.e. the triplets of $\langle q, d^{+} d^{-}\rangle$. Following the settings of ColBERT (Khattab and Zaharia, 2020), we use a ColBERT checkpoint trained using the MS MARCO passage ranking training triplets for 44k batches. We employ the query encoder from the trained ColBERT model to encode the query (the maximum query length is set to 32) and the document encoder to encode the pseudo-relevance feedback documents (the maximum document length is set to 512 for the AAAT training mode and 180 for the OAAT training mode). We set the maximum length to 180 when encoding the positive and negative passages. For ease of notation, we use » to denote a retrieval pipeline, for instance `BM25 » ColBERT` indicates applying the ColBERT reranker on the results obtained from BM25. For setting the hyper-parameters of CWPRF, we use

	Systems	TREC 2019 (Validation)				TREC 2020 (Test)			
		MAP	nDCG@10	Recall	Mean-SMP	MAP	nDCG@10	Recall	Mean-SMP
Sparse	(a) BM25	0.2864	0.4795	0.7553	-	0.2930	0.4936	0.8103	-
	(b) BM25 » ColBERT	0.4597	0.6969	0.7553	0.3244	0.4721	0.6891	0.8072	0.3546
	(c) BM25+RM3	0.3108	0.5156	0.7756	-	0.3203	0.5043	0.8423	-
	(d) BM25+RM3 » ColBERT	0.4732	0.7059	0.7756	0.3404	0.4801	0.6866	0.8423	0.3560
Dense	(e) ANCE	0.3715	0.6537	0.7571	-	0.4070	0.6447	0.7737	-
	(f) ColBERT E2E	0.4310	0.6934	0.7892	0.3332	0.4648	0.6871	0.8245	0.3684
L-Sparse	(g) SPLADE-v2 » ColBERT	0.4579	0.6957	0.8723	0.3327	0.4730	0.6794	**0.8987**	0.3682
	(-) DeepImpact » ColBERT	-	0.7220	-	-	-	0.6910	-	-
	(h) DocT5Query » ColBERT	0.5009	0.7136	0.8263	0.3400	0.4733	0.6934	0.8456	0.3618
D-PRF	(i) ANCE-PRF	0.4253	0.6807	0.7912	-	0.4452	0.6948	0.8148	-
	(j) ColBERT-PRF	0.5244	0.7276	**0.8760**	0.3592	0.4904	0.6958	0.8858	0.3837
	(-) Vector-PRF	0.4151	0.6629	0.6962	-	0.4341†	0.6598†	0.7948†	-
Ours	CWPRF-AAAT	**0.5319**^{acefgi}	**0.7444**^{acefgi}	0.8596^{abefi}	0.2814	**0.5136**^{abcefgi}	**0.7246**^{abcdefgj}	0.8783^{abefi}	0.3240
	CWPRF-OAAT	0.5252^{acefgi}	0.7244^{ace}	0.8722^{abefi}	0.2923	0.5049^{abcefgi}	0.7204^{acdefg}	0.8783^{abefi}	0.3265

Table 1: Main results on both TREC 2019 and TREC 2020 queries. The superscripts ‘a-j’ denote significant improvements over the indicated baseline model. The highest effectiveness value for each metric is boldfaced. Results not available for significance testing are denoted with ‘-’. † denotes results over-fitted to the test set.

the TREC 2019 queries as our validation set; the resulting settings of $f_p = 3$, $f_e = 10$ and $\beta = 5$ are obtained, as reported later in Appendix A.1. However, we note that $f_p = 3$, $f_e = 10$ is also the recommended setting for ColBERT-PRF (Wang et al., 2021). The high β value indicates the high contribution of the CWPRF identified expansion embeddings for semantic ranking. We further provide the ablations of performing only the expansion embeddings in Appendix A.1. For both CWPRF and ColBERT-PRF, we perform 5 sets of experiments with varied random seeds for each variant and report the median results.

Compared Systems: To test the effect of CWPRF, we compare the retrieval effectiveness of a CWPRF-based retrieval system with the following 4 families of retrieval approaches: (1) *Sparse Retrieval Systems* (denoted as Sparse in Table 1): We compare with the traditional lexical retrieval models, namely BM25 and BM25+RM3 (Abdul-Jaleel et al., 2004), and both with and without the ColBERT reranker, namely BM25 » ColBERT and BM25+RM3 » ColBERT models; (2) *Dense Retrieval Systems* (denoted as Dense): We compare with both single-representation and multiple-representation dense retrieval models, namely ANCE (Xiong et al., 2021) and ColBERT (Khattab and Zaharia, 2020); (3) *Learned Sparse Retrieval Systems* (denoted as L-Sparse): We compare with SPLADE-v2 (Formal et al., 2022), DeepImpact (Mallia et al., 2021) and DocT5Query (Nogueira et al., 2020), which are reranked using ColBERT; (4) *Dense PRF models* (denoted as D-PRF): we compare with the ANCE-PRF (Yu et al., 2021), Vector-PRF (Li et al., 2023) and ColBERT-PRF (Wang et al., 2021) models. We compare our proposed CWPRF model with the more effective ColBERT-PRF Ranker model using the default KMeans clustering (Wang et al., 2021), rather than comparing with the Reranker. Moreover, when measuring the efficiency of CWPRF, we also compare with the recently proposed variants of ColBERT-PRF, which avoid costly ANN lookups when calculating IDF values for embeddings: KMedoids and KMeans-Closest (Wang et al., 2022b).

5 Results

This section studies the effectiveness as well as the efficiency performance of CWPRF in Section 5.1. The effects of the various training strategies are investigated in Section 5.2. We also provide qualitative analysis of CWPRF in Appendix A.3 and a breakdown performance of CWPRF according to various query types in Appendix A.4.

5.1 Main Results

Effectiveness: To evaluate the effectiveness of implementing the CWPRF model in a dense pseudo-relevance feedback framework, we compare CWPRF with various families of baselines in Table 1.

Among the variants of CWPRF, we observe that when comparing the CWPRF-AAAT and CWPRF-OAAT models (the bottom block), CWPRF-AAAT, which is trained with all PRF passages processed as a single sequence, consistently obtains a higher performance than CWPRF-OAAT, where the PRF sequences are considered individually. This suggests that AAAT provides more relevant context than OAAT for the CWPRF model.

Next, we compare our CWPRF model with other baseline models. Firstly, we observe that the CWPRF models significantly outperform the sparse retrieval models and exhibit marked improvements over sparse-retrieval reranked with the ColBERT reranker. When compared with dense retrieval models, the CWPRF models significantly outperform both types of dense retrieval models. In particular CWPRF exhibits 7.4% (TREC 2019 queries) and 5.5% (TREC 2020 queries) improvements in terms of nDCG@10 than the ColBERT E2E model where no expansion embeddings are appended to the original query. This indicates the usefulness of our CWPRF model for selecting expansion embeddings to augment the query representation. We also compare the CWPRF models with the learned sparse systems, where the document tokens are enriched and reweighted, then applied with a more advanced reranker. We find that the CWPRF models significantly outperform the learned sparse models, indicating the effectiveness of learning the feedback weights and refining the query representation compared with document enrichment.

Finally, when comparing with existing dense PRF models, namely the ANCE-PRF, Vector-PRF and ColBERT-PRF models, we find that the CWPRF models exhibit significant improvements over ANCE-PRF on both query sets and significantly improves over ColBERT-PRF on the TREC 2020 query set. This indicates that our proposed CWPRF approach can select more appropriate expansion embeddings that can help to retrieve more relevant documents, and minimise topic drift.

Overall these results show that the retrieval effectiveness can be markedly improved with the CWPRF feedback weighting technique. Training CWPRF with all PRF passages as one context gives more precise retrieval at top ranks. In particular, the CWPRF approaches achieve the highest nDCG@10 and MAP performances on both query sets and exhibit upto 4.7% improvements on MAP and a 4.1% improvement on nDCG@10 for the TREC 2020 queries compared to ColBERT-PRF.

Semantic Match Proportion: To further explain the effect of implementing CWPRF for dense query expansion, following (Wang et al., 2022b), we also report the mean *semantic match proportion* (SMP) values for the models under the ColBERT dense retrieval paradigm in Table 1. In particular, SMP quantifies the extent to which a query token exhibits an exact match (matching with the same document token) and a semantic match (matching with different document tokens) in the top-ranked documents. On analysing Table 1, we find that, for both query sets, the CWPRF models show lower Mean-SMP values than ColBERT-PRF, implying a more 'focused' retrieval. This is because CWPRF's expansion embeddings correspond to the actual tokens while ColBERT-PRF's expansion embeddings can be the centroid embeddings from clustering. By using more focused embeddings, nDCG@10 is improved compared to ColBERT-PRF.

Systems	Mean Execution Time (ms)			
	Stage 1	PRF Stage	Stage 3	ALL
Vector-PRF	67	4	61	132
ANCE-PRF		111	63	241
C-PRF (KMeans) (default)	387	2997	719	4103
C-PRF (KMeans-Closest)		908	757	2052
C-PRF (KMedoids)		218	744	1349
CWPRF-AAAT		320	710	1417
CWPRF-OAAT		642	714	1743

Table 2: Mean execution time of dense pseudo-relevance feedback systems. C-PRF represents ColBERT-PRF. Effectiveness and PRF Stage efficiencies are also presented in Figure 1.

Efficiency: Following the three stages described in Figure 2, we also report the mean execution time of each stage for various dense PRF systems, including Vector-PRF, ANCE-PRF, variants of ColBERT-PRF with differing efficiency and our CWPRF methods. As Table 2 shows, our CWPRF method performs as efficiently as the most efficient ColBERT-PRF variant (KMedoids variant) and brings upto 3.06x speedup than the default ColBERT-PRF method (KMeans variant). Although CWPRF needs a longer execution time than Vector-PRF and ANCE-PRF, according to the effectiveness and efficiency tradeoff in Figure 1, CWPRF can significantly outperform them without adding much computational cost.

In summary, our CWPRF model achieves the highest nDCG@10 on the test set among all the compared baselines, while reducing the computational overhead costs compared with previous ColBERT-PRF approaches.

5.2 Ablation Study

Next, we inspect the effect of each of the training techniques, namely in-batch negative training, initialisation of the model, different learning objectives and training with PRF passages obtained from different retrieval approaches. Experiments for each training strategy are grouped in Table 3.

Effect of In-Batch Negative Sampling: In Table 3, we see that training CWPRF with further in-batch negative samples achieves higher retrieval effectiveness on both the TREC 2019 and TREC 2020 query sets, for both the AAAT and OAAT training modes. In practice, more negative training samples for the pseudo-relevance feedback tokens give more opportunity for the model to learn to properly weight unimportant terms in the feedback. For instance, the stopword "it" might occur in the feedback and positive passages, and not in the negative passage, resulting in a high weight. By applying IBNs, there is more chance for "it" to occur in any of the negative passages, reducing its learned target weight, and resulting in a more effective CWPRF model.

Effect of Model Initialisation: Here, we investigate the training from scratch and training with the parameters initialised from an existing learned sparse model, namely uniCOIL (Lin and Ma, 2021). In the second group of Table 3, we find that this initialisation for CWPRF can lead to higher performance compared with training from scratch.

Effect of Initial Retrieval: Now, we further investigate the training of CWPRF using the PRF passages obtained by sparse retrieval, using BM25, as well as by dense retrieval, using the ColBERT E2E retrieval model. From the final experiment group in Table 3, we observe that there is no obvious effectiveness difference between training CWPRF using different initial retrieval systems. Thus, considering the training efficiency, our default CWPRF is trained using the PRF passages obtained from a sparse BM25 initial retrieval.

Models	**TREC 2019 (Validation)**		**TREC 2020 (Test)**	
	MAP	nDCG@10	MAP	nDCG@10
ColBERT E2E	0.4310	0.6934	0.4648	0.6871
Effect of In-Batch Negative Sampling (IBN)				
CWPRF-AAAT	0.5168†	0.7331	0.4938	0.7079
CWPRF-AAAT-IBN	**0.5244**†	**0.7332**	0.4966†	0.7045
CWPRF-OAAT	0.5050	0.7064	0.5084†	**0.7125**
CWPRF-OAAT-IBN	0.5151†	0.7269	**0.5094**†	0.7118
Effect of Model Initialisation (Init)				
CWPRF-AAAT-Init	0.5304†	0.7301	0.5125†	0.7184†
CWPRF-AAAT-IBN-Init	**0.5319**†	**0.7444**†	**0.5136**†	**0.7246**†
CWPRF-OAAT-Init	0.5151†	0.7269	0.4948†	0.7112
CWPRF-OAAT-IBN-Init	0.5252†	0.7244	0.5049†	0.7204†
Effect of Initial Retrieval Stage				
CWPRF-AAAT (BM25)	**0.5168**†	0.7331	**0.4938**	0.7079
CWPRF-AAAT (ColBERT)	0.5109†	**0.7346**†	0.4869	0.7002
CWPRF-OAAT (BM25)	0.5050	0.7064	0.5084†	**0.7125**
CWPRF-OAAT (ColBERT)	0.5138†	0.7170	0.4983	0.6904

Table 3: Performance of CWPRF with different training strategies on the TREC 2019 & 2020 queries. '†' denotes significant improvements over the ColBERT model. The highest value for each metric within each group is boldfaced.

6 Related Work

Dense Retrieval Models: Different from the popular "cross-encoder" based BERT-rerankers (MacAvaney et al., 2019; Nogueira and Cho, 2019), dense retrieval models usually build upon a BERT-based "bi-encoder" structure. The query and document are encoded separately into dense representations. There are two families of dense retrieval models: single representation dense retrieval and multiple representation dense retrieval models (Macdonald et al., 2021). In particular, in the single representation dense retrieval paradigm, exemplified by DPR (Karpukhin et al., 2020) or ANCE (Xiong et al., 2021), each query or document is represented into a single dense representation. Thus, with the pre-computed document representations, retrieval can be conducted using the Nearest Neighbour search. In contrast, a multiple representation dense retrieval model encodes each token of the query and document into a dense representation, for instance, ColBERT model introduced by Khattab and Zaharia (2020). During retrieval, ColBERT performs an *approximate* nearest neighbour search (using FAISS (Johnson et al., 2019)) for each query embedding, followed by an exact scoring.

Pseudo-Relevance Feedback: Traditional lexical pseudo-relevance feedback (PRF) approaches, such as RM3 (Abdul-Jaleel et al., 2004) and Bo1 (Amati and Van Rijsbergen, 2002), as well as some recent proposed neural PRF models (Naseri et al., 2021; Li et al., 2018; Zheng et al., 2020) are applied upon sparse retrieval. Some initial efforts of implementing PRF mechanism for dense retrieval have been proposed recently: for instance, ColBERT-PRF (Wang et al., 2021), which is the most similar work to ours, selects cluster centroids as expansion embeddings. Different from ColBERT-PRF, where the expansion embeddings are prioritised by the closest token's IDF, our work focuses on learning the contextualised weights of the PRF tokens and identifies the prominent ones as the expansion tokens that can better differentiate between the positive and negative documents. On the other hand, ANCE-PRF (Yu et al., 2021) is a supervised PRF approach, which trains an additional query encoder. Similar to

CWPRF-AAAT, the query and passages are passed to this new encoder. However, unlike CWPRF, ANCE-PRF is trained to produce a new single embedding for the query. Due to the nature of its single embedding output, it is infeasible to analyse how the query representation has been refined in ANCE-PRF, while CWPRF provides explicit weights for each selected expansion embedding.

Contrastive Learning in IR: The contrastive learning technique has been used to optimise the query and document representations produced by the BERT-based dense retrieval models in IR. More specifically, some works focus on employing various negative sampling methods, such as the in-batch (Yih et al., 2011) and cross-batch negative sampling (Qu et al., 2021), while some works mine hard negative samples for more effective dense retrieval model (Xiong et al., 2021; Zhan et al., 2021). To the best of our knowledge, our work is the first to leverage contrastive learning for optimising the expansion weights for dense query expansion.

Feedback Weighting for PRF: Various sparse PRF models have been proposed for weighting the importance of terms occurring in the feedback documents. For instance, Clinchant and Gaussier (2011) emphasised the importance of term rarity (cf. IDF) in selecting expansion terms, a finding echoed by Roy et al. (2019) – indeed, the importance of IDF is a key insight brought into ColBERT-PRF. Going further, while there have been several approaches that have proposed supervised models for selecting high-quality expansion terms for sparse retrieval, e.g., (Cao et al., 2008; Imani et al., 2019), none of these have tackled the problem from a dense retrieval perspective, as proposed in CWPRF.

7 Conclusions

Pseudo-relevance feedback has recently been shown to be effective for dense retrieval. In this work, we propose a deep language model-based contrastive weighting approach (CWPRF) for selecting useful query expansion embeddings and calibrating their expansion weights for semantic search. In particular, CWPRF is trained with a contrastive objective to learn to assign a high weight for feedback embeddings that can distinguish relevant documents from non-relevant documents. During retrieval, the embeddings of tokens appearing in the feedback documents that CWPRF predicts to be important are appended to the query embeddings. Extensive experiments performed on two query sets demonstrate that our proposed CWPRF approach can significantly outperform the ColBERT dense retrieval model. In particular, CWPRF significantly improves over ColBERT-PRF by 4.1% in terms of nDCG@10 on the TREC 2020 query set without requiring high computational cost.

Limitations and Future Work

Our approach makes it feasible to learn the discriminative ability of an expansion embedding for dense retrieval. However, it is unclear how it may be adapted for the single-representation dense retrieval PRF model. In addition, in this work, we did not test the effect of the hard negative sampling and the number of negative samples for CWPRF. Finally, while we have focused on passage retrieval, longer document retrieval can be addressed through splitting documents into passages during indexing, retrieval and PRF, and applying a max-passage aggregation (Dai and Callan, 2019) to obtain a document ranking.

For future work, we will consider a hybrid approach to incorporate both the learned weights produced by CWPRF and the statistical information in the expansion embedding identification process. While PRF approaches typically increase query response time, they can also be used as teacher approaches to realise more effective and efficient student models (e.g., ColBERT-PRF is applied as teacher by Kim et al. (2022)). This means that improved PRF approaches, such as CWPRF, can also have downstream benefits to other retrieval approaches.

Acknowledgements

Xiao Wang acknowledges support by the China Scholarship Council (CSC) from the Ministry of Education of P.R. China.

References

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *Proceedings of TREC*.

Giambattista Amati, Claudio Carpineto, and Giovanni Romano. 2004. Query difficulty, robustness, and selective application of query expansion. In *Proceedings of ECIR*, pages 127–137.

Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)*, 20(4):357–389.

Valeriia Bolotova, Vladislav Blinov, Falk Scholer, W Bruce Croft, and Mark Sanderson. 2022. A non-factoid question-answering taxonomy. In *Proceedings of SIGIR*, pages 1196–1207.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of SIGIR*, pages 243–250.

Ben Carterette, James Allan, and Ramesh Sitaraman. 2006. Minimal test collections for retrieval evaluation. In *Proceedings of SIGIR*, pages 268–275.

Stéphane Clinchant and Eric Gaussier. 2011. Is document frequency important for PRF? In *Proceedings of ICTIR*, pages 89–100. Springer.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M Voorhees, and Ian Soboroff. 2021. TREC Deep Learning Track: reusable test collections in the large data regime. In *Proceedings of SIGIR*, pages 2369–2375.

W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of SIGIR*, page 985–988.

Zhuyun Dai and Jamie Callan. 2020. Context-aware document term weighting for ad-hoc search. In *Proceedings of WWW*, pages 1897–1907.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural IR models more effective. In *Proceedings of SIGIR*, pages 2353–2359.

Ayyoob Imani, Amir Vakili, Ali Montazer, and Azadeh Shakery. 2019. Deep neural networks for query expansion using word embeddings. In *Proceddings of ECIR*, pages 203–210.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. In *IEEE Transactions on Big Data*, pages 535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, pages 6769–6781.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of SIGIR*, pages 39–48.

Jihyuk Kim, Minsoo Kim, and Seung-won Hwang. 2022. Collective relevance labeling for passage retrieval. In *Proceedings of NAACL*.

Canjia Li, Yingfei Sun, Ben He, Le Wang, Kai Hui, Andrew Yates, Le Sun, and Jungang Xu. 2018. NPRF: A neural pseudo relevance feedback framework for ad-hoc information retrieval. In *Proceedings of EMNLP*, pages 4482–4491.

Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo relevance feedback with deep language models and dense retrievers: Successes and pitfalls. *ACM Transactions on Information Systems (TOIS)*, 41(3):1–40.

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*.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained transformers for text ranking: BERT and beyond. *arXiv preprint arXiv:2010.06467*.

Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. Streamlining evaluation with ir-measures. In *Proceedings of ECIR*, page 305–310.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of SIGIR*, pages 1101–1104.

Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified data wrangling with ir_datasets. In *Proceedings of SIGIR*, pages 2429–2436.

Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2021. On single and multiple representations in dense passage retrieval. In *IIR 2021 Workshop*.

Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of SIGIR*, pages 1723–1727.

Shahrzad Naseri, Jeffrey Dalton, Andrew Yates, and James Allan. 2021. CEQE: Contextualized embeddings for query expansion. In *Proceedings of ECIR*, pages 467–482.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, L Deng, and MS MARCO. 2016. A human generated machine reading comprehension dataset. *arXiv preprint ArXiv:1607.06275*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Proceedings of EMNLP: Findings*, pages 708–718.

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 NAACL*, pages 5835–5847.

Dwaipayan Roy, Sumit Bhatia, and Mandar Mitra. 2019. Selecting discriminative terms for relevance model. In *Proceedings of SIGIR*, pages 1253–1256.

Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. In *Neural Information Retrieval Workshop*. arXiv:1606.07608.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceddings of NeurIPS*.

Xiao Wang, Craig Macdonald, and Iadh Ounis. 2022a. Improving zero-shot retrieval using dense external expansion. *Information Processing & Management*, 59(5):103026.

Xiao Wang, Craig Macdonald, and Nicola Tonellotto. 2021. Pseudo-relevance feedback for multiple representation dense retrieval. In *Proceedings of ICTIR*, pages 297–306.

Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2022b. ColBERT-PRF: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Transactions on the Web*, 17(1):1–39.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of ICLR*.

Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of ACL*, pages 247–256.

HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. In *Proceedings of CIKM*, pages 3592–3596.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of SIGIR*, pages 1503–1512.

Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: Contextualized query expansion for document re-ranking. In *Proceedings of EMNLP: Findings*, pages 4718–4728.

Model Training Details	
Mathematical setting	cf. Section 3
Source code	`https://anonymous.4open.science/r/CWPRF-31E0/`
Computing infrastructure	NVIDIA RTX TITAN
Training time	8h
Inference time	cf. Figure 1 & Table 2
Batch size	12
Number of parameters	109M
Validation performance	cf. Table 1
Evaluation Metrics	cf. Section 4; implemented by `ir-measures` (MacAvaney et al., 2022)
Number of training runs	5
Number of evaluation runs	1
Hyper-parameter Experiments	
Bounds for hyper-parameters	$1 \leq f_p \leq 5$; $1 \leq f_e \leq 128$; $0 < \beta \leq 10$.
Hyper-parameter configurations	cf. Appendix A.1
Number of hyper-parameter search trials	3
Method of choosing hyper-parameter values	Highest retrieval effectiveness (MAP@1000) on validation set
Dataset	
Dataset Languages	English
Number of examples in datasets	Training: 39,780,811; validation: 43; test: 54.
MSMARCO obtained from	`https://microsoft.github.io/msmarco/`
Training dataset	`triples.train.small.tar.gz`
Validation & Test sets	`https://trec.nist.gov/data/deep.html`
Data pre-processing steps	Using `ir-datasets` (MacAvaney et al., 2021)

Table 4: Summary of reproducibility criteria for CWPRF.

A CWPRF Model Description

For reproducibility purposes, the source code for the training and inference of our CWPRF model is provided in our virtual appendix.[2]

A.1 Hyper-parameter Study

The hyper-parameters for CWPRF are: the number of expansion embeddings f_e and β which controls the overall contribution of the expansion embeddings. In addition, f_p defines the number of feedback documents used during training and retrieval of CWPRF.

We first vary the f_e and β hyper-parameters during retrieval. Figure 4 and Figure 5 presents the effectiveness of applying the CWPRF models while varying f_e and β, respectively. Note that $f_e = 0$ or $\beta = 0$ represents the vanilla ColBERT model without any expansion embeddings appended. From Figure 4, we find that for both CWPRF-AAAT and CWPRF-OAAT models, 10 expansion terms give the highest MAP performance. Thus, we set

[2] `github.com/Xiao0728/CWPRF_VirtualAppendix`

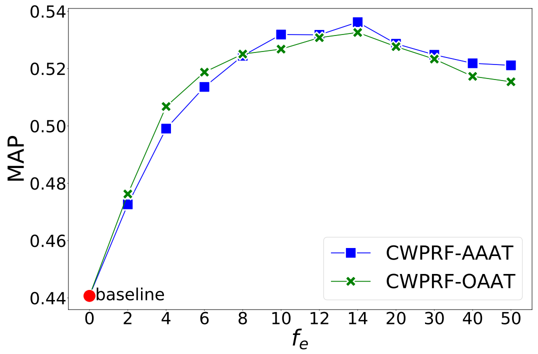


Figure 4: Impact of the number of expansion terms f_e for CWPRF on the TREC 2019 query set. ‘baseline’ represents the model without any expansion, i.e. ColBERT E2E.

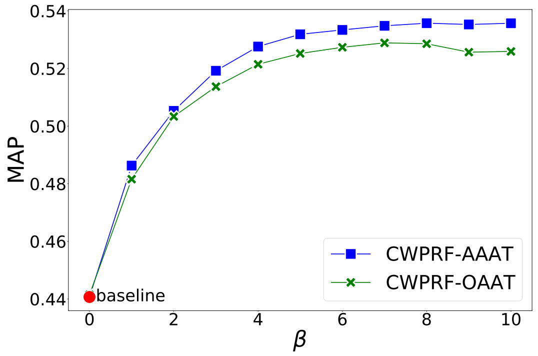


Figure 5: Impact of varying β on the TREC 2019 query set. ‘baseline’ represents the model without contribution from expansion embeddings, i.e. ColBERT E2E.

Systems	MAP	nDCG@10	Recall
ColBERT (only Q)	0.4648	0.6871	0.8245
CWPRF-AAAT (only exp)	0.4824	0.6925	0.8697†
CWPRF-AAAT (Q & exp)	**0.5136†**	**0.7246†**	**0.8783†**
CWPRF-OAAT (only exp)	0.4639	0.6750	0.8600
CWPRF-OAAT (Q & exp)	0.5049†	0.7204†	0.8783†

Table 5: Contribution of the expansion embeddings of CWPRF on the TREC 2020 test query set. † denotes significant differences over ColBERT using paired t-test with $p < 0.05$.

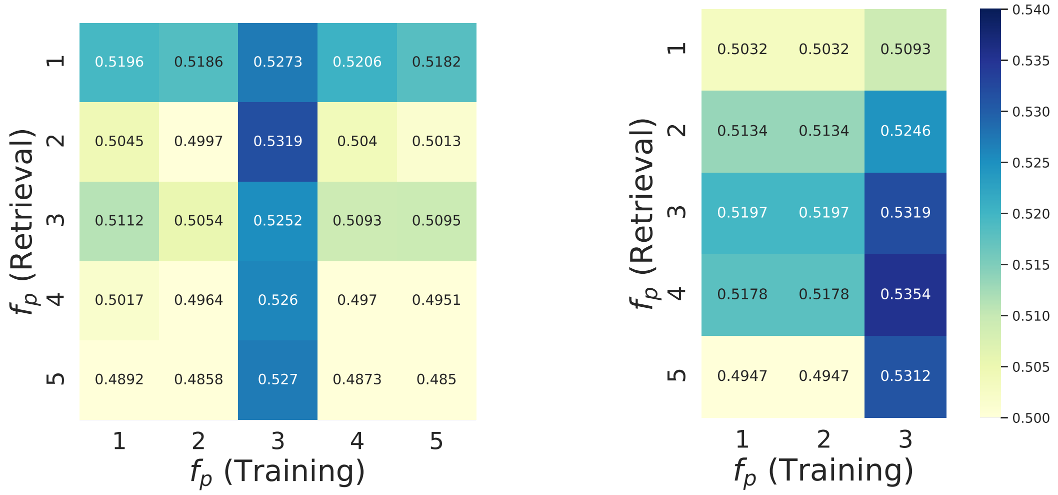
$f_e = 10$ as the default. This echoes the default expansion setting identified for ColBERT-PRF (Wang et al., 2021). For the β parameter (Figure 5), we find that for both CWPRF-AAAT and OAAT models, MAP performance shows a rising trend as higher $\beta \rightarrow 5$ and becomes stable for $\beta > 5$. Indeed for $\beta > 5$, it appears that the feedback embeddings are dominating over the original query embeddings. This indicates the high contribution of the selected expansion embeddings during retrieval. Based on this, we set $\beta = 5$ as default in this work.

Indeed, we further quantify the contribution of the expansion embeddings of CWPRF technique and the original query embeddings in respectively in Table 5. We find that for CWPRF-AAAT, using only the 10 selected expansion embeddings for reranking, markedly outperforms using the query embeddings alone, which verifies the high contribution of CWPRF selected expansion embeddings.

Furthermore, we study how many PRF passages are needed for CWPRF. We conduct experiments to train both the CWPRF-AAAT and CWPRF-OAAT models with a different number of PRF passages. We note that similar to the setting of the ANCE-PRF model, due to the input length of BERT-based encoders, for the CWPRF-AAAT training, the maximum number of PRF passages is set to 3. On the other hand, for the OAAT training mode, as each PRF document is treated independently, there is no such requirement. The nDCG@10 results are presented in Figure 6. We observe that for CW-PRF-OAAT, three feedback documents employed for training alone or evaluation alone give higher performance than other f_p values. Overall, the combination of $f_p = 3$ for both training and retrieval gives the highest performance. In addition, for CWPRF-AAAT, we find that a high MAP performance is achieved by training with only the top two PRF passages. However, this is not stable, as during retrieval, more PRF passages are needed under this setting. This indicates the model might not be trained enough. Moreover, we observe a similar trend for $f_p = 3$ used for both training and retrieval. Thus, based on this observation, we suggest to set $f_p = 3$ as the default for the training and evaluation of CWPRF.

A.2 Performance of CWPRF on BEIR

We examine the performance of the ColBERT and CWPRF (both trained on MSMARCO) in a zero-shot setting, using the BEIR datasets. We choose four datasets from BEIR that have dense judgements (Amati et al., 2004). Table 6 reports the performance of CWPRF as well as that of existing dense PRF models on four BEIR (Thakur et al., 2021) benchmarks. From Table 6, we find that CWPRF shows comparable performance with ColBERT-PRF but with much lower query latency. In addition, CWPRF outperforms ANCE-PRF by a large margin, indicating the superiority of our contrastive weighting method in such zero-shot settings.



(a) CWPRF-OAAT.

(b) CWPRF-AAAT.

Figure 6: Impact of the number of feedback passages f_p during training (x-axis) and retrieval (y-axis) for CWPRF, in terms of MAP on the TREC 2019 query set.

Models	DBPedia	NFCorpus	TREC-COVID	Touché-2020
ANCE	0.265†	0.236†	0.392†	0.291†
ANCE-PRF	0.268†	0.239†	0.430	0.292†
ColBERT	**0.392**	0.316†	0.533	0.307†
ColBERT-PRF	0.387	**0.321**	**0.548**	**0.348**
CWPRF-AAAT	0.385	**0.321**	0.524	**0.348**

Table 6: Effectiveness of CWPRF on BEIR. All scores denote nDCG@10. The best score on a given dataset is boldfaced. † denotes significant differences between CWPRF and the indicated model using paired t-test with $p < 0.05$.

A.3 Qualitative Analysis

Table 7 presents an example of the expansion tokens identified by CWPRF and the ColBERT-PRF technique as well as their retrieved top-ranked document. We observe that the two comparing methods can generate some expansion tokens in common but not necessarily received the same weights. In particular, compared to the ColBERT-PRF model, CWPRF can bring a highly relevant document (Label=2) to the top rank, by expanding with tokens: "revision" and "allows", which are helpful in retrieving the more relevant document (indicated by their darker shading). Indeed, this superior ability to retrieve highly relevant documents at high ranks is more useful in a real-life retrieval scenario. Unexpectedly, "allow" and "allows" are identified by CWPRF as important expansion tokens. This indicates that CWPRF can take the context into account – more so than IDF.

The second example in Table 7 is selected from a case when CWPRF underperforms ColBERT-PRF. Indeed, while CWPRF experiences a performance drop compared to ColBERT-PRF, it can still retrieve a document with label 3 at the top rank. This indicates the benefits of our contrastive weighting technique for bringing more relevant documents to the top positions.

Overall, we see that CWPRF can select more useful expansion embeddings to help bring more relevant documents on top, which would be more useful when implementing in a retrieval system in a real-life scenario.

A.4 Performance of CWPRF across Different Query Types

We further investigate the performance of the CW-PRF models compared to ColBERT on different query types using the query taxonomy of Bolotova et al. (2022). Specifically, we combine the TREC 2019 and TREC 2020 queries to create a single query pool, consisting of 97 queries. Then, the merged queries are classified using a trained query category classifier according to the query taxonomy introduced by Bolotova et al. (2022). Figure 7a and Figure 7b illustrate the absolute difference in performance between the CWPRF-AAAT model and the ColBERT-PRF model in terms of MAP and nDCG@10, respectively. Similarly, Figure 7c and Figure 7d provide comparisons for the CWPRF-OAAT model against ColBERT-PRF. From Figure 7, it is evident that CWPRF-AAAT demonstrates improvement across all query types in terms of MAP and nDCG@10, except

Approach	CWPRF > ColBERT-PRF	QID 156498: Query: **do google docs auto save**	
CWPRF	Expansion tokens	doc google save ##s allows revision automatically deleted allow just	
	Top returned passage after PRF	DOCNO: 104801 TEXT: **Allow Google Docs** to **automatically save** your **document**. As you add new content to your **Google Doc**, the **changes** you make to the **document** are **automatically saved** to your drive. Next to the“Help” tab at the top of your screen, you will see light gray text.	Label=2
ColBERT-PRF	Expansion tokens	##’ doc automatically google document save saves drive changes back	
	Top returned passage after PRF	DOCNO: 104803 TEXT: Allow **Google Docs** save and sync your changes **automatically**. In the offline application, **Google Drive automatically saves changes** made to a **document** every few seconds. When your computer connects to the internet, the **Google Drive** application will function like its online counterpart.	Label=1
	CWPRF < ColBERT-PRF	QID 67316: Query: **can fever cause miscarriage early pregnancy**	
CWPRF	Expansion tokens	fever cause pregnancy mis ##carriage increases baby temperature causing birth	
	Top returned passage after PRF	DOCNO: 6680964 TEXT: 1 A **temperature above 103F** (39.50C) during **early weeks of pregnancy** (usually the first trimester) may be responsible for a **miscarriage**, spinal cord or mental defects in the baby. **Fever in early pregnancy** may cause more harm than fever in late pregnancy.	Label=3
ColBERT-PRF	Expansion tokens	defects ##’ ##ping bath trim fever studies pregnancy early during	
	Top returned passage after PRF	DOCNO: 7348851 TEXT: A **temperature higher than 100.4 degrees** Fahrenheit – or the illness causing the fever – could harm both you and your developing **baby**. **A high fever** increases the risk of **birth defects or miscarriage in early pregnancy**. The higher the fever and the longer it lasts, the higher the risk. If you want to lower your fever without using medicine like acetaminophen – or just don’t have any on hand – you can try these methods: 1 Lie down and place a cool, damp washcloth on your forehead. 2 Take a lukewarm tub bath or sponge bath.	Label=3

Table 7: Example of the expansion tokens identified by the CWPRF and ColBERT-PRF approaches, as well as the top returned passage for each approach after applying PRF. Tokens with a darker red contribute more to nDCG@10.

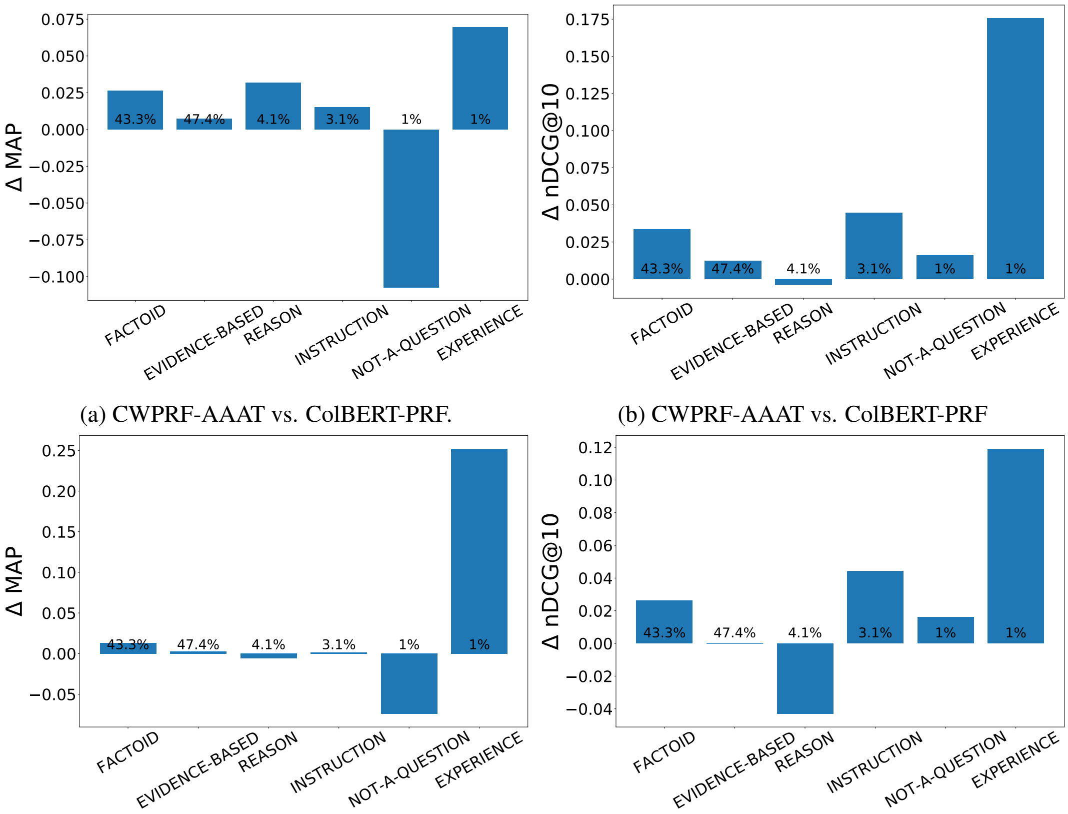
for the NOT-A-QUESTION type. However, it is worth noting that the number of queries belonging to the NOT-A-QUESTION type is quite low, comprising only approximately 1% (a single query) of the total. Similarly, we observe that CWPRF-OAAT also enhances performance across various query types, except for the single NOT-A-QUESTION type in terms of MAP, and the REASON type (with a ratio of approximately 4.1%) in terms of nDCG@10. These observations further highlight the effectiveness and robustness of our proposed CWPRF models compared to ColBERT-PRF across diverse query types.

B Semantic Match Proportion

In ColBERT and other multi-representation models using MaxSim, semantic matching of token-level embeddings occurs when the surface token form of a query embedding is matched with a document embedding that has a different token. To quantify the proportion of the query embeddings performing semantic or exact matching, following Wang et al. (2022a), we report the proportion of average semantic matching occurring for all the ColBERT related models in Table 1. More formally, given a query q and the list R_k of the top-ranked k passages, the *Semantic Match Proportion* (SMP) at rank cutoff k w.r.t. q and R_k is calculated as:

$$\mathrm{SMP}(q, R_k) = \sum_{d\in R_k} \frac{\sum_{i\in \mathsf{toks}(q)} \mathbb{1}[t_i \neq t_j]\cdot \max_{j=1,\ldots,|d|} \phi_{q_i}^T \phi_{d_j}}{\sum_{i\in \mathsf{toks}(q)} \max_{j=1,\ldots,|d|} \phi_{q_i}^T \phi_{d_j}}, \tag{8}$$

where t_i and t_j denote the token ids of the i-th query embedding and j-th passage embedding, respectively. In this work, we report the Mean-SMP values calculated at rank cutoff $k = 10$ in Table 1.



(a) CWPRF-AAAT vs. ColBERT-PRF.

(b) CWPRF-AAAT vs. ColBERT-PRF

(c) CWPRF-OAAT vs. ColBERT-PRF.

(d) CWPRF-OAAT vs. ColBERT-PRF.

Figure 7: Performance of CWPRF compared to ColBERT-PRF across different types of queries according to the query type taxonomy proposed by Bolotova et al. (2022). The percentage of queries within each query type, relative to the total number of queries in the query pool, is indicated within each bar.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations and Future Work

☑ A2. Did you discuss any potential risks of your work?
Limitations and Future Work

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

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

B ☑ Did you use or create scientific artifacts?

Section 3 Contrastive Weighting for Dense PRF

□ B1. Did you cite the creators of artifacts you used?
No response.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Sections 4, 5, 6

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

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

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*We conduct our experiments using the MS MARCO (**?**) passage ranking dataset. The corpus consists of 8.8M passages from web pages, along which are provided 0.5M training queries with sparse document relevance judgements. We employ the TREC-DL 2019 (43 queries with an average of 215 relevant documents per query) query set as our validation set and use TREC 2020 (54 queries with 211 relevance assessments per query) query set as our test set due to their dense judgements, which can provide more reliable evaluations (**??**).*

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

C ☑ Did you run computational experiments?

Sections 4, 5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4 and Appendix

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4 and Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sections 5, 6, 7 and Appendix

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4 and Appendix

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

Left blank.

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

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

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

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

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