ColBERT-Based User Profiles for Personalized Information Retrieval

Aleena Ahmad*, Gibson Nkhata†, Abdul Rafay Bajwa‡, Hannah Marsico†, Bryan Le†, Susan Gauch†

†Department of Electrical Engineering and Computer Science,
University of Arkansas, Fayetteville, USA
Email: {gnkhata, hmarsico, bryanle, sgauch}@uark.edu

‡Syed Babar Ali School of Science and Engineering,
Lahore University of Management Sciences, Lahore,
Pakistan
Email: 25100194@lums.edu.pk

*School of Electrical Engineering and Computer Science,
National University of Sciences and Technology, Islamabad, Pakistan
Email: alahmad.bscs21seecs@seecs.edu.pk

***Abstract*—Personalized Information Retrieval (PIR) improves search relevance by tailoring results to user interests using query history and browsing patterns. Traditional approaches to personalization range from feature engineering to the use of ontologies. Recently, there has been an increase in the exploration of deep learning models for this purpose. These models, such as Contextual Late Interaction over Bidirectional Encoder Representations from Transformers (ColBERT), provide token-level contextual embeddings that can be leveraged to generate semantic user profiles. State-of-the-art approaches use ColBERT to select candidate terms for personalized query expansion from user profiles. This approach poses challenges in accurately choosing user's descriptive keywords, risking the omission of crucial user preferences and repetitive selection of user terms. This study proposes a novel PIR approach that fully encodes user profiles using contextual embeddings and reranks Best Matching 25 (BM25) retrieved documents. Additionally, a frequency-recency weighting mechanism is tested which adjusts query influence based on temporal proximity and repetition frequency. Experimental results on two publicly available datasets demonstrate that our method improves retrieval performance, providing more accurate and context-aware search results.**

Keywords-Personalized information retrieval; user profile generation; ColBERT; reranking algorithms.

I. INTRODUCTION

With the exponential growth and complexity of information on the Web, it has been a daunting task for users to find relevant and interesting information [1]. Hence, Personalized Information Retrieval (PIR) was introduced to tailor search results according to a user's preferences and context, leveraging user-specific data such as query history, clicked documents, and browsing patterns. Unlike traditional retrieval systems that deliver uniform results, PIR systems dynamically adapt to user behavior, significantly enhancing relevance and search efficiency [2]–[4]. This evolution has been driven by advances in user profile modeling, semantic ontologies, and machine learning techniques.

The advent of deep learning and pre-trained language models, such as Bidirectional Encoder Representations from Transformers (BERT) [5], has revolutionized PIR. Models, such as Contextual Late Interaction over BERT (ColBERT) [6], which builds on top of BERT and combines token-level contextual embeddings with efficient retrieval mechanisms, have demonstrated superior performance. ColBERT enhances traditional retrieval methods like Best Matching 25 (BM25) [7] by reranking search results using token-level BERT embeddings. This hybrid approach has proven effective, as evidenced by its application in reranking documents retrieved by BM25 through query expansion [8][9]. On the same note, [9] employs a clustering-based procedure and uses ColBERT embeddings to identify the terms most representative of the user interests to be used for query expansion. Existing methods primarily use contextual word embeddings to select limited terms from user profiles for query expansion [9]–[12], as a result, risking the omission of crucial user preferences and repetitive selection of user's descriptive terms.

This work aims to overcome these limitations by integrating entire user profiles in the personalization approach. The major contributions of this work are outlined as follows:

1) **Full User Profile Representation:** Unlike previous methods which rely on term extraction, this work explores the impact of representing complete user profiles using contextual token-level embeddings, preserving all aspects of user preferences.
2) **Frequency-Recency Weighting Mechanism:** A novel weighting strategy is explored that combines the effect of query recency and frequency. An exponential decay function models temporal influence, while a logarithmic function balances frequent queries.
3) **Personalized Reranking with ColBERT:** ColBERT embeddings are leveraged as a second-stage algorithm to rerank candidate documents retrieved by BM25, ensuring more effective and context-aware personalization compared to term-based query expansion techniques.

The remainder of this paper is organized as follows. Section II reviews related work in PIR. Section III presents our proposed methodology, including user profile construction and our personalization model. Section IV details the experimental setup, datasets, and evaluation metrics. Finally, Section V concludes the paper, highlighting key findings and discussing directions for future work.

II. Related work

This section presents related work on PIR. Early work on PIR employed matching user profile keywords, extracted from previously visited documents, with document vectors by adapting the vector space model [2][13][14]. For example, [14] represents both user profiles and documents as vectors within the same term space, often derived from tags or keywords. The user profile vector encapsulates the user's interests based on previously interacted tags, while the document vector represents the content's tag distribution. This is an easy approach, but it still has shortfalls, e.g., the same user profile keyword can have multiple meanings, like bank as a financial institution vs. bank of a river, and dimensionality inconsistency between keywords and document vectors, leading to irrelevant retrieved results.

To address ambiguity issues, other researchers used ontologies to model user profiles [3][4]. For example, [3] adapts the navigation of information based on a user profile structured as a weighted concept hierarchy. Specifically, every web page visited by a user is classified into this concept hierarchy, and the resulting ontologies are then used for either reranking search results or filtering relevant documents. This approach combated the polysemy problem associated with keywords, still, it lacks context awareness, as deep semantic relationships among words in the documents are not encoded by the ontological user profiles.

Because of their ability to capture words in context, state-of-the-art approaches to PIR use pre-trained word embeddings to model documents and user profiles [6][8][15][16]. These techniques merely leverage contextual word embeddings for query expansion.

While recent work has employed pre-trained word embeddings for PIR, it has not exploited them to fully represent user profile information. As described in the aforementioned works, retrospectively, the common practice in incorporating embeddings is to use them to select terms from the user profile to expand the user query. While this provides personalization, some crucial user information may be overlooked by the selection process, and sometimes, the same terms might habitually be selected from the user profile as candidates for query expansion. Our work, on the other hand, deviates from literature by incorporating the entire user profile into the retrieval process, instead of using it to extract expansion terms. It generates an embedding-based representation of the entire profile, which is used directly to rerank results returned by BM25.

III. Methodology

This section describes the methodology employed to develop and evaluate the proposed approach. The first step was selecting and preprocessing datasets suitable for testing personalization. Next, user profiles were generated using user histories and provided as input for the personalization model. Finally, the results are reranked by the personalization model. Each of these steps is detailed in the following subsections.

A. Dataset and Preprocessing

Experiments on personalization require user-specific data, such as previously issued queries, clicked documents, etc. [3] Two publicly available datasets suitable for this purpose were used.

*1) **AOL4PS** :* This is a dataset generated from the American Online (AOL) query logs. The authors [17] processed the original query logs to construct a dataset suitable for personalization. Each query record has an associated user id, timestamp, session information, the Uniform Resource Locators (URLs) of the top ten retrieved documents, and the index position of the clicked document. The original dataset statistics are presented in Table I.

TABLE I. AOL4PS STATISTICS

Metric	Value
Total number of records	1,339,101
Number of users	12,907
Average number of records per user	103.75
Unique records per user (mean)	47.23

The dataset only included URLs for documents - not textual content - which was required for building user profiles and computing similarity. Hence, the first step was to download the textual content of each URL by scraping the web. Given the dataset's age, many of the URLs in the dataset were no longer available. To recover historical content, the Wayback Machine [18] was used where possible. Each URL was retained only if it was accessible and had sufficient content to be used meaningfully for indexing and similarity computation. This filtering found only **158,235** URLs from the original **951,941** to be valid. Based on these available URLs, we applied the following record-level filtering:

- Records were removed if the **clicked URL** was not available.
- Records were removed if none of the **10 retrieved documents** were available.

The statistics of the filtered data are presented in Table II. The number of records overall and per-user was considerably reduced post-filtering. Further filtering was done before a sample of test users could be extracted, the details of which are presented in Section IV.

TABLE II. DATASET STATISTICS AFTER INITIAL FILTERING

Metric	Value
Total number of records	276,459
Number of users	12,493
Average number of records per user	22.13
Unique records per user (mean)	11.04

*2) **Personalized Results Reranking Benchmark (PRRB)**:* This is a multi-domain dataset, proposed by [9], used for personalized search evaluation. It consists of datasets divided into four domains: Computer Science, Physics, Psychology, and Political Science. It has a total of 1.9 million queries divided across these four domains. The PERSON methodology [19] was used for the construction of the dataset, using published papers to develop triplets of users, queries, and documents. In particular, a paper's title is considered as a query, one of the authors is considered as the user, and the referenced papers are considered as relevant documents. Detailed statistics of the datasets are presented in Table III.

B. Profile Generation

To create representative profiles, we gathered the available data from each user's history. Based on the approaches previously followed [17], we incorporated relevant documents and previously issued queries. The queries issued by users are a direct statement of interest; hence they were included. Similarly, the title and content of the clicked document were included, as they were the ones that the user considered relevant for a particular query.

For a given user u, their profile P_u is constructed by concatenating their past issued queries and clicked documents:

The user profile P_u is represented as:

$$P_u = (Q_u, D_u)$$

where:

- Q_u is the set of past queries issued by the user:

$$Q_u = \{q_1, q_2, \dots, q_N\}$$

- D_u represents the set of past clicked documents (titles and content):

$$D_u = \{d_1, d_2, \dots, d_M\}$$

where: - N is the number of past queries. - M is the number of past clicked documents.

Each document d_i consists of a title t_i and content c_i:

$$d_i = [t_i, c_i]$$

Thus, the final profile representation can be expressed as:

$$P_u = \left(\sum_{i=1}^{N} q_i, \sum_{j=1}^{M} (t_j, c_j) \right)$$

For each dataset, the profile generation process is detailed below:

*1) **AOL4PS**:* As stated in Section III-B above, the associated information with each user in this dataset includes the text of previously issued queries and the corresponding clicked documents for each query.

*2) **PRRB**:* For this dataset, each user's issued queries were titles of the papers authored by the user. The documents in a user's profile were other papers authored by this user. For consistency with AOL4PS, the papers' titles and contents are analogous to user's past queries and clicked-on retrieved documents, respectively.

C. Personalization Model

The final content of a user's profile is represented as contextual word embeddings. The embedding model used for this purpose is ColBERT v2 [6].

The personalization approach tested in this work consists of the following steps:

1) Prior to testing, all documents originally retrieved for each test query are obtained, represented as embeddings, and indexed with ColBERT. This happens in an offline stage.
2) When testing, two arguments are passed to ColBERT's searcher:
   - The 'query' against which the documents will be ranked. In our case, the user profile serves as the query.
   - The list of document IDs that were originally retrieved for the given query. This ensures that the similarity calculation and reranking is done only for the associated documents of a query, instead of the entire index.
3) In cases where the user profile exceeds ColBERT's 32-token limit, it is split into 32-token chunks. Each chunk then separately reranks the associated documents.
4) The results of each chunk are aggregated using maximum pooling, allowing a document to have a high score if it matches any chunk of the user's profile. These aggregated results then become the final reranked document list against a query.

In addition to this basic approach, the Frequency-Recency approach is tested which differs in how the profile is used to rerank the results.

*1) **Frequency-Recency Approach**:* This approach incorporates frequency and recency information of a query into the reranking process. Each query in the dataset had an associated timestamp, which is used to calculate the time difference between it and the currently tested query. In addition, repetitions of queries are taken into consideration.To adjust the influence of past queries based on their time of occurrence and repetition, we define the following weighting functions:

Recency-Based Weighting: We hypothesize that user queries issued in the past may become less relevant over time. To model this, we apply an *exponential decay function*, which reduces the weight of older queries:

$$w_{\text{recency},i} = e^{-\alpha \cdot \Delta t_i} \tag{1}$$

where:

- $w_{\text{recency},i}$ is the recency weight assigned to the historical query i.
- Δt_i is the time difference (in days) between the test query and the past query.
- α is a decay parameter that controls how quickly the influence of older queries diminishes.

A higher value of α causes past queries to decay faster, reducing their contribution to reranking.

Frequency-Based Weighting: Users often repeat queries when searching for specific information. Queries that appear frequently in a user's history likely indicate stronger preferences.

TABLE III. STATISTICS OF THE DATASETS

	PRRB			
	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
# sessions	-	-	-	-
# clicked documents	-	-	-	-

To account for this, we apply a logarithmic transformation to query frequency:

$$w_{\text{frequency},i} = \log(1 + \beta \cdot f_i) \tag{2}$$

where:

- $w_{\text{frequency},i}$ is the frequency weight for query i.
- f_i is the number of times query i has been issued by the user.
- β is a scaling factor that controls the influence of frequency on the final weight.

Using a logarithm ensures that the effect of very high frequencies is moderated, preventing overly frequent queries from dominating the reranking.

Final Weighting Function: To combine both recency and frequency effects, the final query-document pair weight is computed as:

$$w_i = w_{\text{recency},i} \cdot w_{\text{frequency},i} = e^{-\alpha \cdot \Delta t_i} \cdot \log(1 + \beta \cdot f_i) \tag{3}$$

where:

- w_i is the overall weight assigned to the query-document pair.
- The recency term $e^{-\alpha \cdot \Delta t_i}$ ensures that older queries have less influence.
- The frequency term $\log(1 + \beta \cdot f_i)$ ensures frequently issued queries have greater weight.

Prior research has shown the effectiveness of exponential decay in modeling recency [20] and log-based frequency weighting in ranking models [21]. The values of α and β are tuned experimentally to optimize performance. Figure 1 illustrates the entire model framework.

IV. Experiments

This section delves into the experimental details of this work.

A. Experimental Setup

To use ColBERT, we utilized the Python RAGatouille [22] library. This is a framework for easier access and setup of ColBERT, particularly when using Google Colab, where all our experiments were conducted. It provides all functionalities that can be used with barebone ColBERT.

*1) **AOL4PS**:* In addition to the process described in Section III-B, further filtering was performed to ensure constant profile size across a set of users. To do so, users were classified into buckets based on the number of query records available for profile creation. The distribution of the number of query records per users was observed, and the following buckets were chosen **10-15, 16-25, 26-35, 36-45, 46-above records.** Each bucket had an associated profile length, shown in Table IV. This excluded users with fewer than 10 valid records, which formed a major proportion of the filtered dataset. The records of the remaining users were divided into train and test sets. For this, the records were first sorted by time. The initial n records were chosen for the train set (to build user profiles), where n was equal to the profile length of the bucket.

TABLE IV. USER BUCKETS AND ASSOCIATED PROFILE LENGTHS

Bucket	**Tested Profile Length**
10-15	5
16-25	10
26-35	20
36-45	30
46 and above	40

For the test set, the remaining records for each user after extracting train set were chosen. From these, only the records where at least 5 of the URLs originally retrieved against the query were available were kept. This ensured our ranking of the clicked document was standardized.

In both sets, to ensure that all records did not consist of repeated queries, the number of repeated queries was limited to 1/3rd of the length of the records in the set.

As a result, 349 users remained that satisfied the aforementioned filtering and bucketing strategy. These were all useed for testing. and satisfied the filtering criteria and bucketing strategy described above. The distribution of these 349 users across each bucket, and other statistics, are shown in Table V.

Each user bucket was tested not only with its assigned profile length but also with all smaller profile lengths from lower buckets, enabling analysis of how profile size impacts personalization performance.

*2) **PRRB**:* For this dataset, a random sample of 1000 queries was selected from each domain and tested with the baseline and

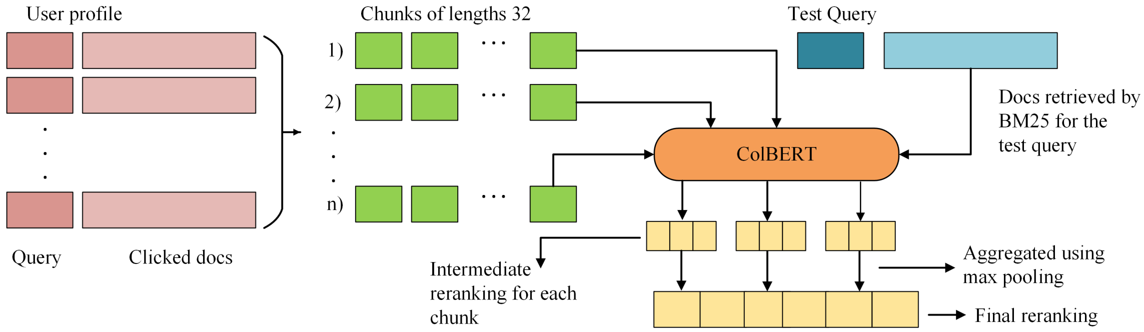

Figure 1: The model framework.

TABLE V. USER BUCKET STATISTICS

Bucket	# Users	# Train Records	# Test Records
10-15	56	280	295
16-25	157	1570	858
26-35	43	860	229
36-45	18	540	94
46 and above	75	3000	569

personalization model. Statistics for this sample are presented in Table VI.

TABLE VI. PRRB STATISTICS FOR SAMPLE 1000.

Domain	# Users	# User Docs	# Docs/User (Avg)
Computer Science	881	78 516	104
Physics	803	61 394	104
Psychology	937	70 399	84
Political Science	837	38038	52

B. Evaluation Metrics

To evaluate the performance of the considered models, the following metrics are employed: (1) Mean Average Precision (MAP), (2) Mean Reciprocal Rank (MRR), as primary evaluation metrics.

*1) **MAP**:* The average precision of the relevant retrieved documents averaged across a set of queries.

*2) **MRR**:* The average of the reciprocal ranks of the first relevant result for a set of queries.

C. Baselines

This section introduces the baselines employed in the comparative evaluation. First, our model is compared with BM25, to assess whether our approach can effectively rerank retrieved documents. Then we consider personalized query expansion approaches based on word embeddings, to verify if our proposed approach is improving over the state-of-the-art techniques. Similar to our work, personalized query expansion techniques are second-stage retrieval algorithms, where BM25 is a first-stage retriever.

1) **BM25** [7]: A ranking function that scores documents based on their relevance to a given query. It uses Term Frequency (tf), Inverse Document Frequency (idf), and document length normalization.
2) **ColBERT-PRF** [10]: A query expansion method based on ColBERT with reliance on Pseudo-Relevance Feedback (PRF) [23]. 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 clustering algorithm and selects the tokens corresponding to the cluster centroids with higher idf scores for query expansion.
3) **Query Expansion for Email Search (QEES)** [11]: A query expansion approach that begins by calculating the cosine similarity between each user-related term embedding and every query term embedding. These similarity scores are then transformed into a probability distribution using softmax normalization. Finally, the method aggregates the log probabilities for each user-related term embedding and selects the highest-scoring terms to expand the query.
4) **Query Expansion with Enriched User Profiles (QEEUP)** [12]: A query expansion technique that 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 query.
5) **Personalized Query Expansion with Contextual Word Embeddings (PQEWC)** [15]: A query expansion technique that builds on ColBERT and devises HDBSCAN [24], a hierarchical density-based clustering method, to identify the terms that better represent the user interests.

Since these baselines' setups are inconsistent with AOL4S, they are evaluated on PRRB only, while AOL4S is used for comparing various derivations of our model.

D. Results

Table VII shows PRRB results using MAP@100 and MRR@10, reflecting the top 100 and 10 reranked results, respectively. Table VIII presents AOL4PS results using MRR@10 and MAP@1. These metrics correspond to previous work in PIR with these datasets, such as [15], [25]. Best scores per domain or bucket are shown in bold. Additional testing of the Personalization Approach with different profile lengths for

TABLE VII. EXPERIMENTAL RESULTS ON PRRB.

Model	Computer Science		Physics		Psychology		Political Science	
	MAP@100	MRR@10	MAP@100	MRR@10	MAP@100	MRR@10	MAP@100	MRR@10
BM25	0.1511	0.4826	0.1295	0.5551	0.2122	0.6297	0.1713	0.5430
ColBERT-PRF	0.1856	0.5682	0.1877	0.6150	0.2192	0.6253	0.1642	0.5351
QEES	0.1813	0.5632	0.1783	0.6118	0.2142	0.6285	0.1598	0.5305
QEEUP	0.1818	0.5686	0.1805	0.6256	0.2137	0.6276	0.1549	0.5285
PQEWC	0.1903	0.5766	0.1917	0.6381	0.2230	0.6421	0.1724	0.5510
Ours	**0.2026**	**0.5871**	**0.1919**	**0.6495**	**0.2278**	**0.6493**	**0.1840**	**0.5694**

TABLE VIII. EXPERIMENTAL RESULTS ON AOL4PS.

Buckets	BM25		ColBERT Non Personalized		Personalization Approach		Recency-Frequency	
	MRR@10	MAP@1	MRR@10	MAP@1	MRR@10	MAP@1	MRR@10	MAP@1
10-15	0.3311	0.1559	0.3671	0.1322	**0.5723**	0.2780	0.5285	**0.2848**
16-25	0.3249	0.1480	0.3789	0.1317	**0.5822**	**0.2984**	0.5304	0.2879
26-35	0.3188	0.1222	0.3906	0.1354	**0.5921**	**0.3188**	0.5186	0.2751
36-45	0.4160	0.2553	0.3425	0.0957	**0.6447**	**0.3404**	0.5111	0.2660
46-above	0.3452	0.1706	0.3684	0.1255	**0.6463**	**0.3667**	0.5102	0.2647

TABLE IX. TESTING WITH DIFFERENT PROFILE LENGTHS

Buckets	5		10		20		30		40	
	MRR@10	MAP@1	MRR@10	MAP@1	MRR@10	MAP@1	MRR@10	MAP@1	MRR@10	MAP@1
10-15	**0.5723**	**0.2780**	-	-	-	-	-	-	-	-
16-25	0.4433	0.0991	**0.5822**	**0.2984**	-	-	-	-	-	-
26-35	0.4203	0.1004	0.4438	0.1223	**0.5921**	**0.3188**	-	-	-	-
36-45	0.4247	0.1064	0.4404	0.1383	0.4679	0.1596	**0.6447**	**0.3404**	-	-
46-above	0.4718	0.1176	0.4735	0.1255	0.4877	0.1353	0.4918	0.1294	**0.6463**	**0.3667**

each bucket are shown in Table IX, allowing for an analysis on how profile length effects personalization effectiveness.

E. Discussion

It can be observed that the proposed personalization approach consistently outperforms the baselines in each test setup. In experiments with the PRRB dataset, the personalization approach outperforms BM25 and other baseline query-expansion methods. Most notably, the greatest improvement in MRR@10 is observed in Computer Science and Physics. The query expansion techniques in Table VII employ pre-trained word embeddings in selecting terms to expand the query, thus outperforming the baseline. Still, they struggle against our method because we use contextual word embeddings to encode the entire user profile, thereby encoding subtle nuances and attributes that help in reranking relevant documents for PIR.

For the AOL4PS dataset, the use of contextual embeddings, with or without personalization, consistently outperforms the BM25 baseline across all buckets and profile sizes. A discrepancy is observed in the 36-45 bucket, where the *ColBERT Non Personalized* approach's results are lower than the BM25 results. This may be due to data limitations or statistical variance. However, the overall trend indicates that personalization significantly enhances ranking effectiveness, particularly as user profile data increases. This is supported by the fact that greater bucket sizes (which corresponds to the amount of data used for profile generation) correspond to higher MRR values.

Furthermore, the results in Table IX show that, for each set of users, the MRR and MAP values generally increase as the profile size increases. The best performance is seen when the greatest number of records are incorporated in the profile.

The recency-frequency weighting approach, while not surpassing our base personalization model, does offer improvements against the BM25 and ColBERT Non-Personalized approaches. An interesting observation is that it performs better with smaller user profiles. This could imply that, with sufficient user history, the need to integrate recency and frequency diminishes and the information needed for personalization can be obtained from the textual content of the profile itself.

Response Time Analysis:
The indexing of documents and the creation of user profiles occur offline, before testing. At runtime, similarity calculations are performed between the user profile chunks with each document and the results are aggregated. Hence, the response time for each query depends on the length of the user profile (number of documents incorporated). The average response times for each profile length when tested with the proposed Personalization approach are summarized in Table X. These times are observed from experimentation done with Google Colab's A100 GPU.

Memory Analysis:

TABLE X. PROFILE LENGTHS AND AVERAGE RESPONSE TIME PER QUERY

Profile Length	Average Response Time (s)
5	3.69
10	7.04
20	14.37
30	22.11
40	29.85

The personalized approach stores the top-k candidate documents and the user profile in memory to compute similarity scores. Memory usage thus depends on the number of documents per profile and the candidate set size.

For AOL4PS, with k = 10 and an average document size of 9 KB, candidate documents require 90 KB. Including the profile, total memory ranges from 135 KB (5-document profile) to 450 KB (40-document profile).

For PRRB, k = 100 and average document size 1 KB, yielding 100 KB for candidates. Including the profile, total memory ranges from 105 KB to 140 KB for 5–40 document profiles.

V. Conclusion and Future Work

This work presents a novel PIR approach that encodes entire user profiles using contextual word embeddings and re-ranks BM25 retrieved documents. Additionally, it tests a frequency-recency weighting mechanism to study the impact of temporal proximity and repetition on personalization performance. Through experimentation on two publicly available datasets, the effectiveness of our approach is confirmed. For the PRRB dataset, our proposed personalization model consistently outperforms BM25 ranking and query-expansion baselines. For the AOL4PS dataset, personalization improves ranking across all user profile sizes, with larger profiles showing better results. The recency-frequency approach offers improvements relative to the baseline, however it benefits users with limited search history more. Overall, this work reinforces the idea that utilizing complete user profiles for PIR is an effective approach. It also highlights the potential of deep learning-based methods to develop rich representations.

Further work can build on the limitations of our study. Our testing with AOL4PS involved a small user sample due to limited valid URLs. Expanding URL scrapping to different geographic locations could increase access to URLs, allowing a greater number of valid records and users for testing. Future work can also study profiling techniques which integrate session information and provide efficient scaling for real-time personalization.

Acknowledgment

This work was supported by the National Science Foundation under Award number OIA-1946391, Data Analytics that are Robust and Trusted (DART).

References

[1] S. Gauch, “Conceptual recommender system for citeseerx”, 2009.

[2] N. J. Belkin and W. B. Croft, “Retrieval techniques”, *Annual Review of Information Science and Technology (ARIST)*, vol. 28, pp. 109–145, 1993.

[3] S. Gauch, J. Chaffee, and A. Pretschner, “Ontology-based personalized search and browsing”, *Web Intelligence and Agent Systems: An international Journal*, vol. 1, no. 3-4, pp. 219–234, 2003.

[4] A. Sieg, B. Mobasher, and R. Burke, “Web search personalization with ontological user profiles”, in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 525–534.

[5] J. Devlin, “Bert: Pre-training of deep bidirectional transformers for language understanding”, *arXiv preprint arXiv:1810.04805*, 2018.

[6] O. Khattab and M. Zaharia, “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*, 2020, pp. 39–48.

[7] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, “Okapi at trec-3: At the interface between probabilistic and vector space models”, in *Proceedings of the Third Text REtrieval Conference (TREC-3)*, National Institute of Standards and Technology (NIST), 1994, pp. 109–126.

[8] A. Salemi, S. Kallumadi, and H. Zamani, “Optimization methods for personalizing large language models through retrieval augmentation”, *arXiv preprint arXiv:2404.05970*, 2024.

[9] E. Bassani, P. Kasela, A. Raganato, and G. Pasi, “A multi-domain benchmark for personalized search evaluation”, in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3822–3827.

[10] X. Wang, C. Macdonald, N. Tonellotto, and I. Ounis, “Pseudo-relevance feedback for multiple representation dense retrieval”, in *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, 2021, pp. 297–306.

[11] S. Kuzi, D. Carmel, A. Libov, and A. Raviv, “Query expansion for email search”, in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 849–852.

[12] D. Zhou, X. Wu, W. Zhao, S. Lawless, and J. Liu, “Query expansion with enriched user profiles for personalized search utilizing folksonomy data”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1536–1548, 2017.

[13] G. Salton, A. Wong, and C. Yang, “A vector space model for automatic indexing”, *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[14] Z. Xu, J. Zhang, J. Li, and J. Li, “Personalized information retrieval through user profile based document filtering”, in *Proceedings of the 2008 IEEE International Conference on Information and Automation*, IEEE, 2008, pp. 1861–1865.

[15] E. Bassani, N. Tonellotto, and G. Pasi, “Personalized query expansion with contextual word embeddings”, *ACM Transactions on Information Systems*, vol. 42, no. 2, pp. 1–35, 2023.

[16] C. Richardson *et al.*, “Integrating summarization and retrieval for enhanced personalization via large language models”, *arXiv preprint arXiv:2310.20081*, 2023.

[17] Q. Guo, W. Chen, and H. Wan, “Aol4ps: A large-scale data set for personalized search”, *Data Intelligence*, vol. 3, no. 4, pp. 548–567, Oct. 2021, ISSN: 2641-435X. DOI: 10.1162/dint_a_00104. eprint: https://direct.mit.edu/dint/article-pdf/3/4/548/1968580/dint_a_00104.pdf.

[18] I. Archive, *Wayback machine*, Accessed April 5, 2025.

[19] S. A. Tabrizi, A. Shakery, H. Zamani, and M. A. Tavallaei, “Person: Personalized information retrieval evaluation based on citation networks”, *Information Processing & Management*, vol. 54, no. 4, pp. 630–656, 2018.
[20] P. Ardagelou and A. Arampatzis, “A half-life decaying model for recommender systems with matrix factorization”, in *TDDL/MDQual/Futurity@ TPDL*, 2017.
[21] D. Vianna and A. Marian, “A frequency-based learning-to-rank approach for personal digital traces”, *arXiv preprint arXiv:2012.13114*, 2020.
[22] B. Clavié, *Ragatouille*, version 0.0.9, Available at: https://github.com/AnswerDotAI/RAGatouille, 2023.
[23] J. J. Rocchio Jr, “Relevance feedback in information retrieval”, *The SMART retrieval system: experiments in automatic document processing*, 1971.
[24] L. McInnes, J. Healy, and S. Astels, “Hdbscan: Hierarchical density based clustering.”, *J. Open Source Softw.*, vol. 2, no. 11, p. 205, 2017.
[25] J. Yao, Z. Dou, and J.-R. Wen, “Employing personal word embeddings for personalized search”, in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1359–1368.