Personalized Query Expansion with Contextual Word Embeddings

ELIAS BASSANI, Independent, Italy
NICOLA TONELLOTTO, University of Pisa, Italy
GABRIELLA PASI, University of Milano-Bicocca, Italy

Personalized Query Expansion, the task of expanding queries with additional terms extracted from the user-related vocabulary, is a well-known solution to improve the retrieval performance of a system w.r.t. short queries. Recent approaches rely on word embeddings to select expansion terms from user-related texts. Although promising results have been delivered with former word embedding techniques, we argue that these methods are not suited for contextual word embeddings, which produce a unique vector representation for each term occurrence.

In this article, we propose a Personalized Query Expansion method designed to solve the issues arising from the use of contextual word embeddings with the current Personalized Query Expansion approaches based on word embeddings. Specifically, we employ a clustering-based procedure to identify the terms that better represent the user interests and to improve the diversity of those selected for expansion, achieving improvements of up to 4% w.r.t. the best-performing baseline in terms of MAP@100. Moreover, our approach outperforms previous ones in terms of efficiency, allowing us to achieve sub-millisecond expansion times even in data-rich scenarios. Finally, we introduce a novel metric to evaluate the expansion terms' diversity and empirically show the unsuitability of previous approaches based on word embeddings when employed along with contextual word embeddings, which cause the selection of semantically overlapping expansion terms.

CCS Concepts: • **Information systems** → **Query reformulation**; **Personalization;**

Additional Key Words and Phrases: Personalization, Query Expansion, contextual word embeddings, dense retrieval

E. Bassani's work was done when affiliated with Consorzio per il Trasferimento Tecnologico C2T and the University of Milano-Bicocca.

This work was partially supported by the Ministero dell'Università e della Ricerca under the grant "Dipartimenti di Eccellenza 2023-2027" of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca, Italy. This work is supported, in part, by the spoke "FutureHPC & BigData" of the ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, the Spoke "Human-centered AI" of the M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research", funded by European Union – NextGenerationEU, and the FoReLab project (Departments of Excellence), funded by the Ministero dell'Istruzione e del Merito of Italy.

Authors' addresses: E. Bassani, Independent, Milan, Italy; e-mail: e.bassani3@campus.unimib.it; N. Tonellotto, University of Pisa, Pisa, Italy; e-mail: nicola.tonellotto@unipi.it; G. Pasi, University of Milano-Bicocca, Milan, Italy; e-mail: gabriella.pasi@unimib.it.

1046-8188/2023/12-ART61 $15.00
https://doi.org/10.1145/3624988

ACM Reference format:
Elias Bassani, Nicola Tonellotto, and Gabriella Pasi. 2023. Personalized Query Expansion with Contextual Word Embeddings. *ACM Trans. Inf. Syst.* 42, 2, Article 61 (December 2023), 35 pages.
https://doi.org/10.1145/3624988

1 INTRODUCTION

Nowadays, most search engines provide users with a simple interface to specify their information needs through short keyword-based queries, which are usually two to three terms long in the case of Web search [42, 92]. However, as a query only broadly describes a user's information need, search engines may struggle to provide satisfactory results. Multiple factors related to how users choose terms for their queries can affect a system's retrieval effectiveness [3]. For example, the terms composing a query can be related to multiple topics, leading the system to provide results not focused on the user's topic of interest. Moreover, out of habit, users often issue queries too short to clearly express complex information needs, ultimately failing to find documents valuable to fulfill them. Finally, users sometimes have only a broad idea of the information they need, and hence they issue queries that are not appropriate to find documents that can answer their information needs. A well-known technique proposed to overcome those issues is Query Expansion, whereby the user's original query is augmented with new terms, known as *expansion terms*, to improve the system's effectiveness. The identification of proper expansion terms aims to clarify the user's search intent and bridges the gap between the original query terms and the documents' vocabulary [23], addressing the well-known vocabulary mismatch problem [34]. Query Expansion techniques can leverage user-related information previously gathered to derive the expansion terms, in which case we talk about Personalized Query Expansion [2, 9, 10, 14, 21, 25, 36, 43, 46, 50, 68, 71, 83, 98, 108, 109]. Personalized Query Expansion techniques rely on user-related documents, such as previously accessed Web pages and user-generated content [49], e.g., product reviews or tweets, to extract expansion terms directly from the users' vocabulary or the vocabulary used in documents of their interest.

Historically, most approaches to Personalized Query Expansion [8–10, 14, 68, 98, 107] focused on leveraging social information derived from folksonomy platforms[1] to extract expansion terms. On those platforms, like the former social bookmarking website `del.icio.us`,[2] users apply public tags to online items, such as Web pages. Works in this area addressed the selection of personalized expansion terms by relying on term co-occurrence-based approaches and social relations analyses.

More recently, to overcome the limitations of lexical matching-driven term co-occurrence analysis, which suffers from the vocabulary mismatch problem, researchers started experimenting with word embedding models, which project text into dense low-dimensional vector spaces where the semantic similarity among terms can be computed as the cosine similarity of their vector representations. Existing approaches [2, 50, 108] rely on the well-known Word2Vec model [66, 67] to generate word embeddings for both queries and user-related texts and on cosine similarity to evaluate their semantic relatedness, which they use to select the personalized expansion terms. A limitation of the word embeddings produced by Word2Vec and similar models is that terms are always mapped to the same vectors regardless of their context, which usually varies for each occurrence of a term.

Recently, to overcome the limitations of traditional word embeddings, new techniques [20, 28, 74, 75] have been introduced. These new methods map each word occurrence to a unique

[1] https://en.wikipedia.org/wiki/Folksonomy
[2] https://en.wikipedia.org/wiki/Delicious_(website)

representation based on its surrounding terms, thereby capturing the different meanings it can assume across varied contexts. The new representations are commonly called *contextual* word embeddings and have allowed reaching a new state of the art in many different Natural Language Processing tasks. In the past few years, contextual word embeddings have also been successfully applied to Information Retrieval [54, 59, 60, 91], advancing the state of the art in multiple tasks and opening new opportunities and challenges for retrieval-enhancing tasks, such as Personalized Query Expansion. In the following, we use the locutions "word embedding" to refer to "word embedding techniques" and "term embedding" to refer to the actual vector representation computed by one of those techniques for a given term.

In this article, we address Personalized Query Expansion using contextual word embeddings, which, as mentioned above, open new opportunities for this task while also introducing new challenges. We argue that two of the main challenges in employing contextual word embeddings to select expansion terms are (1) reducing redundancy among expansion terms and (2) addressing scalability issues. Previous Personalized Query Expansion methods based on word embeddings [2, 50, 108] rely on ranking functions based on cosine similarity to rank all the user-related terms before selecting those for query expansion. However, when working with contextual word embedding models, which produce a unique embedding for each term occurrence, we could end up with very similar embeddings for multiple occurrences of the same term appearing in similar contexts. We argue that, if not carefully handled, this aspect of contextual word embeddings could cause the selection of multiple expansion term embeddings with very close—if not identical—semantic meanings, thus reducing the potential utility of Query Expansion. Because of the lack of a mechanism accounting for the presence of multiple, very similar embeddings, previous methods have a high probability of selecting expansion terms that are redundant with each other, as we will show. This could cause the expansion terms to give strong prominence to a single aspect of the query instead of covering as many query aspects as possible, thus negatively affecting the diversification of the search results [17–19, 55]. Moreover, as previous approaches [2, 50, 108] rely on computing a similarity score between the query and each user-related term embedding to select those most appropriate for expanding the query, they could introduce an overhead proportional to the number of candidate expansion terms. While with traditional word embeddings, such as Word2Vec, each unique term is represented once, with contextual word embeddings, we have a different representation for each term occurrence, potentially making the problem much more severe. This issue could cause query expansion methods based on contextual word embeddings to suffer a low-scalability problem, making their application in data-rich real-world scenarios debatable, such as in Web Search, where we could leverage very long user browsing histories to conduct personalization.

In this article, we present PQEWC (pronounced *"quick"*), a **P**ersonalized **Q**uery **E**xpansion method designed to work **W**ith **C**ontextual word embeddings. To address the scalability issues arising from the adoption of contextual word embeddings in Personalized Query Expansion and to reduce the impact of potentially redundant expansion terms, we employ an offline clustering-based procedure aiming at grouping the user-related terms and identifying those that better represent the user interests. By selecting only the expansion term that better represents the user interests w.r.t. the current query from each cluster, we avoid adding to the query multiple expansion terms with similar semantic meanings, thus reducing the chance of expanding it with redundant expansion terms. Finally, we implement an approximation mechanism for selecting the expansion terms, which allows our proposed approach to achieve a sub-millisecond expansion time even in very data-rich scenarios, making it suitable for many real-world applications.

We acknowledge several previous efforts devoted to modeling user interests for personalization purposes [33, 65, 84]. As later described, our proposed approach to Personalized Query Expansion involves some steps that could be regarded as lightweight user modeling, i.e., representing the topics a user is most interested in. Such a light user representation is functional to our approach, i.e., dealing with the issues arising from the adoption of contextual word embeddings in Personalized Query Expansion, as previously described in this section. Therefore, we leave for the future the adaptation and enhancement of previously proposed user modeling techniques or the proposal of new ones to further boost the retrieval effectiveness of Personalized Query Expansion. To guarantee a fair comparison with the baselines, in the comparative evaluation reported in this manuscript, we relied on the same user-related data for each of the considered approaches without leveraging additional information (e.g., social networks), which would have made it unclear where the improvements (if any) come from.

The rest of the article is organized as follows. Section 2 discusses the related works and positions our work w.r.t. them. In Section 3, we present our novel Personalized Query Expansion approach and discuss our design choices. Section 4 introduces the retrieval task we tackled to evaluate our proposal. Section 5 presents our research questions and describes the experimental setup of our comparative evaluation. Finally, in Section 6, we compare our proposed approach and other query expansion methods at the state of the art [50, 96, 108] in terms of both effectiveness and efficiency and ablate our design choices. The results of our evaluation clearly show the advantages of PQEWC w.r.t. the considered query expansion baselines, which are outperformed in both retrieval effectiveness and efficiency. Across all the considered datasets, the proposed approach improves by up to 5% in terms of MAP@100 over our base retrieval system, based on BM25 [79] and ColBERT [47], and by up to 4% w.r.t. the best-performing baseline [96]. We share all the code to reproduce the experimental evaluation we conducted.[3]

2 RELATED WORK

Query Expansion is a well-established technique in Information Retrieval. It has received significant attention from the research community in the past few decades and continues to attract many researchers. In this section, we first cover the state of the art of Query Expansion, and then we focus on its Personalized counterpart. In both cases, we pay particular attention to the methods based on word embeddings.

2.1 Query Expansion

Among the several approaches proposed for Query Expansion [3, 23], a line of research that still attracts the research community's interest is represented by the methods founded on the pseudo-relevance feedback technique [80]. These methods [22, 24, 27, 41, 53, 58, 101] rely on a first retrieval stage to collect the so-called feedback documents, i.e., documents appearing in the top positions of the ranked list of documents, which are assumed to be relevant w.r.t. the query and from which terms to expand the initial query are extracted. Query Expansion methods based on pseudo-relevance feedback have proved their effectiveness over the years and are still relevant today. The most important of these models is RM3 [41], which leverages statistical information on the occurrences of terms in feedback documents and in the corpus to select the expansion terms. Intuitively, RM3 expands the initial query with terms that are frequent in the feedback documents and infrequent in the corpus.

With the advent of word embedding techniques, new Query Expansion methods leveraging semantic term representations have been proposed [29, 51, 81]. Instead of exploiting the pseudo-relevance feedback documents using statistical methods to select the expansion terms, those methods choose them by evaluating the semantic similarity between the query terms and the corpus vocabulary. Generally, they expand a query with the closest terms in the word embedding space,

[3] https://github.com/AmenRa/pqewc

i.e., the most semantically similar terms. For example, Kuzi et al. [51] propose to use the Word2Vec model [66, 67] to compute latent representations of all terms appearing in the corpus on which the search is conducted, and to apply cosine similarity to select expansion terms that are semantically related to the query. Similarly, Roy et al. [81] rely on Word2Vec to obtain term embeddings for their corpus vocabulary. To select the expansion terms for a given query, the authors employ a k-nearest neighbor method based on cosine similarity. The authors found that their approach could improve over their underlying retrieval model without expansion but not over the pseudo-relevance feedback expansion model RM3. Diaz et al. [29] investigate whether training word embedding models such as Word2Vec and GloVe [73] *locally*, i.e., on the available test collection, instead of using *globally* trained models, i.e., models trained on general domain-agnostic texts, can benefit Query Expansion. The authors found locally trained word embeddings to generally improve the performance of Query Expansion w.r.t. globally trained word embeddings. The most significant drawback of those methods, which do not deliver significant improvements over the pseudo-relevance feedback approaches, is the lack of a mechanism to identify the most prominent terms from a retrieval perspective, i.e., the terms that allow improving the identification of the relevant documents, as only the terms' semantic relatedness is considered. Moreover, after expansion, the authors rely on traditional retrieval models based on lexical term matching, which notoriously do not account for semantic relatedness.

More recently, the contextual word embedding techniques renovated the research community's interest in Query Expansion, and novel approaches based on this new kind of embedding were proposed [69, 96, 106]. The authors of new approaches, aware of the limitations of previous methods based on word embeddings, combine the new contextual word embedding techniques with the pseudo-relevance feedback approach.

Zheng et al. [106] propose a novel Query Expansion method based on contextual word embeddings that leverage a BERT-based [28] re-ranker [70] in a pseudo-relevance feedback fashion. After a first re-ranking round, the most relevant text chunks are extracted from the top re-ranked documents and used to compute additional relevance scores for the documents. Finally, the newly computed relevance scores are aggregated with the original ones to obtain the scores to compute the final documents ranking. Their experimental evaluation shows that the proposed model delivers promising retrieval effectiveness improvements. Naseri et al. [69] revisit the pseudo-relevance feedback approaches to Query Expansion by employing the similarity between the query's contextual word embeddings and those of the feedback documents in deriving probability values to use in place of those of the original formulation. Although improving over non-contextual Query Expansion methods based on word embeddings, the model proposed by Naseri et al. [69] only performs on par with the classic expansion method RM3 [41]. Wang et al. [96] have recently introduced ColBERT-PRF, a novel query expansion method based on the neural retrieval model ColBERT [47] and pseudo-relevance feedback. After a first ranking stage with ColBERT, this method leverages Kmeans clustering [62] to group the term embeddings of a certain number of feedback documents. Then, it selects the tokens corresponding to the cluster centroids with higher Inverse Document Frequency scores [44] for expanding the original query. The authors report encouraging improvements over ColBERT without query expansion as well as many other baselines. Unfortunately, despite some promising improvements in terms of retrieval effectiveness, previous Query Expansion methods based on contextual word embeddings suffer from poor efficiency [96], limiting their applicability in real-world applications.

2.2 Personalized Query Expansion

In the early 2000s, the increasing popularity of social tagging systems, where users can associate public tags with online items such as Web pages, attracted some attention from the research community thanks to the large amount of accessible data provided by those platforms. In particular, researchers leveraged those data to derive test beds for Personalized Information Retrieval in social-network-like environments [11, 12, 15, 16, 95, 100, 107]. Among the approaches for personalization proposed in this period, several Personalized Query Expansion methods were presented [8–10, 13, 14, 68, 98, 107]. Most of the works in this area approach the selection of personalized expansion terms by leveraging both term co-occurrence statistics and social relations among the users. For example, Bender et al. [8], Bertier et al. [9], Mulhem et al. [68], and Wu et al. [98] derive terms for Personalized Query Expansion by leveraging the relations and similarities among users, documents, and tags. Biancalana and Micarelli [10] propose a method for selecting expansion terms based on a three-dimensional co-occurrence matrix from which the authors derive relations among the query terms appearing in a document, terms associated with similarly tagged documents, and those appearing in user-related documents, among which the authors select the expansion terms. Bouadjenek et al. [13, 14] approach Personalized Query Expansion by employing a combination of social proximity and semantic similarity to identify the terms similar to those mostly used by a given user and his or her social relatives. Unfortunately, we found most of the previous works to lack comparisons with other Personalized and non-Personalized Query Expansion methods, making it difficult to draw general conclusions about their effectiveness.

Other than the works related to social tagging systems, the literature comprises some approaches leveraging other contextual data. For example, Zhu et al. [109] leverage the co-occurrences of the query terms with terms from user-related documents located in their desktop environment to select personalized expansion terms. Some works focus on building ontology-based user profiles from previous queries formulated by the user [21] and previously accessed documents [36]. Palleti et al. [71] build user profiles by leveraging collaborative information approaches and derive personalized expansion terms from those. Chirita et al. [25] exploit local user-related information to derive personalized terms and expand the queries before submitting them to Web search engines. This way, the authors preserve the users' privacy and anonymity while enhancing their Web search experience. Sarwar et al. [83] leverage users' status messages from social networks to identify personalized expansion terms for their queries. The authors first retrieve the most relevant status messages with BM25 and then select from those the expansion terms relying on their Inverse Document Frequency. Again, those works lack comparisons with other Personalized and non-Personalized Query Expansion methods.

More recently, some researchers have addressed Personalized Query Expansion using word embeddings [2, 50, 108]. Similarly to previous works leveraging non-contextual word embeddings for Query Expansion, the authors mostly employ term embeddings computed with Word2Vec and evaluate cosine similarity to assess the semantic relatedness of the query terms and, in this case, the user vocabulary to select the expansion terms. Amer et al. [2] conduct an exploratory study on the use of Word2Vec's term embeddings for Personalized Query Expansion. Specifically, they compare the performance of a Query Expansion method that selects expansion terms based on their cosine similarity with the query term embeddings when employing locally trained embeddings, i.e., embeddings trained individually for each user only on the specific user-related texts, and globally trained embeddings, i.e., embeddings trained on the whole corpus. Similarly to previous Query Expansion methods based on word embeddings, the authors employ the term embeddings only during the expansion process and rely on a Language Model with Dirichlet smoothing [104] as their retrieval model. The authors report that the expansion methods did not improve the retrieval effectiveness of the original queries, and the globally trained embeddings outperformed the locally trained ones. Kuzi et al. [50] address the Personalized Query Expansion task in the context of e-mail search. Similarly to the work by Amer et al. [2], the authors compared a Query

Expansion method based on word embeddings with globally trained word embeddings and locally trained ones with the pseudo-relevance feedback expansion model RM1 [52]. The authors report findings similar to those of Amer et al. [2], but the personalized variant of their Query Expansion method based on word embeddings allows to improve the performance of the original queries. Zhou et al. [108] focus on enriching user profiles with information from external sources and propose two Personalized Query Expansion methods based on word embeddings and topic modeling. The model based on word embeddings ranks the user-related term embeddings by their cosine similarity with the sum of the query term embeddings and selects the top n for expansion. The authors report good results on folksonomy-based datasets for both the proposed models.

As we reported for the other works about Personalized Query Expansion, most of the authors of methods based on word embeddings did not compare their approaches with other Personalized Query Expansion methods. We argue that the lack of standard Personalized Search test collections [6] and of publicly available implementations for *all* the presented Personalized Query Expansion methods poses severe issues in determining the state of the art in this context. However, we highlight that many proposals, such as those based on social interactions, are of difficult application in domains different from the original ones.

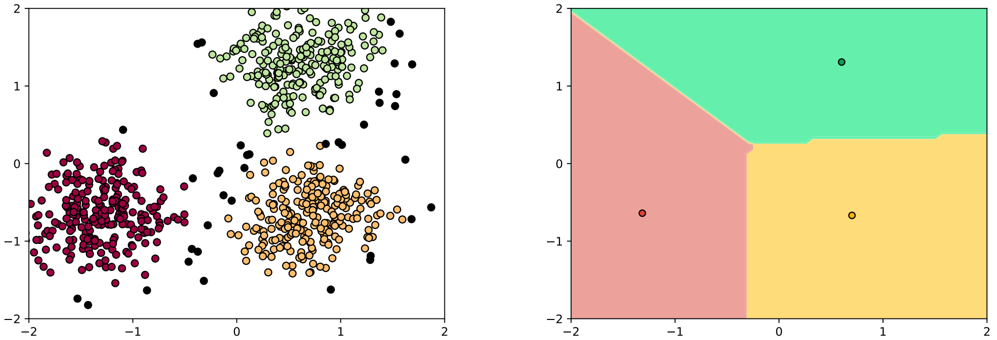
In this work, we focus on the adoption of contextual word embeddings in Personalized Query Expansion. More specifically, to overcome the limitations of both semantic and lexical methods previously reported, we propose an approach that combines the usage of contextual word embeddings with a clustering-based procedure, which allows for identifying the topic of interest for each user and the terms that better represent the user's specific preferences. Moreover, we also pay particular attention to the efficiency issue affecting the Query Expansion methods based on contextual word embeddings reported in the previous non-personalized works [69, 96, 106] and propose an approximation procedure that allows our approach to achieve a sub-millisecond expansion time and to scale even in very data-rich scenarios. For reproducibility, we conduct our experimental evaluation on publicly available datasets (see Section 5.1), and we share the code of both the implementation of the novel Personalized Query Expansion approach we present in Section 3 and those of the baselines (Section 5.2).

3 THE PROPOSED APPROACH

In this section, we present PQEWC,[4] the method we propose to tackle the challenges introduced by the adoption of contextual word embeddings in Personalized Query Expansion, as described in Sections 1 and 2. First, in Section 3.1, we describe the approach we propose to identify the embeddings most representative of the user's interests that are also discriminative from a retrieval perspective (i.e., the embeddings that allow identifying documents relevant w.r.t. the query and the user preferences), and show how to avoid selecting multiple expansion terms with similar semantic meanings. Then, we introduce our expansion term selection strategy and an effective mechanism to drastically reduce the number of computations required in the expansion term selection stage (Section 3.2). Finally, in Section 3.3, we introduce ColBERT [47], a recent state-of-the-art retrieval model, which we enhance with our Personalized Query Expansion approach. We also discuss how we compute the relevance score of a document w.r.t. an expanded query.

In the following, we assume to have gathered for each user a set of related textual content, such as documents authored by the user, previously accessed web pages, user-generated content [49] (e.g., product reviews or tweets), previously issued queries, or other kinds of textual content related to the user.

[4] **P**ersonalized **Q**uery **E**xpansion **W**ith **C**ontextual word embeddings, pronounced *"quick"*.



(a) Clusters identified by HDBSCAN.

(b) Space partitioning as a result of nearest centroid classifier.

Fig. 1. Embedding space partitioning example.

3.1 Term Embeddings Representative of the User Interests

In this section, we introduce the first step of our proposed approach, which aims at identifying the term embeddings that better represent the interests of a specific user that are also discriminative from a retrieval perspective (i.e., the embeddings that allow identifying documents relevant w.r.t. the query and the user preferences).

The method we propose aims at pre-computing bags of candidate personalized expansion terms (in the form of term embeddings), among which we select those most related to the current search performed by the user (see Section 3.2). To identify the specific user's interests, we first partition the embedding space into regions where embeddings with similar semantic meanings lie. As exemplified in Figure 1, we do this in two steps. First, we group the embeddings of all terms in the document collection using the hierarchical density-based clustering method HDBSCAN [64] (Figure 1(a)). Then, we instantiate a nearest centroid classifier [90] (a.k.a. Rocchio classifier [63]) defined upon the clusters identified by HDBSCAN, thus partitioning the embedding space following the document collection's topic distribution in that same latent semantic space (Figure 1(b)). By relying upon this classifier, we group the user-related term embeddings according to the document collection's topics associated with the clusters identified by HDBSCAN. We adopt a density-based clustering method (HDBSCAN) instead of the more commonly used centroid-based methods, such as k-means [62], because those latter methods require defining the number of centroids/clusters a priori, which can be problematic to estimate. Moreover, finding an optimal configuration for this parameter, for example, employing the elbow method [89], can be computationally expensive.

To identify the clusters that better capture the specific user interests and contain discriminative embeddings, we propose a function $\phi : C_u \rightarrow \mathbb{R}$ (where C_u is the set of the clusters related to a user u) inspired by the TF-IDF formula [82] and defined as follows:

$$\phi(c_{ui}) = \frac{|c_{ui}|}{\sum_{j=1}^{k} |c_{uj}|} \cdot \log \frac{\sum_{j=1}^{k} |c_j|}{|c_i|}, \tag{1}$$

where k is the number of the latent semantic space regions identified by the application of HDBSCAN and the nearest centroid classifier to the embeddings of the document collection, $c_{uj} \in C_u$ is the set corresponding to the cluster of the user u's term embeddings lying in the semantic space region j, and c_j is the set corresponding to the cluster of the collection's term embeddings lying in that same region. Similarly, c_{ui} and c_i are the sets corresponding to the cluster of the user u's term

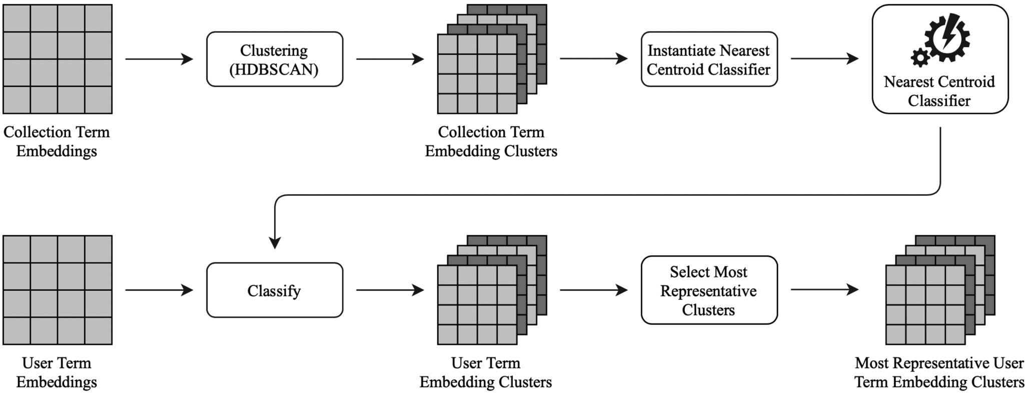

Collection Term Embeddings
Clustering (HDBSCAN)
Collection Term Embedding Clusters
Instantiate Nearest Centroid Classifier
Nearest Centroid Classifier
User Term Embeddings
Classify
User Term Embedding Clusters
Select Most Representative Clusters
Most Representative User Term Embedding Clusters

Fig. 2. Offline step: user term embedding clustering.

embeddings and the cluster of the collection's term embeddings that lie in the semantic space region i, respectively. The first part of the formula, inspired by the Term-Frequency [57], expresses the percentage of the user's term embeddings that lie within a specific latent region (identified by HDBSCAN and the nearest centroid classifier). We interpret this value as the user's interest in the topic associated with the region c_i of the semantic space. The second part of the formula expresses the specificity of a topic (represented here by a term embedding cluster and its corresponding latent region), quantified as the inverse function of the number of term embeddings of the collection that lie in the related region c_i of the latent space. We use this quantity to weigh the user interest in a specific topic w.r.t. its discriminative power, similarly to the Inverse Document Frequency [44] in the TF-IDF formulation. We use the function ϕ to rank and identify the top n clusters of the user's term embeddings from which we select the expansion terms at query time, as will be discussed in the next section. This process is conducted for each user separately. The whole procedure is shown in Figure 2.

We identify the clusters most representative of the user interests—from which we extract the expansion term embeddings—independently from their semantic similarity with the query to limit the selection of expansion terms that could be redundant w.r.t. the query terms. This way, we promote the complementarity of the user-related information carried by the expansion terms w.r.t. the information need expressed by the query. We cluster the term embeddings of the document collection instead of separately generating the term clusters from each user's vocabulary so that the user-term clusters reflect the distribution of the collection's topics in the embedding space. Moreover, as the clustering procedure is independent of the number of term embeddings related to each user, we do not incur to generate low-quality clusters when a user has few associated term embeddings. As we will show in Section 6.4, both these choices allow us to achieve better results than their respective counterparts. Finally, using a clustering-based approach to pre-compute bags of candidate expansion terms for each user has two beneficial effects. First, it reduces the chances of expanding the queries with terms that are redundant to each other, as we assign semantically close terms to the same clusters and select only one expansion term per cluster, as later discussed in Section 3.2. By reducing the redundancy among the expansion terms, we also avoid exacerbating a single aspect of the query. Second, by only considering the top n most representative clusters for each user, we reduce the computations required to choose the expansion terms at query time as the number of candidate expansion terms is drastically lower than the total number of user-related terms, thus allowing to achieve far better efficiency and greater scalability than other recent Query Expansion methods (see Section 6.2).

ALGORITHM 1: Expansion Term Selection

Require: List of the top n term embedding clusters C_u associated to the user u. List of the query term embeddings Q.

```
function SELECT_EXPANSION_TERM_EMBEDDINGS($C_u$, $Q$)
    exp_term_embs ← new List
    for all $c \in C_u$ do                  ▹ $c$ is one of the top $n$ embedding clusters of the user
        max_sims ← new List
        for all $t \in c$ do                ▹ $t$ is a user's term embedding belonging to $c$
            sims ← new List
            for all $q \in Q$ do            ▹ $q$ is the embedding of a query term
                $sim$ ← cos($t$, $q$)
                sims.push($sim$)
            $max\_sim$ ← max(sims)
            max_sims.push($max\_sim$)
        $i$ ← argmax(max_sims)              ▹ index of the highest max sim
        exp_term_embs.push($c[i]$)
    return exp_term_embs
```

3.2 Selection of Expansion Terms

In this section, we introduce the procedure we propose to select the expansion term embeddings from the user-related clusters, and the approximation we employ to drastically decrease its computation time.

Once we have built and identified the most representative clusters for a specific user following the method presented in the previous section, we select from each of them the term embedding with the highest maximum cosine similarity to the query term embeddings and employ the selected ones for expansion. In other words, given a user-related cluster, for each term embedding in the cluster we compute its cosine similarity with each query term embedding and take the *maximum* value. Then, we rank the term embeddings in the cluster according to their maximum similarity score and pick the one with the highest value. We repeat this operation for each of the most representative user-related clusters identified following the method presented in the previous section. The expansion term selection procedure is summarized by Algorithm 1.

During the selection of expansion term embeddings, we are only interested in the maximum similarity value between each user term embedding and query term embedding. Therefore, all the comparisons that do not produce a maximum similarity score are potentially unnecessary. As we cannot predict which comparison will result in a maximum similarity value without actually performing all of them, we propose to approximate our selection procedure to maximize efficiency. As reported in Algorithm 2, we first associate each user-related cluster with the query term embedding closer to the cluster centroid. Then, from each cluster, we select the user term embedding most similar to the query term embedding associated with that cluster to expand the query. This way, we considerably reduce the number of comparisons needed to select the Personalized Query Expansion terms, while leaving the effectiveness practically unaltered, as later shown in Section 6.4. For example, given a user with 16 associated term clusters of 128 terms each and a query representation composed of four embeddings, we reduce the number of comparisons from $16 \times 4 \times 128 = 8\,192$ to only $(16 \times 4) + (16 \times 128) = 2\,112$, thus drastically decreasing the computation time.

ALGORITHM 2: Approximated Expansion Term Selection

Require: List of the top n term embedding clusters C_u associated to the user u. List of the query term embeddings Q.

```
function SELECT_EXPANSION_TERM_EMBEDDINGS_APPROX($C_u$, $Q$)
    exp_term_embs ← new List
    for all $c \in C_u$ do                    ▹ $c$ is one of the top $n$ embedding clusters of the user
        $centroid$ ← average($c$)
        centroid_sims ← new List
        for all $q \in Q$ do                  ▹ $q$ is the embedding of a query term
            $sim$ ← cos($centroid$, $q$)
            centroid_sims.push($sim$)
        $i$ ← argmax(centroid_sims)           ▹ index of the highest sim
        $q$ ← $Q[i]$                          ▹ query term embedding assigned to the cluster $c$
        approx_max_sims ← new List
        for all $t \in c$ do                  ▹ $t$ is a user's term embedding belonging to $c$
            $approx\_max\_sim$ ← cos($t$, $q$)
            approx_max_sims.push($approx\_max\_sim$)
        j ← argmax(approx_max_sims)           ▹ index of the highest max sim
        exp_term_embs.push($c[j]$)
    return exp_term_embs
```

3.3 Query Expansion with ColBERT

ColBERT is a neural retrieval model recently introduced by Khattab and Zaharia [47] that achieves state-of-the-art performances. Unlike other recent retrieval models based on Neural Networks and contextual word embeddings [54], ColBERT directly leverages query and document term embeddings to estimate the relevance scores of the documents in response to a query instead of, for example, comparing query and document embeddings obtained by a pooling operation over their term embeddings [35, 45, 56, 77, 99], such as taking their average. This characteristic makes ColBERT a good candidate model to study Query Expansion with contextual word embeddings, as we can add the expansion term embeddings to the query representation before computing the documents' relevance scores seamlessly. More formally, given a text t consisting of a sequence of tokens $[t_1, \ldots, t_n]$, ColBERT computes a matrix of size $n \times D$, where n is the number of tokens in the text and D is the dimension of each token representation. Under the hood, Colbert relies on BERT [28] to generate contextual vector representations of queries' and documents' terms. On top of BERT, a linear layer with no activation function controls the embeddings' dimension D, compressing the BERT representations to reduce memory consumption. In addition, Colbert leverages BERT's capabilities to augment queries shorter than a predefined length, generating additional vectors that contribute to the estimation of the documents' relevance scores. The final query representations have a fixed size of 32 embeddings. ColBERT computes the relevance score of a document d in response to a query q as the sum of the maximum cosine similarities among the document's and the query's term embeddings:

$$s_{\boldsymbol{q},\boldsymbol{d}} = \sum_{\boldsymbol{q}_i \in \boldsymbol{q}} \max_{\boldsymbol{d}_j \in \boldsymbol{d}} \cos(\boldsymbol{q}_i, \boldsymbol{d}_j), \tag{2}$$

where $\boldsymbol{q}$ and $\boldsymbol{d}$ are the sets of the query term embeddings and the document term embeddings, respectively, and $\boldsymbol{q}_i$ and $\boldsymbol{d}_j$ are the embeddings of specific query and document terms. In the actual implementation, Colbert normalizes the term representations to a unit L2 norm and evaluates the similarity between queries' and documents' term embeddings using the dot product, which is equivalent to the cosine similarity in this particular case.

Although the query augmentation mechanism leveraged by ColBERT is effective in enhancing its retrieval effectiveness, Wang et al. [96] have shown that an additional query expansion stage can improve it even further, paving the way for future studies on query expansion with contextual word embeddings.

In this article, we enhance ColBERT through Personalized Query Expansion with contextual word embeddings and show that our proposed approach significantly improves its retrieval effectiveness with minimal overhead. Our proposed method outperforms the approach of Wang et al. [96] and recent Personalized Query Expansion methods based on word embeddings [50, 108] in both retrieval effectiveness (Section 6.1) and efficiency (Section 6.2). As said before, ColBERT allows us to add the expansion term embeddings to the query representation seamlessly, guaranteeing the fairness of the comparison of different approaches of Query Expansion with contextual word embeddings. Moreover, having a fixed backbone model allows for the clear identification of the advantages and the deficiencies of the compared Query Expansion approaches, as there are no other differences in the retrieval pipeline.

For Query Expansion purposes, we extend Equation (2) to account for the expansion term embeddings by taking a convex combination of the scores of the original query term embeddings and those produced by the expansion term embeddings as follows:

$$s_{\boldsymbol{q},\boldsymbol{e},\boldsymbol{d}} = (1-\gamma) \cdot \sum_{\boldsymbol{q_i} \in \boldsymbol{q}} \max_{\boldsymbol{d_j} \in \boldsymbol{d}} \cos(\boldsymbol{q_i}, \boldsymbol{d_j}) + \gamma \cdot \sum_{\boldsymbol{e_k} \in \boldsymbol{e}} \max_{\boldsymbol{d_j} \in \boldsymbol{d}} \cos(\boldsymbol{e_k}, \boldsymbol{d_j}), \tag{3}$$

where $\boldsymbol{q}$, $\boldsymbol{e}$, and $\boldsymbol{d}$ are the sets of the query term embeddings, the personalized expansion term embeddings, and the document term embeddings, respectively; $\boldsymbol{q_i}$, $\boldsymbol{e_k}$, and $\boldsymbol{d_j}$ are the embeddings of specific query, expansion, and document terms; and γ is a parameter that controls the influence of the original and the expansion term embeddings on the final score.

4 PERSONALIZED QUERY EXPANSION FRAMEWORK

In this section, we describe the Personalized Query Expansion framework we employed for the comparative evaluation reported in the following sections. This framework allowed us to test different Personalized Query Expansion approaches isolating their contribution from the rest on the retrieval pipeline.

Figure 3 depicts the *Personalized Query Expansion framework* we relied on for comparing the Personalized Query Expansion methods presented in Section 5.2 and our newly proposed approach introduced in Section 3. The framework comprises one module that generates the vector representations of the terms of each document of the collection, those of the user-related documents' terms, and those of the query terms. Once the user-related term embeddings and the query term embeddings are computed, the *Expansion Module* selects the term embeddings for expansion among those of the user and adds them to the query. Finally, a *scoring function* computes a personalized relevance score for each document of the document collection by comparing the representations of its terms with those of the expanded query terms. In our experiments, we rely on ColBERT [47] to generate the term representations, one of the Personalized Query Expansion baselines described in Section 5.2, or our novel approach introduced in Section 3 as the *Expansion Module*, and we employed Equation (3) to compute the personalized relevance scores for the documents. As the main contribution we present in this article is the novel Personalized Query Expansion method introduced in Section 3, the framework we implement for the evaluation is functional to

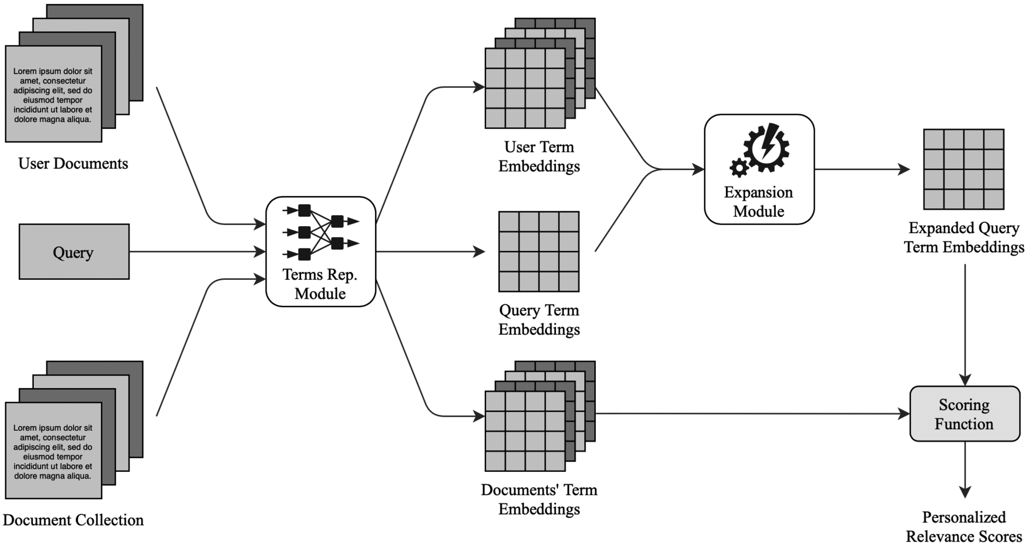

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
User Documents
Query
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Document Collection
Terms Rep. Module
User Term Embeddings
Query Term Embeddings
Documents' Term Embeddings
Expansion Module
Expanded Query Term Embeddings
Scoring Function
Personalized Relevance Scores

Fig. 3. Personalized query expansion framework.

comparatively evaluate the effectiveness of the proposed approach with previous methods at the state of the art with ease, allowing us to switch between the Query Expansion models seamlessly.

Although our Personalized Query Expansion framework can be directly applied to Personalized Search, for evaluation purposes, we use it to re-rank the results retrieved for the initial queries by BM25 [79]. This choice was conditioned by the employed benchmark, which—by construction—is meant to be used for re-ranking the BM25 results (i.e., all the relevant documents for a query are within the top results retrieved by BM25), as described in Section 5.1. As reported by Tabrizi et al. [87] and Bassani et al. [6], the lack of publicly available large-scale datasets of high quality is a known issue in Personalized Search Evaluation. Moreover, the approach used to derive evaluation datasets for Personalized Query Expansion from the data of social tagging platforms used in the past (see Section 2.2) has been recently criticized for the low quality of the obtained benchmarks [6, 87]. Finally, none of the datasets from previous works on Personalized Query Expansion is currently available. We refer the reader to [6] for a detailed description of the current state of the datasets for Personalized Search Evaluation. We also acknowledge that the re-ranking setting is often considered for evaluation purposes of novel retrieval models [47, 54, 70] based on contextual word embeddings and Transformer architectures [93], such as ColBERT, to leverage the efficiency of a fast first-stage retriever while retaining much of the effectiveness on these new models.

In Section 6, we report both retrieval effectiveness statistics when only the re-ranker scores are employed and those obtained when combining them with the BM25 scores. In the latter case, we aggregated the two relevance scores via the weighted sum fusion algorithm provided by `ranx.fuse` [7]. In this context, weighted-sum fusion works as a convex combination of the BM25 and re-ranker scores:

$$\mathit{final\_score} = (1 - \lambda) \cdot a + \lambda \cdot b, \tag{4}$$

where a and b are the relevance scores computed by BM25 and the re-ranker, respectively, and λ is a parameter that controls the influence of the two on the final score.

5 EXPERIMENTAL SETUP

The experiments reported in this section aim to answer the following nine research questions:

RQ1 Can a Personalized Query Expansion approach based on contextual word embeddings enhance ColBERT's retrieval effectiveness?
RQ2 Is our approach more effective than previously proposed expansion methods?
RQ3 Is our approach more robust than previously proposed expansion methods?
RQ4 Is our approach more efficient than previously proposed expansion methods?
RQ5 Does our approach improve the expansion terms' diversity compared to previous Personalized Query Expansion methods?
RQ6 Does clustering the user term embeddings following the clusters of the collection term embeddings allow us to achieve better retrieval effectiveness than directly identifying the user-related clusters from the term embeddings of each user?
RQ7 Is Equation (1) effective in identifying the user-term clusters that better represent the user's interests, thus enhancing the retrieval effectiveness of our proposed approach?
RQ8 Does our approximated expansion term selection perform on par with the original procedure proposed in Section 3.2 in terms of retrieval effectiveness?
RQ9 Does the approximation proposed to select the personalized expansion terms increase the efficiency w.r.t. the original procedure proposed in Section 3.2?

To answer the research questions **RQ1**, **RQ2**, **RQ3**, **RQ4**, and **RQ5**, we conduct a comparative evaluation of the retrieval effectiveness, robustness, and efficiency of different personalized and non-personalized Query Expansion methods and analyze the similarity among the terms they select for expansion. Similarly, to answer the research questions **RQ6**, **RQ7**, **RQ8**, and **RQ9**, we compare our proposed Personalized Query Expansion approach described in Section 3 with several variants.

In the following sections, we present the dataset we employ for conducting our evaluations (Section 5.1), introduce the baselines we have selected (Section 5.2), outline the training setup (Section 5.3) and the hyper-parameter optimization procedure (Section 5.4), and introduce the evaluation metrics (Section 5.5) used to assess the models' effectiveness. We make all our code available for future works and reproducibility purposes.[5]

5.1 Datasets

In this section, we introduce the datasets employed to conduct our experimental evaluation. Due to the lack of standardized test collections for Personalized Search, we rely on the Personalized Results Re-Ranking benchmark proposed recently by Bassani et al. [6]. This benchmark accounts for 18 million documents and 1.9 million queries divided into four datasets in the following domains: Computer Science, Physics, Psychology, and Political Science. The authors built their datasets by applying and refining the PERSON methodology [87], which consists in leveraging academic papers to derive user-query-document triplets. Specifically, the authors of PERSON proposed to consider, for each paper, the title as a query, the documents listed in its reference section as relevant documents, and one of its authors as the user submitting the query. To validate their approach, the authors performed several experiments. Their findings suggest that equivalent conclusions can be drawn from the comparison of Personalized Information Retrieval systems applied to a dataset built following PERSON and human judgments. As reported by Bassani et al. [6], some limitations that affect the methodologies for building synthetic datasets for Personalized Search evaluation also affect PERSON. Most notably, (1) the derived datasets cannot be employed

[5] We will add a link to the repository upon acceptance.

Table 1. Statistics of the Employed Benchmark Datasets

	Computer Science	Physics	Political Science	Psychology
# documents	4 809 684	4 926 753	4 814 084	4 215 384
# users	5 260 279	5 835 016	6 347 092	4 825 578
# train queries	552 798	728 171	162 597	544 882
# validation queries	5 583	7 355	1 642	5 503
# test queries	6 497	6 366	5 715	12 625
# relevant (avg ± sd)	3.25 ± 3.27	4.17 ± 4.15	3.88 ± 5.17	4.73 ± 4.4

for evaluating session-based personalization approaches due to the lack of search sessions, and (2) the synthetic user-query-document triplets are unique as they derive from non-repeatable user actions, such as writing a scientific manuscript. Therefore, approaches relying on re-finding behavior cannot be evaluated.

To compose their benchmark datasets, Bassani et al. [6] started by collecting paper titles, abstracts, references, and other metadata for several millions of papers across multiple disciplines from the Microsoft Academic Knowledge Graph [31, 85]. Once the document collections were composed and cleaned, the authors generated candidate queries following the approach we previously discussed. Then, to ensure the personalization potential for those queries, the authors discarded the queries whose users published fewer than 20 papers before the one used as the query. Since titles of academic papers are well-formed natural language, Bassani et al. [6] proposed to apply stop-word removal and Krovetz stemming to obtain queries closer to real-world ones. As discussed by Tabrizi et al. [87], the authors of PERSON, not all the documents listed in the reference section of a paper are necessarily relevant—from an Information Retrieval perspective—to the topic expressed by a query constructed from the paper's title. Therefore, to reduce the presence of spurious relevant documents and malformed queries, Bassani et al. [6] considered well-formed queries only those for which BM25 [79] places relevant documents in the top-ranking positions. Likewise, for each of the remaining queries, the authors retained only the relevant documents present in the top results retrieved by BM25. Finally, to closely resemble real-world scenarios—where all searches in the test set happen after those in the training set—the datasets were split chronologically into training and test sets. Training sets were then randomly split into training sets and validation sets, using a splitting ratio of 99 : 1. Table 1 reports some statistics about the datasets. As intended by their authors, we considered the four benchmark datasets separately for both training and evaluation. Specifically, we trained the compared models and fine-tuned their hyper-parameters independently on each dataset and considered the documents of each domain as separate collections.

Although the employed datasets rely on academic documents, we did not employ any domain-specific information in our proposed approach to ensure applicability to other domains. The documents used for conducted personalization—the papers published by each user/author—could be intended, for example, as previously accessed documents in a generic search context, such as Web Search.

5.2 Baselines

In this section, we introduce the baselines employed in our comparative evaluation. First, we compare our proposed Personalized Query Expansion-enhanced ColBERT to its original implementation, to assess whether our proposed approach is able to improve its retrieval effectiveness. Then we consider other query expansion approaches based on word embeddings, three of which take into account the user preferences, to verify if our proposed approach is improving over the state of the art. In all our experiments, we consider BM25 [79], our first-stage retriever, for reference.

- **ColBERT:** ColBERT [47] is the recent BERT-based retrieval model introduced in Section 3.3. We consider ColBERT as a baseline to assess whether the compared query expansion methods are able to enhance its retrieval capabilities.
- **ColBERT-PRF:** ColBERT-PRF [96] is a recently introduced query expansion method based on ColBERT relying on pseudo-relevance feedback [80] implemented in PyTerrier [61]. Specifically, given a query, it first ranks the documents using ColBERT, then clusters the term embeddings of a certain number of feedback documents with k-Means [62] and selects the tokens corresponding to the cluster centroids with higher Inverse Document Frequency scores for expanding the original query. We consider ColBERT-PRF as a baseline to assess whether personalization is meaningful for query expansion in our context.
- **Baseline 1:** It is a Personalized Query Expansion method introduced by Kuzi et al. [50] that selects expansion terms based on the cosine similarity between their embeddings and the query term embeddings. Specifically, it first computes the cosine similarity between each user-related term embedding and each query term embedding. Then, it softmax-normalizes those similarities to get a probability distribution of the importance of user-related term embeddings w.r.t. each query term embedding. Finally, it sums the log probabilities of each user-related term embedding and selects the top-scored ones for expanding the original query.
- **Baseline 2:** It is a Personalized Query Expansion method introduced by Zhou et al. [108] that selects expansion terms based on the cosine similarity between their embeddings and the sum of the query term embeddings. That is, it simply computes the cosine similarities among the user-related term embeddings and the sum of the query term embeddings and selects the top-scored ones for expanding the original query.
- **Baseline 3:** It is a variant of Baseline 2 we introduce by using the CLS token embedding in place of the sum of the query term embeddings. The CLS token is a special token appended by BERT [28] at the beginning of each text before computing its contextual word embeddings. It was originally introduced for sentence-level classification tasks, but its embedding was also used in Information Retrieval as a single embedding representation of queries and documents [45, 56, 77, 99]. At a theoretical level, the CLS token embedding is a sort of weighted sum of the other token embeddings and represents the semantic meaning of the input text as a whole.

We apply all the expansion methods before re-ranking the BM25 results with ColBERT. We do not consider the Personalized Query Expansion approach based on word embeddings proposed by Wu et al. [98] as it requires data unavailable in our setting. We also do not consider other Personalized Query Expansion approaches based on word embeddings, such as that proposed by Amer et al. [2], as they are almost identical to the considered baselines or their authors did not report encouraging results. Unfortunately, due to the hardware limitations described in Section 5.3, we are unable to compare the Query Expansion-enhanced variants of ColBERT with recent Transformer-based retrieval models without incurring an unfair comparison. For the sake of completeness, we show in Appendix A that vanilla ColBERT outperforms other retrieval models at the state of the art on the benchmark datasets employed in our experimental evaluation.

5.3 Implementation Details

We relied on PyTorch [72], HuggingFace's Transformers [97], and PyTorch Lightning [30] for implementing and training ColBERT. We employed the cuML's GPU-based implementation of HDBSCAN [76] for clustering purposes. Finally, we implemented and optimized all the considered expansion methods with Numpy [37] to allow for a fair CPU-based efficiency comparison. To further ensure the reproducibility of the experiments, we relied on Hydra [102] to store the experiments' configurations.

We trained ColBERT on an NVidia® RTX A6000 GPU for 20 epochs following the instruction reported in its original paper [47]: learning rate set to 3×10^{-6}, batch size set to 32, number of embeddings per query set to 32, Adam optimizer [48], and pairwise softmax cross-entropy loss over a triplet composed of a query, a relevant document, and a non-relevant document. During training, we sampled hard negatives from the top results retrieved by BM25 and used the other documents in the batch as random negative samples. For tuning the hyper-parameters of the compared Query Expansion approaches as described in Section 5.4, we need to pre-compute the representations of the terms of the documents in the collection and load them into the computer's main memory so as to make the hyper-parameter search feasible. However, the hardware employed to run the experiments has 64 GB of RAM only, which is not enough to store the embeddings of all the terms appearing in the document collection. Therefore, we set the maximum number of embeddings per document to 128 and truncated the longer ones to reduce the memory footprint. We also set the dimension of the embeddings produced by ColBERT to 16 to reduce the memory footprint further. As all the compared models but BM25 share the embeddings generated by ColBERT, they are all affected equally, thus preserving the fairness of our comparison. To reduce the time needed to find the term embedding clusters with HDBSCAN, we heavily down-sampled the embeddings of each collection from ~500 to 10 million. For ColBERT and all its Query Expansion-enhanced variants, we aggregated the newly computed document relevance scores with the BM25 scores shared with the employed datasets [6], using the weighted sum fusion algorithm provided by `ranx.fuse` [7] after optimization on the validation set. Finally, we removed the term embeddings of stop-words from the possible embeddings to choose for query expansion.

5.4 Hyper-parameter Tuning

All the baseline expansion methods considered in our comparison and the proposed one have hyper-parameters controlling their behavior, which we optimize on the validation set. Specifically, they all have a parameter controlling the number of expansion terms to add to the queries, which in the case of PQEWC also corresponds to the number of user-related term embedding clusters to consider as the most representative of the user interests. ColBERT-PRF, Baseline 1, and PQEWC also have a parameter controlling the importance of the expansion terms when computing the document relevance scores, i.e., the expansion terms' weight (in the case of PQEWC, γ from Equation (3)). Finally, ColBERT-PRF has a parameter controlling the number of feedback documents to consider as pseudo-relevance feedback and a parameter controlling the number of clusters for grouping the feedback documents' term embeddings. We also report here the best values for the λ parameter of Equation (4) found on the validation set of each dataset for each model. We consider the following intervals and sets to generate the hyper-parameter configurations during optimization:

- Number of expansion terms in the interval $[1, 32]$
- Expansion term weight in the interval $[0.1, 0.9]$ with a step of 0.1
- Number of feedback documents in the interval $[1, 10]$
- Number of clusters in the set $[8, 16, 24, 32, 40, 48, 56, 64]$
- λ in the interval $[0.1, 0.9]$ with a step of 0.1

We optimized Baseline 2 and Baseline 3 with a greed search on the validation set as they have only one hyper-parameter, the number of expansion terms. We fine-tuned the hyper-parameters of ColBERT-PRF, Baseline 1, and PQEWC with the Python optimization package Optuna [1], testing 100 hyper-parameter configurations for each of them. After the models' parameter optimization, we optimized the λ parameter of Equation (4) using the greed search already implemented in

Table 2. Best Hyper-parameter Configurations

Model	# Feedback Docs				# Clusters				# Expansion Terms				Exp. Terms Weight				λ			
	CS	PHY	PS	PSY	CS	PHY	PS	PSY	CS	PHY	PS	PSY	CS	PHY	PS	PSY	CS	PHY	PS	PSY
ColBERT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0.8	0.8	0.8	0.9
ColBERT-PRF	6	1	1	1	16	24	24	16	8	22	3	13	0.1	0.1	0.1	0.1	0.8	0.8	0.8	0.8
Baseline 1	—	—	—	—	—	—	—	—	25	29	18	3	0.2	0.2	0.2	0.1	0.9	0.9	0.8	0.9
Baseline 2	—	—	—	—	—	—	—	—	16	14	32	22	—	—	—	—	0.9	0.9	0.9	0.9
Baseline 3	—	—	—	—	—	—	—	—	8	4	10	3	—	—	—	—	0.8	0.9	0.8	0.9
PQEWC	—	—	—	—	—	—	—	—	32	32	31	32	0.3	0.3	0.4	0.3	0.9	0.9	0.9	0.9

CS, PHY, PS, PSY stand for Computer Science, Physics, Political Science, and Psychology, respectively.

ranx, the Python library we employed for score fusion. Table 2 reports the best hyper-parameter configuration for each method and dataset.

5.5 Evaluation Metrics

To evaluate the effectiveness of the considered models, we re-ranked the top 1,000 results retrieved by BM25 and we employed (1) ***Mean Average Precision*** **(MAP)**, (2) ***Mean Reciprocal Rank*** **(MRR)**, (3) ***Normalized Discounted Cumulative Gain*** **(NDCG)**, and (4) ***Rank-biased Precision*** **(RBP)**. MRR and NDCG were computed on the top 10 documents retrieved by each model, whereas MAP was computed on the top 100. RBP's persistence was set to 0.95. Statistical significance testing was conducted using a Bonferroni corrected Two-sided Paired Student's t-Test [86] with $p < 0.005$. To evaluate the robustness of the query expansion methods, we employ the **Robustness Index (RI)** [26]. RI is defined as $\frac{N^{+}-N^{-}}{|Q|}$, where N^{+} and N^{-} are the amounts of queries whose result lists are improved or worsened by an expansion method in terms of Average Precision (at 100) w.r.t. ColBERT, and $|Q|$ is the total number of queries. The higher the RI, the more robust an expansion method is. Computation and comparison of metrics were conducted using the Python evaluation library ranx [5].

6 RESULTS AND DISCUSSION

In this section, we present the results of our comparative evaluation. First, we discuss the retrieval effectiveness, efficiency, and diversity of the terms chosen for expansion by the compared models in Sections 6.1 to 6.3, respectively. Then, we ablate the design choices of our proposal in Section 6.4. Finally, we summarize our findings in Section 6.6.

6.1 Effectiveness

In this section, we discuss the performances of each of the compared models as well as the results of their fusion with the first-stage retriever, BM25, aiming to answer our research questions **RQ1**, **RQ2**, and **RQ3**.

First, we compare the results of the re-ranking models without the document score interpolation of Equation (4). As shown in Table 3, all the ColBERT-based re-rankers were able to consistently outperform the first-stage retriever, BM25, by a considerable margin. However, there are some clear differences in the benefits brought by the Query Expansion methods to ColBERT. ColBERT-PRF, our non-personalized Query Expansion baseline, achieved statistically significant improvements over vanilla ColBERT for all the considered datasets in MAP, NDCG, and RBP, but not MRR. In two cases, Physics and Psychology, ColBERT-PRF even decreased in MRR w.r.t. ColBERT. The Personalized Query Expansion baselines (Baseline 1, Baseline 2, and Baseline 3) caused a degradation of the effectiveness of vanilla ColBERT in the large majority of cases. The only exception is Baseline 3 on the Physics dataset, which achieved statistical improvements over vanilla ColBERT in MAP, NDCG, and RBP. The worst case is Political Science, where all the Personalized Query Expansion

Table 3. Effectiveness of the Compared Models

Model	Computer Science					Physics				
	MAP@100	MRR@10	NDCG@10	RBP.95	RI	MAP@100	MRR@10	NDCG@10	RBP.95	RI
BM25	12.25	48.92	22.45	13.22	—	12.77	53.68	26.88	16.05	—
ColBERT	18.10	56.56	28.24	17.62	—	17.91	61.86	32.92	20.20	—
ColBERT-PRF	$\underline{18.56}^{\dagger}$	56.82	$\underline{28.68}^{\dagger}$	$\underline{17.90}^{\dagger}$	**$\underline{20}$**	$\underline{18.77}^{\dagger}$	61.50	$\underline{33.76}^{\dagger}$	$\underline{20.75}^{\dagger}$	$\underline{17}$
Baseline 1	18.13	56.32	28.15	17.63	−1	17.83	61.18	32.58	20.09	−5
Baseline 2	17.92	56.23	28.11	17.47	0	17.93	61.83	32.97	20.26	4
Baseline 3	18.18	$\underline{56.86}$	28.43	17.66	6	18.05†	$\underline{62.56}$	33.14†	20.30†	10
PQEWC	**19.03**‡	**57.66**†	**29.23**‡	**18.23**‡	15	**19.17**‡	**63.81**‡	**34.46**‡	**21.12**‡	**22**
Model	Political Science					Psychology				
	MAP@100	MRR@10	NDCG@10	RBP.95	RI	MAP@100	MRR@10	NDCG@10	RBP.95	RI
BM25	13.27	50.23	24.07	14.24	—	12.58	51.19	23.93	13.84	—
ColBERT	16.06	$\underline{53.51}$	26.40	16.11	—	21.39	62.78	33.39	20.17	—
ColBERT-PRF	$\underline{16.42}^{\dagger}$	$\underline{53.51}$	$\underline{26.86}^{\dagger}$	$\underline{16.33}^{\dagger}$	$\underline{11}$	$\underline{21.92}^{\dagger}$	62.53	$\underline{33.83}^{\dagger}$	$\underline{20.43}^{\dagger}$	$\underline{10}$
Baseline 1	15.98	53.05	26.22	16.08	−3	21.42	62.85	33.41	20.18	4
Baseline 2	15.92	52.80	26.11	16.04	−1	21.22	$\underline{63.26}$	33.40	19.99	−2
Baseline 3	15.49	52.85	25.84	15.74	−5	21.37	62.76	33.37	20.15	4
PQEWC	**17.24**‡	**55.10**‡	**27.71**‡	**16.99**‡	**14**	**22.30**‡	**64.21**‡	**34.47**‡	**20.75**‡	**12**

The symbols † and ‡ denote significant improvements in a Bonferroni corrected Two-sided Paired Student's t-Test with $p < 0.005$ over ColBERT model only and over all models, respectively. Best results are highlighted in boldface. Best baselines' results are underlined.

baselines decreased vanilla ColBERT performance w.r.t. all the considered evaluation metrics. Those results clearly show the unsuitability of previous Personalized Query Expansion Methods based on word embeddings when applied in the presence of contextual word embeddings.

Our proposed Personalized Query Expansion method, PQEWC, achieved the best results on all the considered datasets and significantly improved over ColBERT and all the considered baselines in all the considered search scenarios. On average, it improved over ColBERT by 6%, 2%, 4%, and 4% in MAP, MRR, NDCG, and RBP, respectively, and over the best-performing baselines by 4%, 2%, 2%, and 2% in MAP, MRR, NDCG, and RBP, respectively. Furthermore, it scored a higher Robustness Index than all the other Personalized Query Expansion methods for all the considered datasets and higher than ColBERT-PRF on three datasets out of four. We also notice that our proposed approach is the only Query Expansion approach to achieve statistically significant increments over ColBERT w.r.t. MRR. These results clearly show the benefits of our proposed procedure for Personalized Query Expansion with contextual word embeddings. Moreover, they corroborate our intuition regarding the need for a pre-processing phase to identify the most relevant user interests and reduce the impact of redundant expansion terms.

When the document scores produced by ColBERT and its Query Expansion-enhanced variants are aggregated with the scores produced by the first-stage retriever (BM25) following Equation (4), we record much less difference between vanilla ColBERT and its variants, with the sole exception of the one employing our proposed Personalized Query Expansion method PQEWC. As shown in Table 4, there is generally little to no difference between vanilla ColBERT and the considered baselines regarding retrieval effectiveness. Conversely, PQEWC-enhanced ColBERT achieved the best performances for all the considered metrics and datasets. These results highlight that our proposed method captures personalized relevance signals that are complementary to those of both vanilla ColBERT and BM25. On average, it improved over BM25 + ColBERT by 4%, 2%, 3%, and 3% in MAP, MRR, NDCG, and RBP, respectively.

With and without interpolation, PQEWC outperformed ColBERT and all the other considered baselines and generally reached a higher Robustness Index. These results positively answer our first, second, and third research questions, **RQ1**, **RQ2**, and **RQ3**.

Table 4. Effectiveness of the Compared Models When the Document Scores They Compute Are Interpolated with Those Computed by the First-stage Retriever BM25 Using Equation (4)

Model	Computer Science					Physics				
	MAP@100	MRR@10	NDCG@10	RBP.95	RI	MAP@100	MRR@10	NDCG@10	RBP.95	RI
BM25	12.25	48.92	22.45	13.22	—	12.77	53.68	26.88	16.05	—
BM25 + ColBERT	20.08	60.72	30.99	18.83	—	19.93	65.26	35.74	21.74	—
BM25 + ColBERT-PRF	20.21†	60.56	30.95	18.87†	6	20.12†	64.96	35.77	21.80†	1
BM25 + Baseline 1	19.97	60.47	30.74	18.78	−1	19.88	65.17	35.76	21.70	−1
BM25 + Baseline 2	19.87	60.55	30.86	18.70	1	20.06†	65.56	36.00†	21.88†	8
BM25 + Baseline 3	20.21†	60.72	31.04	18.88	4	19.98	65.73	35.91	21.80	4
BM25 + PQEWC	**20.73**‡	**61.45**†	**31.53**‡	**19.26**‡	**11**	**20.92**‡	**66.28**†	**36.85**‡	**22.45**‡	**21**

Model	Political Science					Psychology				
	MAP@100	MRR@10	NDCG@10	RBP.95	RI	MAP@100	MRR@10	NDCG@10	RBP.95	RI
BM25	13.27	50.23	24.07	14.24	—	12.58	51.19	23.93	13.84	—
BM25 + ColBERT	19.03	59.55	30.39	18.15	—	23.06	66.02	35.73	21.22	—
BM25 + ColBERT-PRF	19.13†	59.21	30.43	18.23†	4	23.02	65.55	35.48	21.12	−4
BM25 + Baseline 1	19.07	59.51	30.43	18.21†	−1	23.09†	66.03	35.74	21.23†	−4
BM25 + Baseline 2	18.95	59.32	30.37	18.10	0	23.11	66.53	35.98†	21.22	0
BM25 + Baseline 3	18.82	58.97	30.22	18.07	0	23.07	66.02	35.74	21.22	5
BM25 + PQEWC	**19.81**‡	**60.57**‡	**31.24**‡	**18.72**‡	**11**	**23.96**‡	**67.14**‡	**36.71**‡	**21.76**‡	**14**

The symbols † and ‡ denote significant improvements in a Bonferroni corrected Two-sided Paired Student's t-Test with $p < 0.005$ over BM25 + ColBERT and over all models, respectively. Best results are highlighted in boldface. Best baselines' results are underlined.

6.2 Efficiency

In this section, we compare the efficiency of the considered expansion methods in terms of the average time required to expand a query on the CPU (an AMD Ryzen™ 5950X, in our case), aiming to answer our research question **RQ4**. We suppose to have already loaded all the data needed for query expansion into memory. This assumption is supported by the fact that embeddings of user-related terms should be already available in the computer's main memory as they are part of the searchable documents, which must reside in the main memory to be processed by ColBERT, regardless of personalization. It is important to note that our evaluation scenario is akin to a generic personalized search setup, where personalization is achieved by leveraging the documents previously accessed by the user, which would reside in the main memory to be accessed by ColBERT. An example of such a search scenario is Personalized Web Search [88]. This way, we can focus on the expansion term selection latency. Note that all the compared models took less than 1 millisecond to re-rank the top 1,000 BM25 results with ColBERT on our GPU. Therefore, we do not report the re-ranking times.

The second to fifth rows of Table 5 report the expansion time required by the considered Query Expansion methods on the datasets employed in our evaluation. ColBERT-PRF was the least efficient of them, requiring 32 ms to expand a query on average. Baseline 1 required 4 ms to expand a query on average, almost 10 times less than ColBERT-PRF, but it was not able to improve over ColBERT. Baseline 2 and Baseline 3, which delivered similar results in terms of effectiveness, both took less than 1 millisecond to expand a query on average. Finally, our proposed Personalized Query Expansion approach, PQEWC, achieved an expansion time inferior to 1 ms while delivering the best retrieval performances across the line.

Although all the considered Personalized Query Expansion methods are very efficient in our context, the number of operations needed by PQEWC is much lower than the other methods. PQEWC only compares the query term embeddings with small subsets of the user term embeddings, as discussed in Section 3.1, allowing our proposed method to be much more scalable than the others. On average, it compares the query term embeddings with less than 15% of the term embeddings

Table 5. Query Expansion Methods' Execution Time in Milliseconds

Dataset	Emb Size	Embs / User	File Size (MB)	Load Time (ms)	ColBERT-PRF	Baseline 1	Baseline 2	Baseline 3	PQEWC
Computer Science	16	12 000	< 1	—	39	5	< 1	< 1	< 1
Physics	16	13 000	< 1	—	34	5	< 1	< 1	< 1
Political Science	16	6 500	< 1	—	28	2	< 1	< 1	< 1
Psychology	16	11 000	< 1	—	27	4	< 1	< 1	< 1
Synthetic	16	10 000	< 1	< 1	—	5	< 1	< 1	< 1
Synthetic	16	100 000	3	1	—	61	7	7	< 1
Synthetic	16	1 000 000	32	10	—	651	88	88	1
Synthetic	128	10 000	3	< 1	—	7	1	1	< 1
Synthetic	128	100 000	26	8	—	72	12	12	< 1
Synthetic	128	1 000 000	256	80	—	740	141	140	9
Synthetic	768	10 000	15	5	—	8	2	2	< 1
Synthetic	768	100 000	154	50	—	147	23	23	3
Synthetic	768	1 000 000	1 536	500	—	1494	249	249	26

Emb Size is the embedding dimension. Embs / User is the average number of term embeddings related to a specific user. File Size is the average file size in megabyte of the term embeddings for each user. Load Time is the time needed on average to load from disk the term embeddings related to a specific user in a cold start scenario. In the case of the datasets employed for our evaluation, the values are affected by document truncation as described in Section 5.3.

associated with a user. In contrast, all the other methods consider all of them. Moreover, as introduced in Section 3.2, it only compares a single query term embedding with the most representative ones of each user.

To further prove our claims on PQWEC scalability, we conducted an empirical evaluation based on synthetically generated data. Since ColBERT-PRF latency was already high on our test sets, we did not consider it in this additional experiment. For each personalized expansion method, we investigated several different scenarios. Specifically, we considered three different embedding sizes and three different amounts of average user-associated term embeddings. As for embedding sizes, we considered 16, 128, and 768, which are the dimension of the embedding we used in the experiment previously reported, the embedding dimension originally proposed for ColBERT, and the dimension of the uncompressed BERT embeddings, respectively. As for the average number of user term embeddings, we considered 10,000, 100,000, and 1,000,000.

As reported in Table 5, Baseline 1 rapidly saturates as more user-related terms becomes available, making it not suitable for real-world scenarios with high availability of user-related texts. On average, Baselines 2 and 3 require the same time to expand a query. Their applicability is mainly affected by the number of available user term embeddings rather than their dimension. As expected, PQEWC is the most scalable of the compared query expansion methods, and it is suited even for data-intensive scenarios. As shown in the table, it took just a fraction of the time required by the other models to expand the queries in each considered scenario. The embedding dimension has a noticeable impact on all the Personalized Query Expansions' execution times. However, the efficiency advantages of PQEWC make it the sole model able to scale to both high user data availability scenarios and high-dimensional vector spaces. To conclude, these results corroborate our claims regarding the scalability of PQEWC and positively answer our fourth research question, **RQ4**.

As previously described, in our evaluation setting, all the user-related term embeddings are already available in the computer's main memory as they belong to the searchable documents, which must be in the main memory to be evaluated by ColBERT. Similarly, in Personalized Web Search based on previously accessed Web pages [88], the user-related data would be available in the computer's main memory for similar reasons. However, in other scenarios, offloading the memory from the user-related data could be convenient to reduce memory consumption. Therefore, in those cases, the system must load the user-related data when needed. We performed a further experiment to evaluate how Personalized Query Expansion methods are affected in those scenarios. Specifically, we analyzed the loading time of the user-related data to assess whether and how it affects the Query Expansion time for the Personalized approaches in the synthetic scenarios previously introduced. To do so, we compared the average size of the user-related term embeddings when saved on disk using Numpy [37] and our disk read speed in the cold start scenario to factor out cached data (~ 3 000 MB/s). As reported in Table 5, the loading time in the synthetic scenario most similar to the considered benchmark datasets (embedding size 16, 10 000 embeddings per user) was inferior to 1 ms. Therefore, in our search setting, the user would not notice any delay even if the system has to load his or her related data from the disk. Surprisingly, we also recorded a sub-millisecond loading time in the case of embeddings of size 128, the original embedding size of ColBERT. Therefore, even factoring out our hardware limitations described in Section 5.3 and the specificity of our evaluation scenarios, the user-related data loading time would not affect the user experience in the case of Personalized Query Expansion methods. For the sake of brevity, we now discuss only the most data-intensive simulated scenario (1M embeddings per user with a dimension of 768), as similar conclusions can be drawn for the others. In this specific scenario, our system took 500 milliseconds on average to load user-related data, introducing a noticeable latency. However, in a real-world scenario, the system could load the user-related data when a specific user connects to it, after login, or while the user types the first query of his or her search session, which would probably take at least a few hundred milliseconds. The system could offload specific user-related data at the end of the user's search session without re-loading them for each query separately, overcoming the data loading time entirely for each search after the initial one. Therefore, we conclude that, if carefully handled, loading user-related data when needed should not introduce any latency perceivable by the user. We highlight that, in this analysis, we did not consider that our proposed approach only compares the query with the term embeddings belonging to the clusters that better represent the user interests as identified by our proposed Equation (1). As previously discussed in this section, we found out that our approach compares the query with less than 15% of the term embeddings associated with a user on average. Therefore, the reported loading times reflect the worst-case scenarios, i.e., when the system must load all the user-related embeddings, which is the usual case for the other considered Personalized Query Expansion approaches.

6.3 Expansion Term Diversity

In this section, we compare the diversity of the user term embeddings selected for expansion by the considered Personalized Query Expansion methods aiming at answering our fifth research question, **RQ5**. Moreover, this analysis allows us to verify our intuitions regarding the potential issues of employing previous Personalized Query Expansion methods based on word embeddings with contextual word embeddings, as discussed in Section 1. In this regard, we introduce a novel metric to evaluate the percentage of semantically non-overlapping expansion terms per query. For each query, we first count the number of expansion term embeddings having a maximum cosine similarity score w.r.t. the other expansion term embeddings below a certain *semantic overlap threshold*. Then, we divide this counter by the number of terms selected for expansion and take the average across all queries. Our **Expansion Terms Diversity (ETD)** metric is as follows:

$$ETD_{\tau} = \frac{1}{n} \sum_{q=1}^{n} \frac{1}{k} \sum_{i=1}^{k} 1 \text{ if } \max_{\boldsymbol{e}_{j \neq i} \in \boldsymbol{E_q}} \cos(\boldsymbol{e_i}, \boldsymbol{e_j}) < \tau \text{ and }, 0 \text{ otherwise}, \tag{5}$$

where n is the number of queries, k is the number of expansion terms for the query q, $\boldsymbol{E_q}$ is the set of expansion term embeddings selected for the query q, $\boldsymbol{e_i}$ is the ith expansion term embedding, and τ is the semantic overlap threshold. We further propose to evaluate ETD with the following values for the semantic overlap threshold parameter τ: 0.99, 0.95, and 0.90. The rationale behind those values is as follows: low ETD.95/ETD.99 scores mean the query expansion method selects term

Table 6. Expansion Term Diversity of the Compared Personalized Query Expansion Methods

Model	Computer Science			Physics			Political Science			Psychology		
	ETD.99	ETD.95	ETD.90	ETD.99	ETD.95	ETD.90	ETD.99	ETD.95	ETD.90	ETD.99	ETD.95	ETD.90
Baseline 1	88.63	62.03	**46.30**	89.68	63.15	50.11	90.65	61.58	**50.64**	94.76	79.88	73.89
Baseline 2	85.05	41.35	20.37	83.43	33.82	11.14	91.40	42.44	13.16	88.86	34.13	11.94
Baseline 3	89.52	40.44	17.99	89.02	27.91	7.71	93.04	39.02	15.12	86.14	33.90	14.38
PQEWC	**99.56**	**88.38**	64.39	**99.41**	**81.24**	**50.06**	**99.68**	**84.72**	55.35	**99.38**	**82.41**	**53.93**

Higher is better for ETD.99 and ETD.95. Values near 0.5 for ETD.90 are better. Reported results are in percentages. Best results are highlighted in boldface.

embeddings with high/extremely high semantic overlap, i.e., almost duplicate term embeddings, while we interpret EDT.90 scores near 0.5 as an indication that the expansion terms are topically focused but not semantically overlapping, and thus they are diverse but semantically related. Note that a high EDT.90 score means the expansion terms are loosely correlated. We highlight that the expansion term diversity score is not directly correlated or proportional to the effectiveness gain brought by a Query Expansion method. However, it can help us understand why a method performs better or worse than another relative to a specific application domain.

Table 6 shows the ETD scores for the compared Personalized Query Expansion methods. The low ETD.99 and ETD.95 scores that Baseline 1, Baseline 2, and Baseline 3 achieved in all datasets tell us those methods are prone to select expansion term embeddings that suffer from semantic overlap. The semantic overlap among the expansion terms can cause the exacerbation of a single aspect of the queries, thus reducing the diversity of the search results, a property not desirable in our search evaluation domains given the unsatisfactory results achieved by those methods, as discussed in Section 6.1. Those results corroborate our intuitions regarding the potential issues of employing previous Personalized Query Expansion methods based on word embeddings with contextual word embeddings discussed in Section 1. By employing a clustering-based procedure to group and find the term embeddings that better represent the user interests and preferences (Section 3.1) and selecting only one embedding per user-related term embedding cluster for query expansion purposes (Section 3.2), PQEWC achieved very high ETD.99 and ETD.95 scores. Therefore, term embeddings selected for query expansion by PQEWC benefit from great diversity. Moreover, the EDT.90 scores tell us that PQEWC generally selects topically focused but not semantically overlapping expansion term embeddings. These results show the ability of PQEWC to enhance multiple aspects of the queries without affecting its topic coherence and positively answer our fifth research question, **RQ5**.

6.4 Ablation Study

In this section, we conduct the ablation study of our proposal to assess whether our design choices are effective.

Effectiveness. To evaluate whether our design choices are functional effectiveness-wise and the approximation method we proposed in Section 3.2 does not harm the retrieval effectiveness of our proposal, we compared it with the three following variants:

- **Local:** This variant derives the clusters of user term embeddings by directly applying HDBSCAN to the term embeddings of each user instead of mapping the user term embeddings into the clusters derived from the document collection, as described in Section 3.1. As there are no direct relations between local and global clusters (i.e., Equation (1) is not applicable), we consider the local clusters with the highest number of associated user term embeddings as the most representative of the user interests and preferences. This variant allows us to assess whether clustering the user terms following the collection topic distribution in the

Table 7. Effectiveness of Our Proposal Variants and Those of ColBERT

Model	Computer Science					Physics				
	MAP@100	MRR@10	NDCG@10	RBP.95	RI	MAP@100	MRR@10	NDCG@10	RBP.95	RI
ColBERT	18.10	56.56	28.24	17.62	—	17.91	61.86	32.92	20.20	—
Local	18.52*	57.27	28.68*	17.88*	7	18.48*	62.83*	33.57*	20.60*	11
Top Clusters	18.80*†	58.47*†	29.33*†	17.98*	11	18.59*	63.36*	33.97*†	20.66*	13
Non-approximated	19.05*†‡	57.58*	29.25*†	18.24*†‡	15	19.17*†‡	63.88*†	34.44*†‡	21.12*†‡	23
PQEWC	19.03*†‡	57.66*	29.23*†	18.23*†‡	15	19.17*†‡	63.81*†	34.46*†‡	21.12*†‡	22

Model	Political Science					Psychology				
	MAP@100	MRR@10	NDCG@10	RBP.95	RI	MAP@100	MRR@10	NDCG@10	RBP.95	RI
ColBERT	16.06	53.51	26.40	16.11	—	21.39	62.78	33.39	20.17	—
Local	16.14	53.24	26.38	16.28*	1	21.56*	62.94	33.56	20.27*	3
Top Clusters	16.78*†	54.92*†	27.41*†	16.56*†	9	21.89*†	64.02*†	34.17*†	20.43*†	12
Non-approximated	16.87*†	53.91	27.18*†	16.73*†‡	7	22.33*†‡	64.24*†	34.51*†‡	20.76*†‡	12
PQEWC	17.24*†‡	55.10*†	27.71*†	16.99*†‡	14	22.30*†‡	64.21*†	34.47*†‡	20.75*†‡	12

The symbols ★, †, and ‡ denote significant improvements in a Bonferroni corrected Two-sided Paired Student's t-Test with $p < 0.005$ over ColBERT, Local, and Top Clusters, respectively.

embedding space improves the retrieval performances over partitioning the embedding space user-wise (**RQ6**).

- **Top Clusters:** Instead of relying on the user-term clusters that better represent the user interests as identified by our proposed Equation (1), this variant focuses on the user-term clusters that are most related to the query, i.e., those most similar to the query in terms of cosine similarity. First, it selects the top n user-term clusters most similar to the query. Then, following the procedure proposed in Section 3.2, it chooses from each top user-term cluster a term embedding to use for query expansion. This variant allows us to assess whether considering the user-term clusters identified with Equation (1) as the best clusters to draw the expansion term embeddings from is an effective design choice (**RQ7**).
- **Non-approximated:** This variant does not employ the approximated expansion term selection strategy described in Section 3.2. It allows us to assess the impact on the retrieval effectiveness of the approximation we proposed to reduce the expansion term selection time (**RQ8**).

As for the main evaluation, all the expansion methods are applied before re-ranking the BM25 results with ColBERT, and their hyper-parameters have been optimized on the validation set.

Table 7 reports the retrieval effectiveness of PQEWC, those of its considered variants, and—for reference—those of ColBERT. The results show that building the user-related clusters following the clusters of the collection term embeddings as proposed in Section 3.1 and considering those top-ranked by Equation (1) as the best from which to draw the expansion term embeddings is more effective than the considered alternatives for all the considered datasets and evaluation metrics. Furthermore, PQEWC and its Non-approximated variant reached comparable effectiveness and robustness on all datasets but Political Science, where PQEWC even increased over Non-approximated. In general, our intuition regarding the computation of many unnecessary comparisons between the query term embeddings and the user-related term embeddings, discussed in Section 3.2, proved to be true. These results positively answer our research questions **RQ6**, **RQ7**, and **RQ8**.

Efficiency. In this section, we compare the efficiency of PQEWC with that of its Non-approximated variant. This comparison aims to evaluate in which contexts the approximation mechanism proposed in Section 3.2 is required and in which it is not. As shown in Table 8, for all the considered datasets PQEWC and its Non-approximated variant achieved sub-millisecond

Table 8. Execution Time of PQEWC Variants in Milliseconds

Dataset	Emb Size	Embs / User	Non-approximated	PQEWC
Computer Science	16	12 000	< 1	< 1
Physics	16	13 000	< 1	< 1
Political Science	16	6 500	< 1	< 1
Psychology	16	11 000	< 1	< 1
Synthetic	16	10 000	< 1	< 1
Synthetic	16	100 000	2	< 1
Synthetic	16	1 000 000	13	1
Synthetic	128	10 000	< 1	< 1
Synthetic	128	100 000	2	< 1
Synthetic	128	1 000 000	20	9
Synthetic	768	10 000	1	< 1
Synthetic	768	100 000	5	3
Synthetic	768	1 000 000	49	26

Emb Size is the embedding dimension. Embs / User is the average number of term embeddings related to a specific user. In the case of the datasets employed for our evaluation, the values are affected by document truncation as described in Section 5.3.

execution time. This result means the prominent factor in achieving top efficiency is limiting the search for expansion term embeddings to only the most representative user-term embedding clusters. However, the reader should consider that the similar expansion times of PQEWC and Non-approximated are also due to the efficient vector operations offered by the Intel® Math Kernel Library [94], which we use as the back-end for Numpy [37]. In fact, the number of term comparisons performed without approximation is 32 times larger than when employing our approximation mechanism, as ColBERT's query representations are always composed of 32 embeddings. Table 8 also reports the average expansion time needed by the two approaches in the same simulated scenarios described in Section 6.2. As shown in the table, the Non-approximated variant suffers in very data-rich scenarios. Nonetheless, it achieves far better efficiency than all the considered Personalized Query Expansion baselines, whose execution times are reported in Table 5. These results positively answer our ninth research question, **RQ9**. We conclude that the most important factor efficiency-wise is restricting the expansion term embedding selection to a small portion of the user-related term embeddings (15% of all of them, in our case). However, as the effectiveness of PQEWC is not inferior to its Non-approximated variant, the additional efficiency brought by the proposed approximation mechanism comes at no cost and still noticeably improves Query Expansion latency.

6.5 Qualitative Analysis

In this section, we carry out a qualitative analysis of the expansion terms selected by our proposed approach and highlight some limitations tied to the intelligibility of contextual word embeddings.

Table 9 lists some examples of tokens—obtained with the tokenizer used by BERT—corresponding to the expansion embeddings selected by the proposed approach. In many cases, highlighted in violet, it is straightforward to relate the tokens with the vocabulary of the topics underlying the original queries. We find that most of those tokens can carry supplementary information w.r.t. the corresponding query, thus clarifying the information needs and the user preferences. However, in other cases, we find understanding the relations between the original query

Table 9. Examples of Tokens Corresponding to the Embeddings Selected by the Proposed Approach for Expanding Queries from the Computer Science Dataset

Original Query	Tokens Corresponding to the Embeddings Selected for Expansion
zipf law password	solving called known signatures ##ecure provide schemes based key ##ing crypt clear others ##ing signing corresponding random key un attack ve whose authentication ##tion ##aries enables another storage distributed signatures normal ##ocation
time aware click model	believe million years users simply always click ##ting ##mming search would effectiveness sentiment micro correspond spa query behavior regardless results topic recommendation technically billion news examining usually user challenging ##ing ranking behaviors
5g internet things survey	considered aims presents through large ##t io ##ero current paper capabilities show drawn sensor introduced networks ga vu ##ns able internet things ##ization novel wireless concurrent ##it ##ch measurements data routing security
fundamental nonparametric bayesian inference	as posterior us distribution mixture prior dir ##ich applied ##pt bay ##esian ##tor ##iate ##s density developed estimation method ##metric rates ##let applications ##para compute important ##sities based est den est several
differential angular image material recognition	##es method develop applications ##ance disc ##ity challenging illumination images realize in performance coarse scenes ##per looks real un propose geometric appearances amount ##s recognition material scale objects applications 2d travels textures

Intelligible expansion tokens are highlighted.

terms and the tokens corresponding to the embeddings selected for expansion not as simple. For example, for the query "time aware click model" we can identify tokens related to the vocabulary of Personalization, Information Retrieval, and Recommender Systems. However, for many others, it is unclear how they relate to the query terms.

This situation is due to factors independent of the proposed approach itself: (1) many tokens are not intelligible due to the BERT tokenizer, as it splits some terms into multiple parts, and (2) all the tokens carrying the meaning of their context in their vector representation are unclear if analyzed as raw symbols. In the future, it may be worthwhile to employ a POS tagger to remove expansion embeddings that do not correspond to noun tokens, in case it was necessary to show expanded queries to users for explainability purposes. However, from our experience, in real-world applications, expansion terms are generally not shown to the user to avoid disorientation. Thus, as long as the expansion terms improve the retrieval effectiveness of the system, we believe this is a non-issue in most applications.

6.6 Summary

In this section, we summarize our findings w.r.t. the research questions introduced in Section 5, which we report here for simplicity.

- **RQ1** *Can a Personalized Query Expansion approach based on contextual word embeddings enhance ColBERT's retrieval effectiveness?* Although none of the considered Personalized Query Expansion baselines was able to consistently improve over ColBERT, our proposed approach PQWEC consistently and significantly outperformed ColBERT in all the considered datasets and for all the considered evaluation metrics.
- **RQ2** *Is our proposed Personalized Query Expansion approach more effective than previously proposed expansion methods?* Among both the compared Query Expansion approaches, PQWEC performed the best, significantly improving over the other approaches w.r.t. MAP@100, MRR@10, NDCG@10, and RBP.95 in almost all cases.
- **RQ3** *Is our proposed Personalized Query Expansion approach more robust than previously proposed expansion methods?* Robustness-wise, PQEWC achieved much higher Robustness Index scores than all the other considered Personalized Query Expansion methods. It also outperformed ColBERT-PRF in all cases but one, Computer Science without BM25's score fusion.
- **RQ4** *Is our proposed Personalized Query Expansion approach more efficient than previously proposed expansion methods?* PQEWC is more efficient than every other expansion method in our experimental evaluation, especially w.r.t. the best-performing baseline (ColBERT-PRF), and achieved sub-millisecond expansion time even in very data-rich scenarios.
- **RQ5** *Does our approach improve the expansion term diversity compared to previous Personalized Query Expansion methods?* Our approach selects topically coherent but not semantically overlapping expansion term embeddings, thus enhancing multiple aspects of the queries. Compared with those chosen by previous methods, the expansion terms selected by PQEWC benefit from higher diversity. These characteristics allowed our approach to reach significantly higher retrieval effectiveness than all the considered baselines in our comparative evaluation.
- **RQ6** *Does clustering the user term embeddings following the clusters of the collection term embeddings allow us to achieve better retrieval effectiveness than directly identifying the user-related clusters from the term embeddings of each user?* The procedure for clustering the user term embeddings we proposed in Section 3.1 allowed us to reach far better improvements over ColBERT than building the user-related clusters directly from the term embeddings of each user, which often achieved mixed results.
- **RQ7** *Is Equation (1) effective in identifying the user-term clusters that better represent the user's interests, thus enhancing the retrieval effectiveness of our proposed approach?* Using Equation (1) for identifying the user-term clusters that better represent the user's interests was generally more effective than selecting user-term clusters using other means.
- **RQ8** *Does our approximated expansion term selection perform on par of the original procedure proposed in Section 3.2 in terms of retrieval effectiveness?* For all the considered datasets, the approximation we proposed to improve the efficiency of the original expansion term selection procedure proposed in Section 3.2 did not negatively affect the retrieval effectiveness improvements brought by our Personalized Query Expansion method.
- **RQ9** *Does the approximation proposed to select the personalized expansion terms increase the efficiency w.r.t. the original procedure proposed in Section 3.2?* Although the original expansion term selection procedure proposed in Section 3.2 is already very efficient, our approximated variant still significantly improved the time needed to expand a given query.

7 CONCLUSION AND FUTURE WORK

In this work, we have addressed some issues arising from employing contextual word embeddings with current Personalized Query Expansion methods and proposed PQEWC, an approach designed to counteract those problems and take full advantage of contextual word embeddings. Specifically, our proposed method employs a clustering-based technique to group and identify the term embeddings most representative of the user interests and preferences, and an approximation procedure of the personalized expansion term selection to increase efficiency. Experimental evaluation shows the benefits of our proposed approach in terms of both efficiency and effectiveness. Moreover, it highlights how the effectiveness and efficiency of Personalized Query Expansion methods based on word embeddings can be greatly improved by adopting specialized procedures. Finally, the ablation study we conducted clearly illustrates the benefits of our design choices and the lack of drawbacks deriving from those.

Despite the significant improvements brought by our proposed Personalized Query Expansion method in terms of both efficiency and effectiveness, we think there still are related topics worth further study. As in previous works, we relied on cosine similarity to rank and select the expansion term embeddings. However, cosine similarity could be replaced by a more sophisticated parameterized function that, instead of just selecting the expansion terms by their semantic similarity with the original query terms, could evaluate their utility and assign them specific importance weights. Moreover, all the compared methods define the number of terms to add to the query as a fixed parameter, but this number is only generally good and not optimal for all the queries. Different queries could benefit from more expansion terms or work better without expansion. Furthermore, a weighing mechanism that can balance the importance of expansion terms one by one could improve the effectiveness brought by Query Expansion. Therefore, trying to predict the number of and the related weights for the expansion terms is a research direction still with much unexplored potential. How to adapt our Personalized Query Expansion technique to learned sparse retrieval models could also be a direction worth pursuing. Finally, the employment of more advanced user modeling techniques could further improve the retrieval effectiveness of Personalized Query Expansion.

A APPENDIX

Here we report an additional experiment to show that the relative performance of recent Transformer-based retrieval models on the benchmark datasets employed for our evaluation is similar to that shown in previous works on the standard dataset used nowadays for evaluating these models, MSMARCO [4]. Specifically, we compare ColBERT [47], BiEncoder [78], Splade$_{\text{MAX}}$[32], and CrossEncoder [70]. We additionally report BM25 performance scores for reference. We do not consider models fine-tuned with Knowledge Distillation [38–40, 103] or multi-step training procedures [99, 105] as there are no architectural differences w.r.t. the considered models, and they can be applied to all of them. An extensive comparison of those approaches on the benchmark datasets employed for our experimental evaluation is out of the scope of our work.

We trained all the considered models from scratch following the procedure described in Section 5.3, without limiting the embedding dimension of ColBERT to 16. Instead, we set it to 128, as proposed in the original paper [47]. We stress that although we can conduct this experiment by computing the representations of the queries and those of the documents to re-rank on the fly, we need to pre-compute those representations to tune the Query Expansion methods compared in Section 6.1, which is not possible with the hardware at our disposal, as described in Section 5.3.

Table 10 reports the result of this additional experiment. As expected, CrossEncoder performed the best across all the datasets. The second best model was ColBERT, followed by Splade$_{\text{MAX}}$ and BiEncoder. These results reflect the relative performance of recent Transformer-based retrieval models shown in previous works. Moreover, the robust improvements of ColBERT w.r.t. those of Splade$_{\text{MAX}}$ and BiEncoder corroborate the choice of this model as the backbone of the retrieval pipeline employed in the experiments presented in Section 6.1. Although the re-ranking time of the CrossEncoder amounts to over 1 second on GPU, making its applicability in real-world applications—one of the goals of our work—not feasible, its results suggest room for improvement.

Table 10. Effectiveness of the Compared Models

Model	Computer Science				Physics			
	MAP@100	MRR@10	NDCG@10	RBP.95	MAP@100	MRR@10	NDCG@10	RBP.95
BM25	12.25	48.92	22.45	13.22	12.77	53.68	26.88	16.05
BiEncoder	19.20	58.93	29.62	18.30	18.70	62.17	33.59	20.84
Splade$_{\text{MAX}}$	19.29	59.58	30.16	18.36	18.77	63.06	34.18	20.91
ColBERT	20.11†	60.24	30.75†	18.87†	19.69†	64.63†	35.08†	21.51†
CrossEncoder	**21.80**‡	**64.61**‡	**33.22**‡	**19.97**‡	**21.12**‡	**67.56**‡	**36.98**‡	**22.48**‡

Model	Political Science				Psychology			
	MAP@100	MRR@10	NDCG@10	RBP.95	MAP@100	MRR@10	NDCG@10	RBP.95
BM25	13.27	50.23	24.07	14.24	12.58	51.19	23.93	13.84
BiEncoder	17.54	54.74	27.64	17.23	22.64	64.85	34.87	21.01
Splade$_{\text{MAX}}$	18.39	57.16	29.18	17.77	22.52	65.43	35.04	20.92
ColBERT	19.07†	58.42†	29.96†	18.15†	23.52†	66.66†	36.06†	21.47†
CrossEncoder	**20.59**‡	**61.14**‡	**31.93**‡	**19.23**‡	**25.31**‡	**68.74**‡	**38.10**‡	**22.66**‡

The symbols † and ‡ denote significant improvements in a Bonferroni corrected Two-sided Paired Student's t-Test with $p < 0.005$ over BM25, Splade$_{\text{MAX}}$, and BiEncoder and over all models, respectively. Best results are highlighted in boldface.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 2623–2631. https://doi.org/10.1145/3292500.3330701

[2] Nawal Ould Amer, Philippe Mulhem, and Mathias Géry. 2016. Toward word embedding for personalized information retrieval. *CoRR* abs/1606.06991 (2016). arXiv:1606.06991. http://arxiv.org/abs/1606.06991

[3] Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: A survey. *Inf. Process. Manag.* 56, 5 (2019), 1698–1735. https://doi.org/10.1016/j.ipm.2019.05.009

[4] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. https://doi.org/10.48550/ARXIV.1611.09268

[5] Elias Bassani. 2022. ranx: A blazing-fast Python library for ranking evaluation and comparison. In *Proceedings on Advances in Information Retrieval - 44th European Conference on IR Research (ECIR '22), Part II (Lecture Notes in Computer Science, Vol. 13186)*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer, 259–264. https://doi.org/10.1007/978-3-030-99739-7_30

[6] Elias Bassani, Pranav Kasela, Alessandro Raganato, and Gabriella Pasi. 2022. A multi-domain benchmark for personalized search evaluation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22)*. ACM.

[7] Elias Bassani and Luca Romelli. 2022. ranx.fuse: A Python library for metasearch. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22)*. ACM.

[8] Matthias Bender, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Xavier Parreira, Ralf Schenkel, and Gerhard Weikum. 2008. Exploiting social relations for query expansion and result ranking. In *Proceedings of the 24th International Conference on Data Engineering Workshops (ICDE '08)*. IEEE Computer Society, 501–506. https://doi.org/10.1109/ICDEW.2008.4498369

[9] Marin Bertier, Rachid Guerraoui, Vincent Leroy, and Anne-Marie Kermarrec. 2009. Toward personalized query expansion. In *Proceedings of the 2nd ACM EuroSys Workshop on Social Network Systems (SNS '09)*, Tao Stein and Meeyoung Cha (Eds.). ACM, 7–12. https://doi.org/10.1145/1578002.1578004

[10] Claudio Biancalana and Alessandro Micarelli. 2009. Social tagging in query expansion: A new way for personalized web search. In *Proceedings of the 12th IEEE International Conference on Computational Science and Engineering (CSE '09)*. IEEE Computer Society, 1060–1065. https://doi.org/10.1109/CSE.2009.492

[11] Mohamed Reda Bouadjenek, Amyn Bennamane, Hakim Hacid, and Mokrane Bouzeghoub. 2013. Evaluation of personalized social ranking functions of information retrieval. In *Proceedings of the 13th International Conference on Web Engineering (ICWE '13), (Lecture Notes in Computer Science, Vol. 7977)*. Springer, 283–290.

[12] Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. 2013. Sopra: A new social personalized ranking function for improving web search. In *The 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13)*. ACM, 861–864.

[13] Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. 2019. Personalized social query expansion using social annotations. *Trans. Large Scale Data Knowl. Centered Syst.* 40 (2019), 1–25. https://doi.org/10.1007/978-3-662-58664-8_1

[14] Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Johann Daigremont. 2011. Personalized social query expansion using social bookmarking systems. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*, Wei-Ying Ma, Jian-Yun Nie, Ricardo Baeza-Yates, Tat-Seng Chua, and W. Bruce Croft (Eds.). ACM, 1113–1114. https://doi.org/10.1145/2009916.2010075

[15] Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Athena Vakali. 2013. Using social annotations to enhance document representation for personalized search. In *The 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '13)*. ACM, 1049–1052.

[16] Mohamed Reda Bouadjenek, Hakim Hacid, Mokrane Bouzeghoub, and Athena Vakali. 2016. PerSaDoR: Personalized social document representation for improving web search. *Inf. Sci.* 369 (2016), 614–633.

[17] Arbi Bouchoucha, Jing He, and Jian-Yun Nie. 2013. Diversified query expansion using conceptnet. In *22nd ACM International Conference on Information and Knowledge Management (CIKM '13)*, Qi He, Arun Iyengar, Wolfgang Nejdl, Jian Pei, and Rajeev Rastogi (Eds.). ACM, 1861–1864. https://doi.org/10.1145/2505515.2507881

[18] Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. 2014. Integrating multiple resources for diversified query expansion. In *Proceedings of Advances in Information Retrieval - 36th European Conference on IR Research (ECIR '14) (Lecture Notes in Computer Science, Vol. 8416)*, Maarten de Rijke, Tom Kenter, Arjen P. de Vries, ChengXiang Zhai, Franciska de Jong, Kira Radinsky, and Katja Hofmann (Eds.). Springer, 437–442. https://doi.org/10.1007/978-3-319-06028-6_38

[19] Arbi Bouchoucha, Xiaohua Liu, and Jian-Yun Nie. 2015. Towards query level resource weighting for diversified query expansion. In *Proceedings of Advances in Information Retrieval - 37th European Conference on IR Research (ECIR '15) (Lecture Notes in Computer Science, Vol. 9022)*, Allan Hanbury, Gabriella Kazai, Andreas Rauber, and Norbert Fuhr (Eds.). 1–12. https://doi.org/10.1007/978-3-319-16354-3_1

[20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020 (NeurIPS '20)*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[21] Silvia Calegari and Gabriella Pasi. 2008. Personalized ontology-based query expansion. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops*. IEEE Computer Society, 256–259. https://doi.org/10.1109/WIIAT.2008.242

[22] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*, Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong (Eds.). ACM, 243–250. https://doi.org/10.1145/1390334.1390377

[23] Claudio Carpineto and Giovanni Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.* 44, 1 (2012), 1:1–1:50. https://doi.org/10.1145/2071389.2071390

[24] Marc-Allen Cartright, James Allan, Victor Lavrenko, and Andrew McGregor. 2010. Fast query expansion using approximations of relevance models. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM '10)*, Jimmy X. Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). ACM, 1573–1576. https://doi.org/10.1145/1871437.1871675

[25] Paul-Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. 2007. Personalized query expansion for the web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07)*, Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando (Eds.). ACM, 7–14. https://doi.org/10.1145/1277741.1277746

[26] Kevyn Collins-Thompson. 2009. Reducing the risk of query expansion via robust constrained optimization. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy Lin (Eds.). ACM, 837–846. https://doi.org/10.1145/1645953.1646059

[27] Kevyn Collins-Thompson and Jamie Callan. 2007. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07)*, Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando (Eds.). ACM, 303–310. https://doi.org/10.1145/1277741.1277795

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '19), Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[29] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL '16), Volume 1: Long Papers*. Association for Computer Linguistics. https://doi.org/10.18653/v1/p16-1035

[30] William Falcon. 2019. Pytorch lightning. *GitHub*. https://github.com/PyTorchLightning/pytorch-lightning

[31] Michael Färber. 2019. The Microsoft academic knowledge graph: A linked data source with 8 billion triples of scholarly data. In *Proceedings of the Semantic Web - 18th International Semantic Web Conference (ISWC '19), Part II (Lecture Notes in Computer Science, Vol. 11779)*. Springer, 113–129.

[32] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR* abs/2109.10086 (2021). arXiv:2109.10086. https://arxiv.org/abs/2109.10086

[33] Enrique Frías-Martínez, George D. Magoulas, Sherry Y. Chen, and Robert D. Macredie. 2006. Automated user modeling for personalized digital libraries. *Int. J. Inf. Manag.* 26, 3 (2006), 234–248. https://doi.org/10.1016/j.ijinfomgt.2006.02.006

[34] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. 1987. The vocabulary problem in human-system communication. *Commun. ACM* 30, 11 (1987), 964–971. https://doi.org/10.1145/32206.32212

[35] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complement lexical retrieval model with semantic residual embeddings. In *Proceedings of Advances in Information Retrieval - 43rd European Conference on IR Research (ECIR '21), Part I (Lecture Notes in Computer Science, Vol. 12656)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 146–160. https://doi.org/10.1007/978-3-030-72113-8_10

[36] Gyeong June Hahm, Mun Yong Yi, Jae-Hyun Lee, and Hyo-Won Suh. 2014. A personalized query expansion approach for engineering document retrieval. *Adv. Eng. Informatics* 28, 4 (2014), 344–359. https://doi.org/10.1016/j.aei.2014.04.002

[37] Charles R. Harris, K. Jarrod Millman, Stéfan van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585 (2020), 357–362.

[38] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR* abs/1503.02531 (2015). arXiv:1503.02531. http://arxiv.org/abs/1503.02531

[39] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *CoRR* abs/2010.02666 (2020). arXiv:2010.02666. https://arxiv.org/abs/2010.02666

[40] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 113–122. https://doi.org/10.1145/3404835.3462891

[41] Nasreen Abdul Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah S. Larkey, Xiaoyan Li, Mark D. Smucker, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *Proceedings of the 13th Text REtrieval Conference (TREC '04) (NIST Special Publication, Vol. 500-261)*, Ellen M. Voorhees and Lori P. Buckland (Eds.). National Institute of Standards and Technology (NIST). http://trec.nist.gov/pubs/trec13/papers/umass.novelty.hard.pdf

[42] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. 2000. Real life, real users, and real needs: A study and analysis of user queries on the web. *Inf. Process. Manag.* 36, 2 (2000), 207–227. https://doi.org/10.1016/S0306-4573(99)00056-4

[43] Xu Jianmin and Liu Chang. 2012. Personalized query expansion based on user interest and domain knowledge. In *2012 3d Global Congress on Intelligent Systems*. IEEE, 394–399.

[44] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28, 1 (1972), 11–21.

[45] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP '20)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[46] Hamid Khalifi, Walid Cherif, Abderrahim El Qadi, and Youssef Ghanou. 2019. Query expansion based on clustering and personalized information retrieval. *Prog. Artif. Intell.* 8, 2 (2019), 241–251. https://doi.org/10.1007/s13748-019-00178-y

[47] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 39–48. https://doi.org/10.1145/3397271.3401075

[48] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Conference Track Proceedings of the 3rd International Conference on Learning Representations (ICLR '15)* , Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[49] John Krumm, Nigel Davies, and Chandra Narayanaswami. 2008. User-generated content. *IEEE Pervasive Comput.* 7, 4 (2008), 10–11. https://doi.org/10.1109/MPRV.2008.85

[50] Saar Kuzi, David Carmel, Alex Libov, and Ariel Raviv. 2017. Query expansion for email search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 849–852. https://doi.org/10.1145/3077136.3080660

[51] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query expansion using word embeddings. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM '16)*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 1929–1932. https://doi.org/10.1145/2983323.2983876

[52] Victor Lavrenko and W. Bruce Croft. 2001. Relevance-based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*, W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel (Eds.). ACM, 120–127. https://doi.org/10.1145/383952.383972

[53] Kyung-Soon Lee, W. Bruce Croft, and James Allan. 2008. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*, Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong (Eds.). ACM, 235–242. https://doi.org/10.1145/1390334.1390376

[54] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. *Pretrained Transformers for Text Ranking: BERT and Beyond.* Morgan & Claypool Publishers. https://doi.org/10.2200/S01123ED1V01Y202108HLT053

[55] Xiaohua Liu, Arbi Bouchoucha, Alessandro Sordoni, and Jian-Yun Nie. 2014. Compact aspect embedding for diversified query expansions. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, Carla E. Brodley and Peter Stone (Eds.). AAAI Press, 115–121. http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8350

[56] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Trans. Assoc. Comput. Linguistics* 9 (2021), 329–345. https://transacl.org/ojs/index.php/tacl/article/view/2383

[57] Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM J. Res. Dev.* 1, 4 (1957), 309–317.

[58] Yuanhua Lv and ChengXiang Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*, Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy (Eds.). ACM, 579–586. https://doi.org/10.1145/1835449.1835546

[59] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient document re-ranking for transformers by precomputing term representations. In *Proc. SIGIR.* 49–58.

[60] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Expansion via prediction of importance with contextualization. In *Proc. SIGIR.* 1573–1576.

[61] Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. PyTerrier: Declarative experimentation in Python from BM25 to dense retrieval. In *Proc. CIKM.* 4526–4533.

[62] J. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability.* Vol. 1.

[63] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval.* Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

[64] Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* 2, 11 (2017), 205. https://doi.org/10.21105/joss.00205

[65] Alessandro Micarelli, Fabio Gasparetti, Filippo Sciarrone, and Susan Gauch. 2007. Personalized search on the world wide web. In *The Adaptive Web, Methods and Strategies of Web Personalization (Lecture Notes in Computer Science, Vol. 4321)*, Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl (Eds.). Springer, 195–230. https://doi.org/10.1007/978-3-540-72079-9_6

[66] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Workshop Track Proceedings of the 1st International Conference on Learning Representations (ICLR '13),* Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1301.3781

[67] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013,* Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 3111–3119. https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[68] Philippe Mulhem, Nawal Ould Amer, and Mathias Géry. 2016. Axiomatic term-based personalized query expansion using bookmarking system. In *Proceedings of Database and Expert Systems Applications - 27th International Conference (DEXA '16), Part II (Lecture Notes in Computer Science, Vol. 9828)*, Sven Hartmann and Hui Ma (Eds.). Springer, 235–243. https://doi.org/10.1007/978-3-319-44406-2_17

[69] Shahrzad Naseri, Jeff Dalton, Andrew Yates, and James Allan. 2021. CEQE: Contextualized embeddings for query expansion. In *Proceedings of Advances in Information Retrieval - 43rd European Conference on IR Research (ECIR '21), Part I (Lecture Notes in Computer Science, Vol. 12656)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 467–482. https://doi.org/10.1007/978-3-030-72113-8_31

[70] Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). arXiv:1901.04085. http://arxiv.org/abs/1901.04085

[71] Pallavi Palleti, Harish Karnick, and Pabitra Mitra. 2007. Personalized web search using probabilistic query expansion. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops*. IEEE Computer Society, 83–86. https://doi.org/10.1109/WIIATW.2007.4427545

[72] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019 (NeurIPS '19)*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 8024–8035. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[73] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14), A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1532–1543. https://doi.org/10.3115/v1/d14-1162

[74] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '18), Volume 1 (Long Papers)*, Marilyn A. Walker, Heng Ji, and Amanda Stent (Eds.). Association for Computational Linguistics, 2227–2237. https://doi.org/10.18653/v1/n18-1202

[75] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. http://jmlr.org/papers/v21/20-074.html

[76] Sebastian Raschka, Joshua Patterson, and Corey Nolet. 2020. Machine learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *arXiv preprint arXiv:2002.04803* (2020).

[77] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410

[78] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP '19)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/v1/D19-1410

[79] Stephen E. Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (Special Issue of the SIGIR Forum)*, W. Bruce Croft and C. J. van Rijsbergen (Eds.). ACM/Springer, 232–241. https://doi.org/10.1007/978-1-4471-2099-5_24

[80] Joseph Rocchio. 1971. Relevance feedback in information retrieval. In *The Smart Retrieval System: Experiments in Automatic Document Processing*. 313–323.

[81] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using word embeddings for automatic query expansion. *CoRR* abs/1606.07608 (2016). arXiv:1606.07608. http://arxiv.org/abs/1606.07608

[82] Gerard Salton and Chris Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* 24, 5 (1988), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

[83] Sheikh Muhammad Sarwar, Md. Anowarul Abedin, A. H. M. Sofi Ullah, and Abdullah Al-Mamun. 2013. Personalized query expansion for web search using social keywords. In *The 15th International Conference on Information Integration and Web-based Applications & Services (IIWAS '13)*, Edgar R. Weippl, Maria Indrawan-Santiago, Matthias Steinbauer, Gabriele Kotsis, and Ismail Khalil (Eds.). ACM, 610. https://doi.org/10.1145/2539150.2539266

[84] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management*, Ottheín Herzog, Hans-Jörg Schek, Norbert Fuhr, Abdur Chowdhury, and Wilfried Teiken (Eds.). ACM, 824–831. https://doi.org/10.1145/1099554.1099747

[85] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (MAS) and applications. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. Association for Computing Machinery, New York, NY, 243–246.

[86] Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM '07)*, Mário J. Silva, Alberto H. F. Laender, Ricardo A. Baeza-Yates, Deborah L. McGuinness, Bjørn Olstad, Øystein Haug Olsen, and André O. Falcão (Eds.). ACM, 623–632. https://doi.org/10.1145/1321440.1321528

[87] Shayan A. Tabrizi, Azadeh Shakery, Hamed Zamani, and Mohammad Ali Tavallaei. 2018. PERSON: Personalized information retrieval evaluation based on citation networks. *Inf. Process. Manag.* 54, 4 (2018), 630–656. https://doi.org/10.1016/j.ipm.2018.04.004

[88] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait (Eds.). ACM, 449–456. https://doi.org/10.1145/1076034.1076111

[89] Robert L Thorndike. 1953. Who belongs in the family. In *Psychometrika*. Citeseer.

[90] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl. Acad. Sci.* 99, 10 (2002), 6567–6572.

[91] Nicola Tonellotto and Craig Macdonald. 2021. Query embedding pruning for dense retrieval. In *Proc. CIKM*. 3453–3457.

[92] Nicola Tonellotto, Craig Macdonald, and Iadh Ounis. 2018. Efficient query processing for scalable web search. *Found. Trends Inf. Retr.* 12, 4–5 (2018), 319–492.

[93] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[94] Endong Wang, Qing Zhang, Bo Shen, Guangyong Zhang, Xiaowei Lu, Qing Wu, and Yajuan Wang. 2014. Intel math kernel library. In *High-Performance Computing on the Intel® Xeon Phi™*, Jimmy X. Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). Springer, 167–188.

[95] Qihua Wang and Hongxia Jin. 2010. Exploring online social activities for adaptive search personalization. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM '10)*. ACM, 999–1008.

[96] Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2021. Pseudo-relevance feedback for multiple representation dense retrieval. In *The 2021 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '21)*, Faegheh Hasibi, Yi Fang, and Akiko Aizawa (Eds.). ACM, 297–306. https://doi.org/10.1145/3471158.3472250

[97] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *CoRR* abs/1910.03771 (2019). arXiv:1910.03771. http://arxiv.org/abs/1910.03771

[98] Xuan Wu, Dong Zhou, Yu Xu, and Séamus Lawless. 2017. Personalized query expansion utilizing multi-relational social data. In *12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP '17)*, Mária Bieliková and Marián Simko (Eds.). IEEE, 65–70. https://doi.org/10.1109/SMAP.2017.8022669

[99] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations (ICLR '21)*. OpenReview.net. https://openreview.net/forum?id=zeFrfgyZln

[100] Shengliang Xu, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. 2008. Exploring folksonomy for personalized search. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, 155–162.

[101] Yang Xu, Gareth J. F. Jones, and Bin Wang. 2009. Query dependent pseudo-relevance feedback based on Wikipedia. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*, James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel (Eds.). ACM, 59–66. https://doi.org/10.1145/1571941.1571954

[102] Omry Yadan. 2019. Hydra - A Framework for Elegantly Configuring Complex Applications. Github. https://github.com/facebookresearch/hydra

[103] Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum learning for dense retrieval distillation. In *The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 1979–1983. https://doi.org/10.1145/3477495.3531791

[104] ChengXiang Zhai and John D. Lafferty. 2001. A study of smoothing methods for language models applied to Ad Hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*, W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel (Eds.). ACM, 334–342. https://doi.org/10.1145/383952.384019

[105] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 1503–1512. https://doi.org/10.1145/3404835.3462880

[106] Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. 2020. BERT-QE: Contextualized query expansion for document re-ranking. In *Findings of the Association for Computational Linguistics (EMNLP '20) (Findings of ACL, Vol. EMNLP 2020)*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4718–4728. https://doi.org/10.18653/v1/2020.findings-emnlp.424

[107] Dong Zhou, Séamus Lawless, and Vincent Wade. 2012. Improving search via personalized query expansion using social media. *Inf. Retr.* 15, 3–4 (2012), 218–242. https://doi.org/10.1007/s10791-012-9191-2

[108] Dong Zhou, Xuan Wu, Wenyu Zhao, Séamus Lawless, and Jianxun Liu. 2017. Query expansion with enriched user profiles for personalized search utilizing folksonomy data. *IEEE Trans. Knowl. Data Eng.* 29, 7 (2017), 1536–1548. https://doi.org/10.1109/TKDE.2017.2668419

[109] Zhengyu Zhu, Jingqiu Xu, Xiang Ren, Yunyan Tian, and Lipei Li. 2007. Query expansion based on a personalized web search model. In *3rd International Conference on Semantics, Knowledge and Grid*. IEEE Computer Society, 128–133. https://doi.org/10.1109/SKG.2007.83

Received 18 November 2022; revised 19 July 2023; accepted 4 September 2023