Pseudo-Relevance Feedback for Multiple Representation Dense Retrieval

Xiao Wang[1], Craig Macdonald[1], Nicola Tonellotto[2], Iadh Ounis[1]
[1]University of Glasgow, [2]University of Pisa
UK, Italy

ABSTRACT

Pseudo-relevance feedback mechanisms, from Rocchio to the relevance models, have shown the usefulness of expanding and reweighting the users' initial queries using information occurring in an initial set of retrieved documents, known as the pseudo-relevant set. Recently, dense retrieval – through the use of neural contextual language models such as BERT for analysing the documents' and queries' contents and computing their relevance scores – has shown a promising performance on several information retrieval tasks still relying on the traditional inverted index for identifying documents relevant to a query. Two different dense retrieval families have emerged: the use of single embedded representations for each passage and query (e.g. using BERT's [CLS] token), or via multiple representations (e.g. using an embedding for each token of the query and document). In this work, we conduct the first study into the potential for multiple representation dense retrieval to be enhanced using pseudo-relevance feedback. In particular, based on the pseudo-relevant set of documents identified using a first-pass dense retrieval, we extract representative feedback embeddings (using KMeans clustering) – while ensuring that these embeddings discriminate among passages (based on IDF) – which are then added to the query representation. These additional feedback embeddings are shown to both enhance the effectiveness of a reranking as well as an additional dense retrieval operation. Indeed, experiments on the MSMARCO passage ranking dataset show that MAP can be improved by upto 26% on the TREC 2019 query set and 10% on the TREC 2020 query set by the application of our proposed ColBERT-PRF method on a ColBERT dense retrieval approach.

ACM Reference Format:
Xiao Wang, Craig Macdonald, Nicola Tonellotto, Iadh Ounis. 2021. Pseudo-Relevance Feedback for Multiple Representation Dense Retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '21), July 11, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3471158.3472250

1 INTRODUCTION

Query expansion approaches, which rewrite the user's query, have been shown to be an effective approach to alleviate the vocabulary discrepancies between the user query and the relevant documents, by modifying the user's original query to improve the retrieval effectiveness. Many approaches follow the pseudo-relevance feedback (PRF) paradigm – such as Rocchio's algorithm [28], the RM3 relevance language model [1], or the DFR query expansion models [4] – where terms appearing in the top-ranked documents for the initial query are used to expand it. Query expansion (QE) approaches have also found a useful role when integrated with effective BERT-based neural reranking models, by providing a high quality set of candidate documents obtained using the expanded query, which can then be reranked [27, 32, 35].

On the other hand, many studies have focused on the use of *static* word embeddings, such as Word2Vec, within query expansion methods [11, 17, 30, 31]. Indeed, most of the existing embedding-based QE methods [11, 17, 30, 31, 36] are based on static embeddings, where a word embedding is always the same within different sentences, and hence they do not address contextualised language models such as BERT. Recently, CEQE [23] was proposed, which makes use of contextualised BERT embeddings for query expansion. The resulting refined query representation is then used for a further round of retrieval using a conventional inverted index. In contrast, in this paper, we focus on implementing contextualised embedding-based query expansion for dense retrieval.

Indeed, the BERT models have demonstrated further promise in being a suitable basis for *dense retrieval*. In particular, instead of using a classical inverted index, in dense retrieval, the documents and queries are represented using embeddings. Then, the documents can be retrieved using an approximate nearest neighbour algorithm – as exemplified by the FAISS toolkit [14]. Two distinct families of approaches have emerged: single representation dense retrieval and multiple representation dense retrieval. In single-representation dense retrieval, as used by DPR [15] and ANCE [34], each query or document is represented entirely by the single embedding of the [CLS] (classification) token computed by BERT. Query-document relevance is estimated in terms of the similarity of the corresponding [CLS] embeddings. In contrast, in multiple representation dense retrieval – as proposed by ColBERT [16] – each term of the queries and documents is represented by a single embedding. For each query embedding, one per query term, the nearest document token embeddings are identified using an approximate nearest neighbour search, before a final re-scoring to obtain exact relevance estimations.

In this work, we are concerned with applying pseudo-relevance feedback in a multiple representation dense retrieval setting. Indeed, as retrieval uses multiple representations, this allows additional useful embeddings to be appended to the query representation. Furthermore, the exact scoring stage provides the document embeddings in response to the original query, which can be used as pseudo-relevance information. Thus, in this work, we propose a pseudo-relevance feedback mechanism called ColBERT-PRF for dense retrieval. In particular, as embeddings cannot be counted, ColBERT-PRF applies clustering to the embeddings occurring in

ICTIR '21, July 11, 2021, Virtual Event, Canada

the pseudo-relevant set, and then identifies the most discriminative embeddings among the cluster centroids. These centroids are then appended to the embeddings of the original query. ColBERT-PRF is focussed on multiple representation dense retrieval settings; However, compared to existing work, our approach is the first work to apply pseudo-relevance feedback to any form of dense retrieval setting; moreover, among the existing approaches applying deep learning for pseudo-relevance feedback, our work in this paper is the first that can improve the recall of the candidate set by re-executing the expanded query representation upon the dense retrieval index, and thereby identify more relevant documents that can be highly ranked for the user. In summary, our work makes the following contributions: (1) we propose a novel contextualised pseudo-relevance feedback mechanism for multiple representation dense retrieval; (2) we cluster and rank the feedback document embeddings for selecting candidate expansion embeddings; (3) we evaluate our proposed contextualised PRF model in both ranking and reranking settings and discuss its efficiency.

The remainder of this paper is as follows: Section 2 positions this work among existing approaches to pseudo-relevance feedback; Section 3 describes a multi-representation dense retrieval, while Section 4 presents our proposed dense PRF method. Next, we discuss our experimental setup and results in Section 5 & 6, respectively; We provide concluding remarks in Section 7.

2 RELATED WORK

Pseudo-relevance feedback approaches have a long history in Information Retrieval (IR) going back to Rocchio [28] who generated refined query reformulations through linear combinations of the sparse vectors (e.g. containing term frequency information) representing the query and the top-ranked feedback documents. Refined classical PRF models, such as Divergence from Randomness's Bo1 [4], KL [2], and RM3 relevance models [1] have demonstrated their effectiveness on many test collections. Typically, these models identify and weight feedback terms that are frequent in the feedback documents and infrequent in the corpus, by exploiting statistical information about the occurrence of terms in the documents and in the whole collection.

Recently, deep learning solutions based on transformer networks have been used to enrich the statistical information about terms by rewriting or expanding the collection of documents. For instance, DeepCT [9] reweights terms occurring in the documents according to a fine-tuned BERT model to highlight important terms. This results in *augmented* document representations, which can be indexed using a traditional inverted indexer. Similarly, doc2query [25] and its more modern variant docT5query [24] apply text-to-text translation models to each document in the collection to suggest queries that may be relevant to the document. When the suggested queries are indexed along with the original document, the retrieval effectiveness is enhanced.

More recently, instead of leveraging (augmented) statistical information such as the in-document and collection frequency of terms to model a query or a document, dense representations, also known as embeddings, are becoming commonplace. Embeddings encode terms in queries and documents by learning a vector representation for each term, which takes into account the word semantic and context. Instead of identifying the related terms in the pseudo-relevance feedback documents using statistical methods, embedding-based query expansion methods [11, 17, 30, 31, 36] expand a query with terms that are closest to the query terms in the word embedding space. However, the expansion terms may not be sufficiently informative to distinguish relevant documents from non-relevant documents – for instance, the embedding of "grows" may be closest to "grow" in the embedding space, but adding "grows" to the query may not help to identify more relevant documents. Moreover, all these embedding-based method are based on non-contextualised embeddings, where a word embedding is always the same within different sentences, and hence they do not address contextualised language models. Pre-trained contextualised language models such as BERT [10] have brought large effectiveness improvements over prior art in information retrieval tasks. In particular, deep learning is able to successfully exploit general language features in order to capture the contextual semantic signals allowing to better estimate the relevance of documents w.r.t. a given query.

Query expansion approaches have been used for generating a high quality pool of candidate documents to be reranked by effective BERT-based neural reranking models [27, 32, 35]. However the use of BERT models directly within the pseudo-relevance feedback mechanism has seen comparatively little use in the literature. The current approaches leveraging the BERT contextualised embeddings for PRF are Neural PRF [18], BERT-QE [37] and CEQE [23].

In particular, Neural PRF uses neural ranking models, such as DRMM [13] and KNRM [33], to score the similarity of a document to a top-ranked feedback document. BERT-QE is conceptually similar to Neural PRF, but it measures the similarity of each document w.r.t. feedback chunks that are extracted from the top-ranked feedback documents. This results in an expensive application of many BERT computations – approximately 11× as many GPU operations than a simple BERT reranker [37]. Both Neural PRF and BERT-QE approaches leverage contextualised language models to rerank an initial ranking of documents retrieved by a preliminary sparse retrieval system. However, they cannot identify any new relevant documents from the collection that were not retrieved in the initial ranking.

CEQE exploits BERT to compute contextualised representations for the query as well as for the terms in the top-ranked feedback documents, and then selects as expansion terms those which are the closest to the query embeddings according to some similarity measure. In contrast to Neural PRF and BERT-QE, CEQE is used to generate a new query of terms for execution upon a conventional inverted index. This means that the contextual meaning of an expansion term is lost - for instance, a polysemous word added to the query can result in a topic drift.

In contrast to the aforementioned approaches, our proposed ColBERT-PRF approach can be exploited in a dense retrieval system, both in end-to-end ranking and reranking scenarios. Dense retrieval approaches, exemplified by ANCE [34] and ColBERT [16], are of increasing interest, due to their use of the BERT embedding(s) for representing queries and documents. By using directly the BERT embeddings for retrieval, topic drifts for polysemous words can be avoided. To the best of our knowledge, our paper is the first work investigating PRF in a dense retrieval setting.

3 MULTI REPRESENTATION DENSE RETRIEVAL

The queries and documents are represented by tokens from a vocabulary V. Each token occurrence has a contextualised real-valued vector with dimension d, called an embedding. More formally, let $f : V^n \rightarrow \mathbb{R}^{n \times d}$ be a function mapping a sequence of terms $\{t_1, \ldots, t_n\}$, representing a query q, composed by $|q|$ tokens into a set of embeddings $\{\phi_{q_1}, \ldots, \phi_{q_{|q|}}\}$ and a document composed by $|d|$ tokens into a set of embeddings $\{\phi_{d_1}, \ldots, \phi_{d_{|d|}}\}$. Khattab & Zaharia [16] recommended that the number of query embeddings be 32, with extra [MASK] tokens being used as query augmentation. Indeed, these mask tokens are a differentiable mechanism that allows documents to gain score contributions from embeddings that do not actually occur in the query, but which the model assumes could be present in the query.

The similarity of two embeddings is computed by the dot product. Hence, for a query q and a document d, their similarity score $s(q, d)$ is obtained by summing the maximum similarity between the query token embeddings and the document token embeddings [16]:

$$s(q, d) = \sum_{i=1}^{|q|} \max_{j=1,\ldots,|d|} \phi_{q_i}^T \phi_{d_j} \tag{1}$$

Indeed, Formal et al. [12] showed that the dot product $\phi_{q_i}^T \phi_{d_j}$ used by ColBERT implicitly encapsulates token importance, by giving higher scores to tokens that have higher IDF values.

To obtain a first set of candidate documents, Khattab & Zaharia [16] make use of FAISS, an approximate nearest neighbour search library, on the pre-computed document embeddings. Conceptually, FAISS allows to retrieve the k' documents containing the nearest neighbour document embeddings to a query embedding ϕ_{q_i}, i.e., it provides a function $\mathcal{F}_d(\phi_{q_i}, k') \rightarrow (d, \ldots)$ that returns a list of k' documents, sorted in decreasing approximate scores.

However, these approximate scores are insufficient for accurately depicting the similarity scores of the documents, hence the accurate final document scores are computed using Equation (1) in a second pass. Typically, for each query embedding, the nearest $k' = 1,000$ documents are identified. The union of these documents are reranked using Equation (1). A separate index data structure (typically in memory) is used to store the uncompressed embeddings for each document. To the best of our knowledge, ColBERT [16] exemplifies the implementation of an end-to-end IR system that uses multiple representation. Algorithm 1 summarises the ColBERT retrieval algorithm for the end-to-end dense retrieval approach proposed by Khattab & Zaharia, while the top part of Table 1 summarises the notation for the main components of the algorithm.

The easy access to the document embeddings used by ColBERT provides an excellent basis for our dense retrieval pseudo-relevance feedback approach. Indeed, while the use of embeddings (including [MASK] embeddings) in ColBERT addresses the vocabulary mismatch problem, we argue that identifying more related embeddings from the top-ranked documents may help to further refine the document ranking. In particular, as we will show, this permits representative embeddings from a set of pseudo-relevant documents to be used to refine the query representation ϕ.

Table 1: Summary of notation – top group for ColBERT dense retrieval; bottom group for ColBERT-PRF.

Symbol	Meaning
ϕ_{q_i}, ϕ_{d_j}	An embedding for a query token q_i or a document token d_j
$\mathcal{F}_d(\phi_{q_i}, k')$	Function returning a list of the k' documents closest to embedding ϕ_{q_i}
Φ	Set of feedback embeddings from f_b top-ranked feedback documents
v_i	A representative (centroid) embedding selected by applying KMeans among Φ
K	Number of representative embeddings to select, i.e., number of clusters for KMeans
$\mathcal{F}_t(v_i, r)$	Function returning the r token ids corresponding to the r closest document embeddings to embedding v_i
σ_i	Importance score of v_i, calculated as the IDF score of its most likely token
F_e	Set of expansion embeddings
f_e	Number of expansion embeddings selected from K representative embeddings
f_b	Number of feedback documents
β	Parameter weighting the contribution of the expansion embeddings

Algorithm 1: The ColBERT E2E algorithm

Input : A query Q
Output: A set A of (docid, score) pairs
ColBERT E2E(Q):

$\phi_{q_1}, \ldots, \phi_{q_n} \leftarrow \text{Encode}(Q)$
$D \leftarrow \emptyset$
for ϕ_{q_i} **in** $\phi_{q_1}, \ldots, \phi_{q_n}$ **do**
  $D \leftarrow D \cup \mathcal{F}_d(\phi_{q_i}, k')$
$A \leftarrow \emptyset$
for d **in** D **do**
  $s \leftarrow \sum_{i=1}^{|q|} \max_{j=1,\ldots,|d|} \phi_{q_i}^T \phi_{d_j}$
  $A \leftarrow A \cup \{(d, s)\}$
return A

4 DENSE PSEUDO-RELEVANCE FEEDBACK

The aim of a pseudo-relevance feedback approach is typically to generate a refined query representation by analysing the text of the feedback documents. In our proposed ColBERT-PRF approach, we are inspired by conventional PRF approaches such as Bo1 [4] and RM3 [1], which assume that good expansion terms will occur frequently in the feedback set (and hence are somehow *representative* of the information need underlying the query), but infrequent in the collection as a whole (therefore are sufficiently *discriminative*). Therefore, we aim to encapsulate these intuitions while operating in the embedding space $\mathbb{R}^d$, where the exact counting of frequencies is not actually possible. In particular, the bottom part of Table 1 summarises the main notations that we use in describing ColBERT-PRF.

In this section, we detail how we identify representative (centroid) embeddings from the feedback documents (Section 4.1), how we ensure that those centroid embeddings are sufficiently discriminative (Section 4.2), and how we apply these discriminative representative centroid embeddings for (re)ranking (Section 4.3). We conclude with an illustrative example (Section 4.4) and a discussion of the novelty of ColBERT-PRF (Section 4.5).

4.1 Representative Embeddings in Feedback Documents

First, we need to identify representative embeddings $\{v_1, \ldots, v_K\}$ among all embeddings in the feedback documents set. A typical "sparse" PRF approach – such as RM3 – would count the frequency of terms occurring in the feedback set to identify representative ones. However, in a dense embedded setting, the document embeddings are not countable. Instead, we resort to clustering to identify patterns in the embedding space that are representative of embeddings.

Specifically, let $\Phi(q, f_b)$ be the set of all document embeddings from the f_b top-ranked feedback documents. Then, we apply the KMeans clustering algorithm to $\Phi(q, f_b)$:

$$\{v_1, .., v_K\} = \mathsf{KMeans}\big(K, \Phi(q, f_b)\big). \tag{2}$$

By applying clustering, we obtain K representative centroid embeddings of the feedback documents. Next, we determine how well these centroids discriminate among the documents in the corpus.

4.2 Identifying Discriminative Embeddings among Representative Embeddings

Many of the K representative embeddings may represent stopwords and therefore are not sufficiently informative when retrieving documents. Typically, identifying informative and discriminative expansion terms from feedback documents would involve examining the collection frequency or the document frequency of the constituent terms [6, 29]. However, there may not be a one-to-one relationship between query/centroid embeddings and actual tokens, hence we seek to map each centroid v_i to a possible token t.

We resort to FAISS to achieve this, through the function $\mathcal{F}_t(v_i, r) \rightarrow (t, \ldots)$ that, given the centroid embedding v_i and r, returns the list of the r *token ids* corresponding to the r closest document embeddings to the centroid.[1] From a probabilistic viewpoint, the likelihood $P(t|v_i)$ of a token t given an embedding v_i can be obtained as:

$$P(t|v_i) = \frac{1}{r} \sum_{\tau \in \mathcal{F}_t(v_i, r)} \mathbb{1}[\tau = t], \tag{3}$$

where $\mathbb{1}[]$ is the indicator function. In this work, for simplicity, we choose the most likely token id, i.e., $t_i = \arg\max_t P(t|v_i)$. Mapping back to a token id allows us to make use of Inverse Document Frequency (IDF), which can be pre-recorded for each token id. The importance σ_i of a centroid embedding v_i is obtained using a traditional IDF formula[2]: $\sigma_i = \log\left(\frac{N+1}{N_i+1}\right)$, where N_i is the number of passages containing the token t_i and N is the total number of passages in the collection. While this approximation of embedding informativeness is obtained by mapping back to tokens, as we shall show, it is very effective. We leave to future work the derivation of a tailored informativeness measure based upon embeddings alone, for instance using kernel density estimation upon the embedding space. Finally, we select the f_e most informative centroids as expansion embeddings based on the σ_i importance scores as follows:

$$F_e = \mathsf{TopScoring}\Big(\big\{(v_1, \sigma_1), \ldots, (v_K, \sigma_K)\big\}, f_e\Big) \tag{4}$$

where $\mathsf{TopScoring}(A, c)$ returns the c elements of A with the highest importance score.

[1] This additional mapping can be recorded at indexing time, using the same FAISS index as for dense retrieval, increasing the index size by 3%.

[2] We have observed no marked empirical benefits in using other IDF formulations.

4.3 Ranking and Reranking with ColBERT-PRF

Given the original $|q|$ query embeddings and the f_e expansion embeddings, we incorporate the score contributions of the expansion embeddings in Eq. (1) as follows:

$$s(q, d) = \sum_{i=1}^{|q|} \max_{j=1,\ldots,|d|} \phi_{q_i}^T \phi_{d_j} + \beta \sum_{(v_i, \sigma_i) \in F_e} \max_{j=1,\ldots,|d|} \sigma_i v_i^T \phi_{d_j}, \tag{5}$$

where $\beta > 0$ is a parameter weighting the contribution of the expansion embeddings, and the score produced by each expansion embedding is further weighted by the IDF weight of its most likely token, σ_i. Note that Equation (5) can be applied to rerank the documents obtained from the initial query, or as part of a full re-execution of the full dense retrieval operation including the additional f_e expansion embeddings. In both ranking and reranking, ColBERT-PRF has 4 parameters: f_b, the number of feedback documents; K, the number of clusters; $f_e \leq K$, the number of expansion embeddings; and β, the importance of the expansion embeddings during scoring. Furthermore, we provide the pseudo-code of our proposed ColBERT PRF ReRanker in Algorithm 2. The ColBERT-PRF Ranker can be easily obtained by inserting lines 3-4 of Algorithm 1 at line 10 of Algorithm 2, and adapting the max-sim scoring in Eq. (1) to encapsulate the original query embeddings as well as the expansion embeddings.

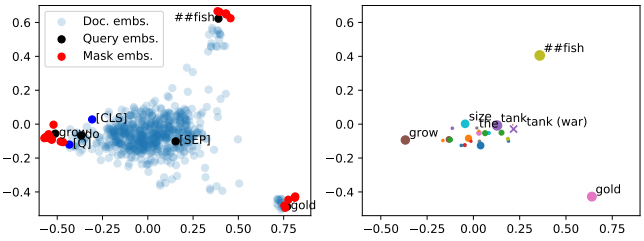
4.4 Illustrative Example

We now illustrate the effect of ColBERT-PRF upon one query from the TREC 2019 Deep Learning track, 'do goldfish grow'. We use PCA to quantize the 128-dimension embeddings into 2 dimensions purely to allow visualisation. Firstly, Figure 1(a) shows the embeddings of the original query (black ellipses); the red [MASK] tokens are also visible, clustered around the original query terms (##fish, gold, grow). Meanwhile, document embeddings extracted from 10 feedback documents are shown as light blue ellipses in Figure 1(a). There appear to be visible clusters of document embeddings near the query embeddings, but also other document embeddings exhibit some clustering. The mass of embeddings near the origin is not distinguishable in PCA. Figure 1(b) demonstrates the application of KMeans clustering upon the document embeddings; we map back to the original tokens by virtue of Equation (3). In Figure 1(b), the point size is indicative of the IDF of the corresponding token. We can see that the cluster centroids with high IDF correspond to the original query tokens ('gold', '##fish', 'grow'), as well as the related terms ('tank', 'size'). In contrast, a centroid with low IDF is 'the'. This illustrates the utility of our proposed ColBERT-PRF approach in using KMeans to identify representative clusters of embeddings, as well as using IDF to differentiate useful clusters.

Algorithm 2: The ColBERT PRF (reranking) algorithm

Input : A query Q,
number of feedback documents f_b,
number of representative embeddings K,
number of expansion embeddings f_e
Output: A set B of (docid, score) pairs
ColBERT PRF(Q):

$A \leftarrow$ ColBERT E2E(Q)
$\Phi(Q, f_b) \leftarrow$ set of all document embeddings from the f_b top-scored documents in A
$V \leftarrow \emptyset$
$v_1, \ldots, v_K = \mathsf{KMeans}\big(K, \Phi(Q, f_b)\big)$
for v_i **in** $v_1, \ldots, v_K$ **do**
  $t_i \leftarrow \arg\max_t \frac{1}{r} \sum_{\tau \in \mathcal{F}_t(v_i, r)} \mathbb{1}[\tau = t]$
  $\sigma_i \leftarrow \log\left(\frac{N+1}{N_i+1}\right)$
  $V \leftarrow V \cup \{(v_i, \sigma_i)\}$
$F_e \leftarrow$TopScoring(V, f_e)
$B \leftarrow \emptyset$
for (d, s) **in** A **do**
  $s \leftarrow s + \beta \sum_{(v_i, \sigma_i) \in F_e} \max_{j=1,\ldots,|d|} \sigma_i v_i^T \phi_{d_j}$
  $B \leftarrow B \cup \{(d, s)\}$
return B



(a) Query & doc. embeddings. (b) Cluster centroids, $K = 24$.

Figure 1: Example showing how ColBERT-PRF operates for the query 'do goldfish grow' in a 2D PCA space. In Figure 1(b), the point size is representative of IDF; five high IDF and one low IDF centroids are shown. For contrast, × 'tank (war)' denotes the embedding of 'tank' occurring in a non-fish context.

Furthermore, Figure 1(b) also includes, marked by an × and denoted 'tank (war)', the embedding for the word 'tank' when placed in the passage *"While the soldiers advanced, the tank bombarded the troops with artillery"*. It can be seen that, even in the highly compressed PCA space, the 'tank' centroid embedding is distinct from the embedding of 'tank (war)'. This shows the utility of ColBERT-PRF when operating in the embedding space, as the PRF process for the query 'do goldfish grow' will not retrieve documents containing 'tank (war)', but will focus on a fish-related context, thereby dealing with the polysemous nature of a word such as 'tank'. To the best of our knowledge, this is a unique feature of ColBERT-PRF among PRF approaches.

4.5 Discussion

To the best of our knowledge ColBERT-PRF is the first investigation of pseudo-relevance feedback for dense retrieval. Existing works on neural pseudo-relevance feedback, such as Neural PRF [18] and BERT-QE [37] only function as rerankers. Other approaches such as DeepCT [9] and doc2query [24, 25] use neural models to augment documents before indexing using a traditional inverted index. CEQE [23] generates words to expand the initial query, which is then executed on the inverted index. However, returning the BERT embeddings back to textual word forms can result in polysemous words negatively affecting retrieval. In contrast, ColBERT-PRF operates entirely on an existing dense index representation (without augmenting documents), and can function for both ranking as well as reranking. By retrieving using feedback embeddings directly, ColBERT-PRF addresses polysemous words (such as 'tank', illustrated above). Furthermore, it also requires no additional neural network training beyond that of ColBERT.

5 EXPERIMENTAL SETUP

Our experiments address the three following research questions:

- RQ1: Can a multiple representation dense retrieval approach be enhanced by pseudo-relevance feedback, i.e. can ColBERT-PRF outperform ColBERT dense retrieval?
- RQ2: How does ColBERT-PRF compare to other existing baseline and state-of-the-art approaches, namely:
(a) lexical (sparse) baselines, including using PRF,
(b) neural augmentation approaches, namely DeepCT and docT5query,
(c) BERT-QE Reranking models?
- RQ3: What is the impact of the parameters of ColBERT-PRF, namely the number of clusters and expansion embeddings, the number of feedback documents and the β parameter controlling the influence of the expansion embeddings?

5.1 Dataset & Measures

Experiments are conducted on the MSMARCO passage corpus, using the TREC 2019 Deep Learning track topics (43 topics with an average of 95.4 relevance judgements per query), as well as the TREC 2020 Deep Learning track topics (54 topics with an average of 66.8 relevance judgements per query). We omit topics from the MSMARCO Dev set, which have only sparse judgements, ~1.1 per query. Indeed, pseudo-relevance feedback approaches are known to be not effective on test collections with few judged documents [3].

We report the commonly used metrics for the TREC 2019 and TREC 2020 query sets following the corresponding track overview papers [7, 8]: we report mean reciprocal rank (MRR) and normalised discounted cumulative gain (NDCG) calculated at rank 10, as well as Recall and Mean Average Precision (MAP) at rank 1000 [8]. For the MRR, MAP and Recall metrics, we treat passages with label grade 1 as non-relevant, following [7, 8]. In addition, we also report the Mean Response Time (MRT) for each retrieval system. For significance testing, we use the paired t-test ($p < 0.05$) and apply the Holm-Bonferroni multiple testing correction.

5.2 Implementation and Settings

We conduct experiments using PyTerrier [22] and, in particular using our PyTerrier_ColBERT plugin[3], which includes ColBERT-PRF as well as our adaptations of the ColBERT source code. In terms of the ColBERT configuration, we train ColBERT upon the MSMARCO passage ranking triples file for 44,000 batches, applying the parameters specified by Khattab & Zaharia in [16]: Maximum document length is set to 180 tokens and queries are encoded into 32 query embeddings (including [MASK] tokens); We encode all passages to a FAISS index that has been trained using 5% of all embeddings; At retrieval time, FAISS retrieves $k' = 1000$ document embeddings for every query embedding.

ColBERT-PRF is implemented using the KMeans implementation [5] of sci-kit learn (sklearn). For query expansion settings, we follow the default settings of Terrier [26], which is 10 expansion terms obtained from 3 feedback documents; we follow the same default setting for ColBERT-PRF, additionally using representative values, namely $K = 24$ clusters[4], and $\beta = \{0.5, 1\}$ for the weight of the expansion embeddings. We later show the impact of these parameters when we address RQ3.

5.3 Baselines

To test the effectiveness of our proposed dense PRF approach, we compare with four families of baseline models, for which we vary the use of a BERT-based reranker (namely BERT or ColBERT). For the BERT reranker, we use OpenNIR [21] and `capreolus/bert-base-msmarco` fine-tuned model from [19]. For the ColBERT reranker, unless otherwise noted, we use the existing pre-indexed ColBERT representation of documents for efficient reranking. The four families are:

Lexical Retrieval Approaches: These are traditional retrieval models using a sparse inverted index, with and without BERT and ColBERT rerankers, namely: (i) BM25 (ii) BM25+BERT (iii) BM25+ColBERT, (iv) BM25+RM3, (v) BM25+RM3+BERT and (vi) BM25+RM3+ColBERT.

Neural Augmentation Approaches: These use neural components to augment the (sparse) inverted index: (i) BM25+DeepCT and (ii) BM25+docT5query, both without and with BERT and ColBERT rerankers. For BM25+docT5query+ColBERT, the ColBERT reranker is applied on expanded document texts encoded at querying time, rather than the indexed ColBERT representation. The response time for BM25+docT5query+ColBERT reflects this difference.

Dense Retrieval Models: This family consists of the dense retrieval approaches: (i) ANCE: The ANCE [34] model is a single representation dense retrieval model. We use the trained models provided by the authors trained on MSMARCO training data. (ii) ColBERT E2E: ColBERT end-to-end (E2E) [16] is the dense retrieval version of ColBERT, as defined in Section 3.

BERT-QE Models: We apply BERT-QE [37] on top of a strong sparse baseline and our dense retrieval baseline, ColBERT E2E, i.e. (i) BM25+RM3+ColBERT+BERT-QE and (ii) ColBERT E2E+BERT-QE; Where possible, we use the ColBERT index for scoring passages; for identifying the top scoring chunks within passages, we use ColBERT in a slower "text" mode, i.e. without using the index. For the BERT-QE parameters, we follow the settings in [37], in particular using the recommended settings of $\alpha = 0.4$ and $\beta = 0.9$, which are also the most effective on MSMARCO. Indeed, to the best our knowledge, this is the first application of BERT-QE upon dense retrieval, the first application of BERT-QE on MSMARCO and the first application using ColBERT. We did attempt to apply BERT-QE using the BERT re-ranker, but we found it to be ineffective on MSMARCO, and exhibiting a response time exceeding 30 seconds per query, hence we omit it from our experiments.

Note that, at the time of writing, there is no publicly available implementation of the very recent CEQE [23] approach, and hence we omit it from our experiments. Code to reproduce ColBERT-PRF is available in the PyTerrier_ColBERT repository.[3]

[3] https://github.com/terrierteam/pyterrier_colbert [4] Indeed, $K = 24$ gave reasonable looking clusters in our initial investigations, and, as we shall see in Section 6.3, is an effective setting for the TREC 2019 query set.

6 RESULTS

In the following, we analyse the performance of ColBERT-PRF wrt. RQs 1-3. For RQs 1 & 2 (Sections 6.1 & 6.2), we analyse Table 2, which reports the performances of all the baselines as well as the ColBERT-PRF models on the 43 TREC 2019 & 54 TREC 2020 queries. Baselines are grouped by the retrieval families discussed in Section 5.3.

6.1 Results for RQ1

In this section, we examine the effectiveness of a pseudo-relevance feedback technique for the ColBERT dense retrieval model. On analysing Table 2, we first note that the ColBERT dense retrieval approach outperforms the single representation dense retrieval model, i.e. ANCE for all metrics on both test query sets, probably because the single-representation used in ANCE provides limited information for matching queries and documents [20]. Based on this, we then compare the performances of our proposed ColBERT-PRF models, instantiated as ColBERT-PRF Ranker & ColBERT-PRF ReRanker, with the more effective ColBERT E2E model. We find that both the Ranker and ReRanker models outperform ColBERT E2E on all the metrics for both used query sets. Typically, on the TREC 2019 test queries, both the Ranker and ReRanker models exhibit significant improvements in terms of MAP over the ColBERT E2E model. In particular, we observe a 26% increase in MAP on TREC 2019[5] and 10% for TREC 2020 over ColBERT E2E for the ColBERT-PRF Ranker. In addition, both ColBERT-PRF Ranker and ReRanker exhibit significant improvements over ColBERT E2E in terms of NDCG@10 on TREC 2019 queries.

The high effectiveness of ColBERT-PRF Ranker (which is indeed higher than ColBERT-PRF ReRanker) can be explained in that the expanded query obtained using the PRF process introduces more relevant documents, thus it increases recall after re-executing the query on the dense index. As can be seen from Table 2, ColBERT-PRF Ranker exhibits significant improvements over both ANCE and ColBERT E2E models on Recall. On the other hand, the effectiveness of ColBERT-PRF ReRanker also suggests that the expanded query provides a better query representation, which can better rank documents in the existing candidate set. We further investigate the actual expansion tokens using the ColBERT-PRF model. Table 3 lists three example queries from both the TREC 2019 and 2020 query sets and their tokenised forms as well as the expansion tokens generated by the ColBERT-PRF model. For a given query, we used our default setting for the ColBERT-PRF model, i.e. selecting ten

[5] Indeed, this is 8% higher than the highest MAP among all TREC 2019 participants [8].

Table 2: Comparison with baselines. Superscripts a...p denote significant improvements over the indicated baseline model(s). The highest value in each column is boldfaced. The higher MRT of BM25+docT5query+ColBERT is expected, as we do not have a ColBERT index for the docT5query representation.

	TREC 2019 (43 queries)					TREC 2020 (54 queries)				
	MAP	NDCG@10	MRR@10	Recall	MRT	MAP	NDCG@10	MRR@10	Recall	MRT
Lexical Retrieval Approaches										
BM25 (a)	0.2864	0.4795	0.6416	0.7553	132.4	0.2930	0.4936	0.5912	0.8103	129.1
BM25+BERT (b)	0.4441	0.6855	0.8295	0.7553	3588.98	0.4699	0.6716	0.8069	0.8103	3553.36
BM25+ColBERT (c)	0.4582	0.6950	0.8580	0.7553	201.9	0.4752	0.6931	0.8546	0.8103	203.2
BM25+RM3 (d)	0.3108	0.5156	0.6093	0.7756	201.4	0.3203	0.5043	0.5912	0.8423	247.7
BM25+RM3+BERT (e)	0.4531	0.6862	0.8275	0.7756	4035.01	0.4739	0.6704	0.8079	0.8423	4003.09
BM25+RM3+ColBERT (f)	0.4709	0.7055	0.8651	0.7756	319.9	0.4800	0.6877	**0.8560**	0.8423	228.2
Neural Augmentation Approaches										
BM25+DeepCT (g)	0.3169	0.5599	0.7155	0.7321	54.3	0.3570	0.5603	0.7090	0.8008	63.8
BM25+DeepCT+BERT (h)	0.4308	0.7011	0.8483	0.7321	3736.35	0.4671	0.6852	0.8068	0.8008	3718.29
BM25+DeepCT+ColBERT (i)	0.4416	0.7004	0.8541	0.7321	129.4	0.4757	0.7071	0.8549	0.8008	140.5
BM25+docT5query (j)	0.4044	0.6308	0.7614	0.8263	281.8	0.4082	0.6228	0.7434	0.8456	295.4
BM25+docT5query+BERT (k)	0.4802	0.7123	0.8483	0.8263	8024.90	0.4714	0.6810	0.8160	0.8456	3887.78
BM25+docT5query+ColBERT (l)	0.5009	0.7136	0.8367	0.8263	2361.88	0.4733	0.6934	0.8021	0.8456	2381.09
Dense Retrieval Models										
ANCE (m)	0.3715	0.6537	0.8590	0.7571	199.2	0.4070	0.6447	0.7898	0.7737	178.4
ColBERT E2E (n)	0.4318	0.6934	0.8529	0.7892	599.6	0.4654	0.6871	0.8525	0.8245	530.8
BERT-QE Reranking Models										
BM25 + RM3 + ColBERT + BERT-QE (o)	0.4832	0.7179	0.8754	0.7756	1129.88	0.4842	0.6909	0.8315	0.8423	1595.12
ColBERT E2E + BERT-QE (p)	0.4423	0.7013	0.8683	0.7892	1260.92	0.4749	0.6911	0.8315	0.8245	1327.88
ColBERT-PRF Models										
ColBERT-PRF Ranker (β=1)	**0.5431**^{abcdghijmnp}	0.7352^{adg}	0.8858^{ad}	0.8706^{abhmn}	4391.21	0.4962^{adgjm}	0.6993^{adg}	0.8396^{ad}	**0.8892**^{abghlmn}	4233.34
ColBERT-PRF ReRanker (β=1)	0.5040^{adgmnp}	0.7369^{adg}	0.8858^{ad}	0.7961	3598.23	0.4919^{adgj}	0.7006^{adg}	0.8396^{ad}	0.8431^{m}	3607.18
ColBERT-PRF Ranker (β=0.5)	0.5427^{abcdghijmnp}	0.7395^{adgjm}	**0.8897**^{ad}	**0.8711**^{abhmn}	4132.30	**0.5116**^{adgjmn}	0.7153^{adgj}	0.8439^{ad}	0.8837^{aghlmn}	4300.58
ColBERT-PRF ReRanker (β=0.5)	0.5026^{adgmnp}	**0.7409**^{adgjm}	**0.8897**^{ad}	0.7977	3576.62	0.5063^{adgjm}	**0.7161**^{adgj}	0.8439^{ad}	0.8443^{m}	3535.69

expansion embeddings; Equation (3) is used to resolve embeddings to tokens. We find that most of the expansion tokens identified are credible supplementary information for each user query and can indeed clarify the information needs. Overall, in response to RQ1, we conclude that our proposed ColBERT-PRF model is effective compared to the ColBERT E2E dense retrieval model.

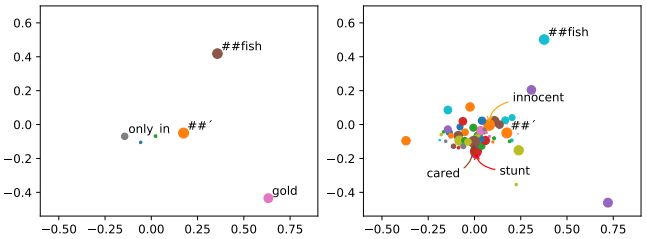
6.2 Results for RQ2

Next, to address RQ2(a)-(c), we analyse the performances of the ColBERT-PRF Ranker and ColBERT-PRF ReRanker approaches in comparison to different groups of baselines, namely sparse (lexical) retrieval approaches, neural augmented baselines, and BERT-QE.

For RQ2(a), we compare the ColBERT-PRF Ranker and ReRanker models with the lexical retrieval approaches. For both query sets, both Ranker and ReRanker provide significant improvements on all evaluation measures compared to the BM25 and BM25+RM3 models. This is mainly due to the more effective contexualised representation employed in the ColBERT-PRF models than the traditional sparse representation used in the lexical retrieval approaches. Furthermore, both ColBERT-PRF Ranker and ReRanker outperform the sparse retrieval approaches when reranked by either the BERT or the ColBERT models – e.g. BM25+(Col)BERT and BM25+RM3+(Col)BERT – on all metrics. In particular, ColBERT-PRF Ranker exhibits marked improvements over the BM25 with BERT or ColBERT reranking approach for MAP on the TREC 2019 queries. This indicates that our query expansion in the contextualised embedding space produces query representations that result in improved retrieval effectiveness. Hence, in answer to RQ2(a), we find that our proposed ColBERT-PRF models show significant improvements in retrieval effectiveness over sparse baselines.

For RQ2(b), on analysing the neural augmentation approaches, we observe that both the DeepCT and docT5query neural components could lead to effectiveness improvements over the corresponding lexical retrieval models without neural augmentation. However, despite their improved effectiveness, our proposed ColBERT-PRF models exhibit marked improvements over the neural augmentation approaches. Specifically, on the TREC 2019 query set, ColBERT-PRF Ranker significantly outperforms 4 out of 6 neural augmentation baselines and the BM25+DeepCT baseline on MAP. Meanwhile, both ColBERT-PRF Ranker and ReRanker exhibit significant improvements over BM25+DeepCT and BM25+docT5query on MAP for TREC 2020 queries, and exhibit improvements upto 9.5% improvements over neural augmentation approaches with neural reranking (e.g. MAP 0.4671→0.5116). On analysing these comparisons, the effectiveness of the ColBERT-PRF models indicates that the query representation enrichment in a contextualised embedding space leads to a higher effectiveness performance than the sparse representation document enrichment. Thus, in response to RQ2(b), the ColBERT-PRF models exhibit markedly higher performances than the neural augmentation approaches.

We further compare the ColBERT-PRF models with the recently proposed BERT-QE Reranking model. In particular, we provide results when using BERT-QE to rerank both BM25+RM3 as well as ColBERT E2E. Before comparing the ColBERT-PRF models with the BERT-QE rerankers, we first note that BERT-QE doesn't provide benefit to MAP on either query set, but can lead to a marginal improvement for NDCG@10 and MRR@10. However, the BERT-QE



(a) Cluster centroids, $K = 8$. **(b) Cluster centroids, $K = 64$.**

Figure 2: Embeddings selected using different number of clustering centroids K for the query 'do goldfish grow', the point size is representative of IDF.

reranker models still underperform compared to our ColBERT-PRF models. Indeed, ColBERT E2E+BERT-QE exhibits a performance significantly lower than both ColBERT-PRF Ranker and ReRanker on the TREC 2019 query set. Hence, in response to RQ2(c), we find that the ColBERT-PRF models significantly outperform the BERT-QE reranking models.

Finally, we consider the mean response times reported in Table 2, noting that ColBERT PRF exhibits higher response times than other ColBERT-based baselines, and similar to BERT-based re-rankers. There are several reasons for ColBERT PRF's speed: Firstly, the KMeans clustering of the feedback embeddings is conducted online, and the scikit-learn implementation we used is fairly slow – we tried other markedly faster KMeans implementations, but they were limited in terms of effectiveness (particularly for MAP), perhaps due to the lack of the KMeans++ initialisation procedure [5], which scikit-learn adopts; Secondly ColBERT PRF adds more expansion embeddings to the query - for the ranking setup, each feedback embedding can potentially cause a further $k' = 1000$ documents to be scored - further tuning of ColBERT's k' parameter may allow efficiency improvements for ColBERT-PRF without much loss of effectiveness, at least for the first retrieval stage. Overall, we contend that the effectiveness benefits exhibited by ColBERT-PRF demonstrate the promise of our approach, and hence we leave further studies of improving the efficiency of ColBERT-PRF to future work, such as through deploying efficient clustering algorithms (either online, i.e., per-query, or offline, i.e., based on the existing FAISS index), as well as more conservative second-pass retrieval settings.

6.3 Results for RQ3.

To address RQ3, we investigate the impact of the parameters of ColBERT-PRF. Firstly, concerning the number of clusters, K, and the number of expansion embeddings f_e selected from those clusters ($f_e \leq K$), Figures 3(a) and (b) report, for ColBERT-PRF Ranker and ColBERT-PRF ReRanker, respectively, the MAP (y-axis) performance for different f_e (x-axis) selected from K clusters (different curves). We observe that, with the same number of clusters and expansion embeddings, ColBERT-PRF Ranker exhibits a higher MAP performance than ColBERT-PRF ReRanker – as we also observed in Section 6.1.

Then, for a given f_e value, Figures 3(a) and (b) show that the best performance is achieved by ColBERT-PRF when using $K = 24$. To explain this, we refer to Figure 2 together with Figure 1(b), which both show the centroid embeddings obtained using different numbers of clusters K. Indeed, if the number of clusters K is too small, the informativeness of the returned embeddings would be limited. For instance, in Figure 2(a), the centroid embeddings represent stopwords such as 'in' and '##' are included, which are unlikely to be helpful for retrieving more relevant documents. However, if K is too large, the returned embeddings contain more noise, and hence are not suitable for expansion – for instance, using $K = 64$, feedback embeddings representing 'innocent' and 'stunt' are identified in Figure 2(b), which could cause a topic drift.

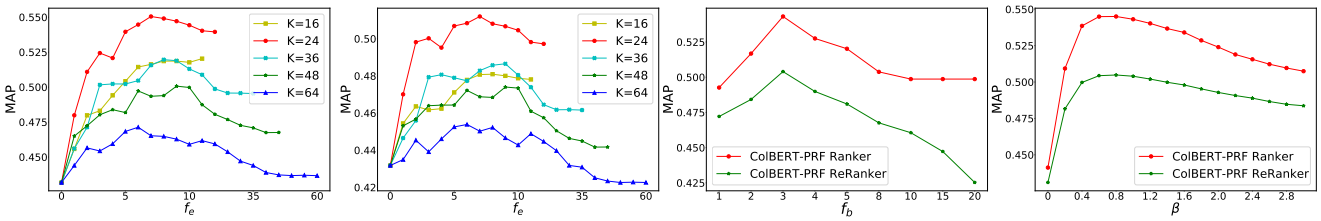
Next, we analyse the impact of the number of feedback documents, f_b. Figure 3(c) reports the MAP performance in response to different number of f_b for both ColBERT-PRF Ranker and ReRanker. We observe that, when $f_b = 3$, both Ranker and ReRanker obtain their peak MAP values. In addition, for a given f_b value, the Ranker exhibits a higher performance than the ReRanker. Similar to existing PRF models, we also find that considering too many feedback documents causes a query drift, in this case by identifying unrelated embeddings.

Finally, we analyse the impact of the β parameter, which controls the emphasis of the expansion embeddings during the final document scoring. Figure 3(d) reports MAP as β is varied for ColBERT-PRF Ranker and ReRanker. From the figure, we observe that in both scenarios, the highest MAP is obtained for $\beta \in [0.6, 0.8]$, but good effectiveness is maintained for higher values of β, which emphasises the high utility of the centroid embeddings for effective retrieval.

Overall, in response to RQ3, we find that ColBERT-PRF, similar to existing PRF approaches, is sensitive to the number of feedback documents and the number of expansion embeddings that are added to the query (f_b & f_e) as well as their relative importance during scoring (c.f. β). However, going further, the K parameter of KMeans has a notable impact on performance: if too high, noisy clusters can be obtained; too low and the obtained centroids can represent stopwords. Yet, the stable and effective results across the hyperparameters demonstrate the overall promise of ColBERT-PRF.

7 CONCLUSIONS

This work is the first to propose a contextualised pseudo-relevance feedback mechanism for dense retrieval. For multiple representation dense retrieval, based on the feedback documents obtained from the first-pass retrieval, our proposed ColBERT-PRF approach extracts representative feedback embeddings using a clustering technique. It then identifies discriminative embeddings among the representative embeddings and appends them to the query representation. The ColBERT-PRF model can be effectively applied in both ranking and reranking scenarios, and requires no further neural network training beyond that of ColBERT. Indeed, our experimental results – on the TREC 2019 and 2020 Deep Learning track passage ranking query sets – show that our proposed approach can significantly improve the retrieval effectiveness of the state-of-the-art ColBERT dense retrieval approach. Our proposed ColBERT-PRF model is a novel and extremely promising approach into applying PRF in dense retrieval. It may also be adaptable to further multiple representation dense retrieval approaches beyond ColBERT. In future work, we plan to verify the effectiveness of ColBERT-PRF on test collections with longer documents and further explore variations of ColBERT-PRF, for instance replacing the clustering algorithm with



(a) Impact of K and f_e on ColBERT-PRF Ranker. **(b) Impact of K and f_e on ColBERT-PRF ReRanker.** **(c) Impact of pseudo-relevant feedback size f_b.** **(d) Impact of expansion embedding weight β.**

Figure 3: MAP on the TREC 2019 query set while varying the number of clusters (K), number of expansion embeddings (f_e), as well as the feedback set size f_b and expansion embedding weight β. $\beta = 0$ & $f_e = 0$ correspond to the original ColBERT.

Table 3: Examples of the expanded queries by the ColBERT PRF model on the TREC 2019 & 2020 query sets. The symbol | denotes that there are multiple tokens that are highly likely for a particular expansion embedding.[6]

	Original query terms	Original query tokens	Most likely tokens for expansion embeddings
TREC 2019 queries	what is a active margin	what is a active margin	(by|opposition) oceanic volcanoes ##cton (margin|margins) (breeds|##kshi) continental plate an each
	what is wifi vs bluetooth	what is wi ##fi vs blue ##tooth	##tooth (breeds|##kshi) phones devices wi ##fi blue systems access point
	what is the most popular food in switzerland	what is the most popular food in switzerland	##hs (swiss|switzerland) (influences|includes) (breeds|##kshi) potato (dishes|food) (bologna|hog) cheese gr (italians|french)
TREC 2020 queries	what is mamey	what is ma ##me ##y	(is|upset) (breeds|##kshi) flesh sap ##ote fruit ma ##me (larger|more) central
	average annual income data analyst	average annual income data analyst	(analyst|analysts) (breeds|##kshi) (55|96) (grow|growth) salary computer tax 2015 depending ##k
	do google docs auto save	do google doc ##s auto save	(breeds|##kshi) doc (to|automatically) google document save (saves|saved) drive (changes|revisions) (back|to)

more efficient variants, or replacing the token-level IDF calculation for identifying discriminative embeddings.

ACKNOWLEDGEMENTS

Nicola Tonellotto was partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the Cross-Lab project (Departments of Excellence). Xiao Wang acknowledges support by the China Scholarship Council (CSC) from the Ministry of Education of P.R. China. Craig Macdonald and Iadh Ounis acknowledge EPSRC grant EP/ R018634/1: Closed-Loop Data Science for Complex, Computationally- & Data-Intensive Analytics.

REFERENCES

[1] 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*.
[2] Giambattista Amati. 2003. Probability models for information retrieval based on divergence from randomness Ph.D. thesis. *University of Glasgow* (2003).
[3] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. 2004. Query difficulty, robustness, and selective application of query expansion. In *Proceedings of ECIR*. 127–137.
[4] 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 (2002), 357–389.
[5] David Arthur and Sergei Vassilvitskii. 2007. K-Means++: The Advantages of Careful Seeding. In *Proceedings of SODA*. 1027–1035.
[6] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of SIGIR*. 243–250.
[7] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. In *Proceedings of TREC*.
[8] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the TREC 2019 deep learning track. In *Proceedings of TREC*.
[9] Zhuyun Dai and Jamie Callan. 2020. Context-aware document term weighting for ad-hoc search. In *Proceedings of WWW*. 1897–1907.
[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of ACL*. 4171–4186.
[11] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query Expansion with Locally-Trained Word Embeddings. In *Proceedings of ACL*. 367–377.
[12] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. A White Box Analysis of ColBERT. In *Proceedings of ECIR*.
[13] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of CIKM*. 55–64.
[14] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).
[15] 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*. 6769–6781.
[16] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of SIGIR*. 39–48.
[17] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of CIKM*. 1929–1932.
[18] 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*. 4482–4491.
[19] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2021. PARADE: Passage Representation Aggregation for Document Reranking. arXiv:2008.09093 [cs.IR]
[20] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. In *Proceedings of TACL*.
[21] Sean MacAvaney. 2020. OpenNIR: A Complete Neural Ad-Hoc Ranking Pipeline. In *WSDM 2020*.
[22] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation in Information Retrieval Using PyTerrier. In *Proceedings of ICTIR*.
[23] Shahrzad Naseri, Jeffrey Dalton, Andrew Yates, and James Allan. 2021. CEQE: Contextualized Embeddings for Query Expansion. *Proceedings of ECIR* (2021).
[24] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* (2019).

[25] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).
[26] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. 2005. Terrier information retrieval platform. In *Proceedings of ECIR*. 517–519.
[27] Ramith Padaki, Zhuyun Dai, and Jamie Callan. 2020. Rethinking query expansion for BERT reranking. In *Proceedings of ECIR*. 297–304.
[28] Joseph Rocchio. 1971. Relevance feedback in information retrieval. *The Smart Retrieval System-experiments in Automatic Document Processing* (1971), 313–323.
[29] Dwaipayan Roy, Sumit Bhatia, and Mandar Mitra. 2019. Selecting Discriminative Terms for Relevance Model. In *Proceedings of SIGIR*. 1253–1256.
[30] Dwaipayan Roy, Debasis Ganguly, Sumit Bhatia, Srikanta Bedathur, and Mandar Mitra. 2018. Using word embeddings for information retrieval: How collection and term normalization choices affect performance. In *Proceedings of CIKM*. 1835–1838.
[31] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. In *Proceedings of SIGIR Workshop on Neural Information Retrieval*. arXiv:1606.07608.
[32] Junmei Wang, Min Pan, Tingting He, Xiang Huang, Xueyan Wang, and Xinhui Tu. 2020. A Pseudo-relevance feedback framework combining relevance matching and semantic matching for information retrieval. *Information Processing & Management* 57, 6 (2020), 102342.
[33] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of SIGIR*. 55–64.
[34] 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*.
[35] HongChien Yu, Zhuyun Dai, and Jamie Callan. 2021. PGT: Pseudo Relevance Feedback Using a Graph-Based Transformer. In *Proceedings of ECIR*.
[36] Hamed Zamani and W Bruce Croft. 2016. Embedding-based query language models. In *Proceedings of ICTIR*. 147–156.
[37] 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*. 4718–4728.

[6] The expansion embedding '(breeds|##kshi)', which appears for each query, is thought to be close to the embedding of the [D] token, which ColBERT places in each document.