AOL4PS: A Large-scale Data Set for Personalized Search (Postprint)

Authors: Guo, Qian, Chen, Wei, Huaiyu Wan, Wan, Huaiyu

Date: 2022-11-18T00:00:00+00:00

Abstract

Personalized search is a promising way to improve the quality of Web search, and it has attracted much attention from both academic and industrial communities. Much of the current related research is based on commercial search engine data, which cannot be released publicly for such reasons as privacy protection and information security. This leads to a serious lack of accessible public datasets in this field. The few publicly available datasets have not become widely used in academia because of the complexity of the processing process required to study personalized search methods. The lack of datasets together with the difficulties of data processing has brought obstacles to fair comparison and evaluation of personalized search models. In this paper, we constructed a large-scale dataset AOL4PS to evaluate personalized search methods, collected and processed from AOL query logs. We present the complete and detailed data processing and construction process. Specifically, to address the challenges of processing time and storage space demands brought by massive data volumes, we optimized the process of dataset construction and proposed an improved BM25 algorithm. Experiments are performed on AOL4PS with some classic and state-of-the-art personalized search methods, and the experiment results demonstrate that AOL4PS can measure the effect of personalized search models.

Full Text

Preamble

AOL4PS: A Large-scale Data Set for Personalized Search

Qian Guo[1],[2], Wei Chen[1],[2] & Huaiyu Wan[1],[2]†

[1]School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
[2]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing 100044, China

Keywords: Personalized search; Text data processing; Data set construction

Citation: Guo, Q., et al.: AOL4PS: A large-scale data set for personalized search. Data Intelligence 3(4), 548-567 (2021). doi: 10.1162/dint_a_{00104}

Received: March 12, 2021; **Revised:** April 25, 2021; **Accepted:** May 18, 2021

ABSTRACT

Personalized search represents a promising approach to improving Web search quality and has attracted considerable attention from both academic and industrial communities. However, much of the current research relies on commercial search engine data that cannot be publicly released due to privacy protection and information security concerns, leading to a severe shortage of accessible public datasets in this field. The few publicly available datasets have not gained widespread adoption in academia because of the complex processing required to study personalized search methods. This lack of datasets, combined with the difficulties of data processing, has created obstacles for fair comparison and evaluation of personalized search models. In this paper, we construct AOL4PS, a large-scale dataset for evaluating personalized search methods, collected and processed from AOL query logs. We present a complete and detailed data processing and construction pipeline. Specifically, to address the computational time and storage space challenges posed by massive data volumes, we optimized the dataset construction process and proposed an improved BM25 algorithm. Experiments conducted on AOL4PS using several classic and state-of-the-art personalized search methods demonstrate that AOL4PS can effectively measure the performance of personalized search models.

Corresponding author: Huaiyu Wan (Email: hywan@bjtu.edu.cn; ORCID: 0000-0003-1747-3472)

1. INTRODUCTION

The search engine is one of the primary means by which people obtain useful information from the Internet. When given a query, a search engine ranks documents according to the degree of matching between the query and the document. Generally, a non-personalized search engine returns the same results for the same query from different users, ignoring their varied hidden interests. However, for the same query, the true intentions of different users are often different, particularly when the query is ambiguous. For example, for the query "Giant,"some users may seek information about Giant Bicycles, while others may be interested in the film titled *Giant*, and still others may want to learn about

the English word "giant." Non-personalized search engines cannot distinguish such varied intents.

Personalized search, designed to return a personalized list of documents for users, offers one approach to addressing the query ambiguity problem described above. With the proliferation of the Internet and big data, personalized search has become an important technology in modern search engines.

Although personalized search is receiving increasing attention from researchers, accessing suitable datasets remains challenging. Moreover, despite the large volume of datasets available in the field of information retrieval, most are not appropriate for personalized search research. The reasons include: (1) datasets from commercial search engines are not public; (2) datasets lack necessary information such as long-term click behaviors, unified user identifiers, or raw text of queries and documents; and (3) datasets have not been processed in a uniform manner.

For instance, LETOR [1], a benchmark collection for information retrieval, lacks temporal information about user historical behaviors. Similarly, SogouQ [2], query logs from the Sogou search engine, lacks unified user identifiers. Some publicly available datasets, such as Yandex and SEARCH17 [3], do not provide raw text for queries or documents. For another well-known collection, AOL query logs [4, 5], no publicly available personalized search datasets have been derived.

In this paper, building upon the work of [6], we propose a complete and detailed dataset construction process to create AOL4PS, a dataset for personalized search derived from AOL query logs. When generating candidate documents for queries, we propose an improved BM25 algorithm to enhance computational efficiency and reduce storage space requirements. Additionally, we provide comprehensive statistics for AOL4PS and conduct experiments to validate its effectiveness using several classic and state-of-the-art personalized search methods.

The remainder of this paper is structured as follows. Section 1 introduces the motivation for personalized search and the dataset scarcity problem. Section 2 reviews the current landscape of personalized search datasets. Section 3 describes the complete dataset construction process and presents the content and statistics of AOL4PS. Section 4 analyzes the distribution of AOL4PS and describes validation experiments. Finally, Section 5 concludes our work.

2. RELATED WORK

Crowdsourced datasets. Anikó et al. [7] collected two real-world datasets by posting tasks on Amazon' s Mechanical Turk, recruiting 300 workers (200 for Google Search, 100 for Bing). Kumar and Sharan [8] used browsing histories from 10 different users for 50 queries to simulate search scenarios. Generally, query logs constructed via crowdsourcing suffer from limited numbers of users

and queries, as well as time-consuming collection processes, which are critical limitations for personalized search where large-scale user and query data are essential.

Non-public datasets. Datasets derived from MSN query logs [9, 10], anonymized logs of the Microsoft Bing search engine [11, 12, 13], and query logs from the Yahoo search engine [14] are based on commercial search engine data that are either unpublished or no longer available.

Datasets without raw text. In 2013, Yandex released a large-scale anonymized dataset for the "Yandex Personalized Web Search Challenge." Similarly, Nguyen et al. [3] released SEARCH17 in 2019. The Yandex dataset contains information on anonymized user identifiers, queries, query terms, URLs, URL domains, and clicks. However, despite its large scale, Yandex' s anonymization of queries and URLs prevents researchers from accessing raw text. Due to this lack of raw text, machine learning models not based on textual information have been widely used on the Yandex dataset [15, 16]. However, such datasets are unsuitable for the deep learning era, where text rich in semantic information is crucial for enhancing performance across various natural language processing tasks through representation learning.

AOL query logs. In 2006, America Online (AOL) released query logs suitable for information retrieval, query recommendation, and personalized search. A key advantage of AOL query logs is that they contain the original corpus of queries and documents. Compared to Yandex and SEARCH17, which lack raw text, AOL query logs are more suitable for deep learning-based methods. Carman et al. [17, 18] improved topic models for personalized search, generating new ranked lists rather than re-ranking existing ones. Tyler et al. [19] proposed a re-ranking method utilizing re-finding behavior recognition. They resubmitted each query to the AOL search engine and crawled returned documents to generate candidates. Ahmad et al. [6] used AOL query logs for document ranking and query suggestion, crawling document titles and generating candidate documents using the BM25 algorithm to address the absence of candidate documents in the original logs.

In previous studies, the processing of AOL query logs for personalized search dataset construction has not been sufficiently documented. Existing personalized search datasets based on AOL query logs at different scales are largely unavailable publicly [4, 5]. The absence of a complete and clear construction process and the lack of publicly available personalized search datasets represent a gap in the research field. Therefore, a unified approach to constructing personalized search datasets from AOL query logs is both necessary and significant. This paper presents a comprehensive processing pipeline for AOL query logs to construct a publicly available personalized search dataset and validates its effectiveness on several classic and state-of-the-art personalized search models.

3.1 Content of AOL Query Logs

In 2006, AOL Search released a collection of user query logs comprising a large number of queries from 657,426 users over a three-month period [20]. The original content of AOL query logs is shown in Table 1 .

Specifically, AnonID is an anonymous user ID for privacy protection. For example, the entry "142 westchester.gov 2006-03-20 03:55:57 1 http://www.westchestergov.com" indicates that user ID 142 submitted the query "westchester.gov" on March 20, 2006, and clicked on http://www.westchestergov.com, which was ranked #1.

Table 1. Field descriptions for AOL query logs.

Field	Description
AnonID	An anonymous user ID number
Query	The query issued by the user, case-shifted with most punctuation removed
QueryTime	The time at which the query was submitted for search
ItemRank	If the user clicked on a search result, the rank of the clicked item
ClickURL	If the user clicked on a search result, the domain portion of the URL in the clicked result

The statistical summary of AOL query logs is shown in Table 2 .

Table 2. Statistics of AOL query logs.

Metric	Value
Number of distinct user IDs	657,426
Number of distinct queries	10,154,742
Number of distinct documents	1,632,789
Number of records	36,389,567

3.2 Data Construction Method

AOL query logs contain only records of user-clicked documents for queries, without records of candidate documents returned by the search engine. Therefore, we generated candidate document lists for each query and annotated documents that users found satisfactory. Given the large scale of AOL query logs, which consumes substantial time and space during dataset construction, traditional data processing approaches are impractical under limited hardware resources.

To address this bottleneck, we proposed an improved BM25 algorithm to reduce time consumption in data processing.

3.2.1 Data Preprocessing

Data preprocessing removes erroneous and redundant data. Although the logs are structured, the log file is not a database, and logs were not validated for completeness when generated, so there is no guarantee that server-generated logs are correct or complete. Additionally, data redundancy occurs when log files are merged, as commercial search engines typically use multiple servers to handle large query volumes, causing the same record to appear multiple times in the final merged logs.

3.2.2 Data Crawling

We crawled document text since AOL query logs contain only URLs, not text content. Crawling serves two purposes: first, relevance scores are calculated based on text matching between queries and documents during dataset construction; second, text content of both queries and documents constitutes an important feature type in personalized search models.

Following [6], we crawled webpage titles corresponding to documents. For example, the title for "www.orlandosentinel.com" is "Orlando news, weather, sports, business | Orlando Sentinel." Crawling titles offers two benefits: first, titles are more similar to queries than body text [21]; second, using titles instead of body text significantly reduces storage requirements.

Crawled titles are categorized into four groups based on their text content (Table 3): (1) titles containing valid information; (2) titles with no crawled text (e.g., NAN or whitespace); (3) titles with only invalid information (e.g., "404 Not Found," "403 Forbidden," "502 Bad Gateway"); and (4) titles in non-English characters (e.g., Chinese, Japanese, Russian).

Table 3. Category percentages and examples of crawled titles.

Category	Percentage	Examples
Valid text	-	New Cars, Used Cars For Sale, Car Prices & Reviews at Automotive.com
No text	-	NAN; white space
Invalid text	-	404 Not Found; 403 Forbidden; Access Denied
Non-English text	-	Tomaszów Mazowiecki, Łódz´, mieszkania

Since AOL query logs were released in 2006, many URLs no longer exist or content has been updated. Approximately 35.72% of documents have titles identical to their domain name (e.g., "http://www.clanboyd.info" has the title "clanboyd.info"). Given missing textual content in many documents, we use domain names as document text.

3.2.3 Data Cleaning

The goal of data cleaning is to preserve English and numeric content in texts (including queries and documents). Our text cleaning methods include normalization, tokenization, word segmentation, lemmatization, and stemming.

Data Cleaning Process. The cleaning flow comprises two stages: first, normalization, tokenization, and word segmentation are performed sequentially; second, depending on task requirements, either lemmatization or stemming is applied. For example, when calculating similarity between queries and documents, stemming minimizes word count to facilitate matching. When computing representations of queries and documents, lemmatization reduces memory usage while retaining semantic information.

Data Cleaning Effect. We evaluate cleaning effectiveness using word coverage rate, defined in Equation (1):

$$\text{WordCoverageRate} = \frac{N}{n}$$

where N represents the number of words in AOL query logs' queries and documents, and n represents the number of words in GloVe6b, an open-source dataset containing 400,000 words total. The word coverage rates are 72.32% for stemming and 93.52% for lemmatization. Calculating word coverage rate helps assess cleaning effectiveness; rates exceeding 70% demonstrate effective cleaning. Stemming reduces word count but tends to generate tokens rather than words and may lose original meaning. Lemmatization retains more word forms and thus provides richer semantic information.

3.2.4 Document Similarity Calculation

Introduction to BM25. BM25 [22] is a classic information retrieval algorithm widely used in mature search engines such as Lucene and Elasticsearch. Based on probabilistic retrieval models, BM25 evaluates similarity between queries and documents through relevance scores measuring word matching degree. We generate candidate document series by ranking these relevance scores. BM25 scores are calculated using Equation (2):

$$\text{Score}(Q, d) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f_i(q_i, d) \cdot (k+1)}{f_i(q_i, d) + k \cdot \left(1 - b + b \cdot \frac{dl}{\text{avgdl}}\right)}$$

where Q is the query to retrieve, query Q contains words $\{q_1, q_2, ...\}$, n represents the number of words in query Q, d is a document returned by the search engine, $f_i(q_i, d)$ represents word q_i frequency in d, dl is document length, and avgdl is the average length of all documents. Parameters k and b adjust the effect of document length on similarity, with $k = 2$ and $b = 0.75$. IDF(q_i) represents the inverse document frequency of q_i, calculated using Equation (3):

$$\mathrm{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

where N is the number of documents in AOL query logs and $n(q_i)$ is the number of documents containing retrieval word q_i.

Improved BM25 Algorithm. BM25 is an efficient and stable information retrieval algorithm. Following [6], we used BM25 to compute relevance scores between queries and documents, then generated candidate documents for all queries after ranking these scores. Although many open-source NLP tools provide BM25 interfaces, their computation time does not meet large-scale data requirements. As shown in Table 2, AOL query logs contain millions of queries and documents, making relevance score calculations on the order of billions. To solve excessive computation time, we proposed an improved BM25 algorithm using matrix operations to enhance efficiency. The improved algorithm accelerates the original by at least 6 times, with relevance scores calculated using Equation (4):

$$\mathrm{RevanceScores} = X \times \mathrm{IDF} \times \frac{(k+1) \times F}{F + k \times \left(1 - b + b \times \frac{dl}{\mathrm{avgdl}}\right)}$$

where $X \in \mathbb{R}^{ql \times wl}$ represents word frequencies in queries, $F \in \mathbb{R}^{wl \times dl}$ represents word frequencies in documents, IDF $\in \mathbb{R}^{wl \times wl}$ is a diagonal matrix of inverse document frequencies, $dl \in \mathbb{R}^{dl}$ represents document lengths, and avgdl is the average length of all documents. ql, wl, and dl represent query length, word length, and document length, respectively.

Block Matrix Multiplication. Our improved BM25 algorithm is a matrix implementation of the original. While matrix computation saves time, matrix storage presents a critical challenge. As shown in Table 2, the number of queries exceeds 3 million, documents exceed 1 million, and words total approximately 80,000. Such large matrices require hundreds of gigabytes of memory. However, the average query length is about 4 words and document length about 7 words, indicating these matrices are very sparse. Sparse matrices can reduce storage space, but while X^T, IDF, and F are sparse, the relevance scores matrix produced by our improved BM25 algorithm is dense, still causing memory shortages during computation. Consequently, we employed block matrix multiplication to reduce memory usage. Additionally, we optimized document extraction via matrix multiplication when sorting relevance scores.

Extract Related Documents. After calculating relevance scores with the improved BM25 algorithm, we extracted related documents from the full document set for each query. Following [23], we treated the top 1,000 documents as relevant and remaining documents as irrelevant. Queries with at least one relevant document clicked by users were retained as valid; otherwise, they were removed. This focuses the dataset on queries with related documents. We then deleted queries without related documents and generated related documents for retained queries. We centered clicked documents in a window and selected surrounding documents as candidates. To reduce storage and accelerate training, we set window size to 10.

3.2.5 Data Annotation

SAT-click. A SAT-click is defined as a click with dwell time exceeding a predefined threshold (typically 30 seconds) or the last click in a search session [24, 25, 26]. SAT-clicks serve as indicators of true user interests.

Session Partition. A session comprises a series of queries issued by the same user over a short period. Queries submitted within a session are generally believed to share related intent. Following [27], we first used a 30-minute interval for session partitioning, then used cosine similarity between successive queries. Queries are represented by TF-IDF weighted sum of word embedding vectors. Following [6], the cosine similarity threshold for session partition is set to 0.5.

3.2.6 Data Division

We divided AOL query logs into four categories: historical data, training data, validation data, and test data. Historical data creates user profiles, while training, validation, and test data are used to train, validate, and test personalized search models. Based on [28], we split historical and other data by query time. For 12 weeks of data, we used the first 9 weeks as historical data. The last 3 weeks were divided into training, validation, and test data with a 4:1:1 ratio. Additionally, we filtered out users with no clicks in the first 9 weeks or fewer than 6 records in the last 3 weeks.

3.3 Contents and Statistics of AOL4PS

The processed dataset contents are shown in Table 4 , with statistics provided in Tables 5 and 6.

Table 4. Fields and descriptions for AOL4PS.

Field	Description
AnonID	An anonymous user ID number
Query	The query issued by the user

Field	Description
QueryTime	The time at which the query was submitted
ClickPos	The position of clicked candidate documents
ClickDoc	The document clicked by the user
CandidateList	A list of candidate documents returned for the query
SessionNo	The session number of a user search action

Table 5 . Basic statistics of AOL4PS.

Metric	Value
Date range	2006/3/1-2006/5/31
#users	12,907
#queries	1,339,101
#distinct queries	382,222
#distinct URLs	746,998
#sessions	953,592
#SAT-clicks	1,339,101
#queries/#users	103.75
#sessions/#users	73.88
#SAT-clicks/#users	103.75
#queries/#sessions	1.40
#SAT-clicks/#sessions	1.40
#SAT-clicks/#distinct queries	3.50
#SAT-clicks/#distinct URLs	1.79
average query length	3.23
average document length	6.87

Table 6 . Division of AOL4PS.

Split	#queries	#sessions
Training	218,559	155,386
Validation	54,230	38,977
Test	53,357	38,977

3.4 Comparison of Data Sets

Table 7 compares AOL4PS with existing datasets Yandex and SEARCH17. AOL4PS' s advantage lies in its original text information, including query content and document URLs, whereas Yandex and SEARCH17 contain only query IDs and document IDs without original text. In personalized search, longer

query time spans and more query records help models learn more accurate user interest features. Simultaneously, moderate numbers of users and query records make hardware and time resource consumption more manageable during model training. AOL4PS offers advantages through its rich original content, large user base, long query time spans, and extensive user query records.

Table 7. Comparison of AOL4PS with existing datasets.

Dataset	#Days	#Users	#Queries	#Queries/#Users	Text Information
Yandex	-	5,736,333	64,693,054	11.28	No original text, only query IDs and document IDs
SEARCH17	8,016	1,339,101	12,907	103.75	No original text, only query IDs and document IDs
AOL4PS	91	12,907	1,339,101	103.75	Original query text and document URLs

3.5 Usage of AOL4PS

As the current query moves backward in a user' s query sequence, all preceding queries are regarded as historical queries for the current query. Therefore, multiple samples can be constructed from a single user query sequence. As shown in Figure 1 [Figure 1: see original paper], assuming a user has 15 query records, with 9 historical queries, 4 training queries, and 1 validation and 1 test query, we can construct 4 training samples, 1 validation sample, and 1 test sample. In each sample, as the current query moves backward, the number of historical queries used to generate user interest increases. In AOL4PS, each user query sequence can support construction of training, validation, and test samples.

Figure 1. Sample construction description.

4.1 Distribution of Queries

In AOL4PS, over 60% of distinct queries appear only once during the 3-month period, and approximately 87% are single-user queries. These statistics align with those reported in [9] for commercial query logs. About 46% of queries in the test set appear in the training set. Furthermore, 35.75% of repeated queries in the test set are resubmitted by the same user.

4.2 Distribution of Query Click Entropy

We utilized query click entropy [29] to measure query ambiguity and the degree of personalization needed, calculated using Equation (5):

$$\text{ClickEntropy}(q) = -\sum_{d \in P(d)} P(q|d) \log_2 P(q|d)$$

where q represents a user-issued query, d represents a document returned by the search engine, and ClickEntroy(q) is the click entropy of query q. $P(d)$ is the collection of documents clicked for query q. $P(q|d)$ is the percentage of clicks on document d among all clicks on query q, calculated using Equation (6):

$$P(q|d) = \frac{\text{Clicks}(q,d)}{\sum_{d' \in P(d)} \text{Clicks}(q,d')}$$

where Clicks(q,d) represents the number of clicks on document d for query q and the denominator represents the total number of clicks on query q.

As shown in Figure 2 [Figure 2: see original paper], we calculated click entropy for queries issued by multiple users. Higher click entropy indicates greater need for personalized search optimization in AOL4PS. Over 30% of queries have click entropy greater than 0, demonstrating clear need for personalized search optimization.

Figure 2. Distribution of query click entropy.

4.3 Distribution of Sessions

Figure 3 [Figure 3: see original paper] shows the distribution of queries per session. Over 25% of sessions contain at least two queries, indicating that users sometimes submit multiple queries to fulfill an information need. This observation aligns with findings in [9].

Figure 3. Distribution of query number per session.

4.4 Distribution of Users

Figure 4 Figure 4: see original paper shows the distribution of historical days, and Figure 4(b) shows query frequency in search history. We found that 68% of users in the test set have histories spanning over 25 days, and about 68% submit more than 50 queries during their historical period. Figure 5 Figure

5: see original paper shows user query frequency distribution over the 3-month period, with over 94% of users submitting at least 50 queries. Figure 5(b) shows session count distribution, with over 70% of users having at least 50 sessions. This indicates that AOL4PS, built on long-term historical behaviors, provides sufficient data for learning user interest features.

(a) Distribution of user historical query days
(b) Distribution of user historical query times

Figure 4. Distribution of user historical queries.

(a) Distribution of user query times
(b) Distribution of session numbers of users

Figure 5. Distribution of user queries.

4.5 Evaluation Measures

We measured personalized search accuracy using mean reciprocal rank (MRR), P@K, and average click position.

4.5.1 MRR

Mean reciprocal rank (MRR) is the average of reciprocal ranks of all SAT-clicks, defined in Equation (7):

$$\mathrm{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\mathrm{rank}_i}$$

where rank_i is the rank of the first relevant document in the ranking list, and N is the number of documents.

4.5.2 P@K

The second metric is P@K, measuring accuracy of top K documents for a given query. Since users typically focus on the first few results, we use Equation (8):

$$P@K = \frac{n}{N}$$

where n represents the number of satisfied documents ranked within the top K, N is the number of documents, and we set $K = 1$ in this paper.

4.5.3 Avg.Click

We further use average click position (Avg.Click) [30] of SAT-clicks to evaluate re-ranking effectiveness. Lower Avg.Click indicates better personalized search ranking, defined in Equation (9):

$$\text{Avg.Click} = \frac{1}{N}\sum_{i=1}^{N}\text{ClickPosition}_i$$

where ClickPosition_i is the position of SAT-click i in the ranking list, and N is the number of documents.

4.6 Comparison Methods and Experiment Settings

Personalized search is a typical learning-to-rank application in information retrieval. Learning-to-rank methods are generally classified into three categories: pointwise, pairwise, and listwise methods [31], which transform ranking into regression, classification, or other tasks. In AOL4PS, we labeled whether users clicked documents and the rank of clicked documents. The ground truth in personalized search is the relative rank of document pairs. We evaluated five classic or state-of-the-art personalized search models: statistics-based methods (BM25, P-Click), a listwise method (SLTB), and pairwise methods (HRNN, GRADP).

BM25. As previously mentioned, BM25 [22] is a non-personalized document ranking algorithm that determines query-document relevance through text matching. The original ranking of all candidate documents in AOL4PS derives from BM25. We set adjustment factors $k = 1.5$ and $b = 0.75$.

P-Click. Dou et al. [9] proposed P-Click, which re-ranks documents based on click counts under the same query. P-Click is biased toward repeated queries. We set smoothing parameter b in the score calculation function to 0.5.

SLTB. Bennett et al. [11] analyzed user interests by extracting over 100 features from short- and long-term historical behaviors. All features train a LambdaMart [32] model to generate personalized rankings. SLTB is among the best machine learning-based models for personalized search. Implementation parameters: number of leaves = 70, minimum instances in leaf node = 2000, learning rate = 0.3, number of trees = 50.

HRNN. Ge et al. [28] used hierarchical recurrent neural networks with query-aware attention to dynamically build short- and long-term user profiles based on the current query. Documents are then re-ranked using the user profile and other features. Implementation parameters: word embedding size = 200, short-term interest vector size = 200, long-term interest vector size = 600, hidden units in attention MLP = 512, learning rate = 1e-3.

GRADP. Zhou et al. [33] used recurrent neural networks with attention mechanisms to capture the dynamicity and randomness of user interests. They extracted several query features including click entropy and topic entropy to build effective user profiles. Implementation parameters: word embedding size = 200, GRU hidden size = 600, hidden units in attention MLP = 512, n in nCS = 2, n in nRS = 3, learning rate = 1e-3.

4.7 Experiment Results and Analysis

We split AOL4PS into four sub-datasets based on user query record counts, each containing over 3,000 users. To reduce storage and time costs, we selected only 5 candidate documents from the full set of 10. Table 8 shows model performance on the four sub-datasets.

Table 8. Performance of different personalized search models.

Dataset	Method	MRR	P@1	Avg.Click
Data1	BM25	0.6554	0.4734	2.0295
	P-Click	0.7212	0.5718	1.8270
	SLTB	0.7437	0.6075	1.7647
	HRNN	0.8096	0.7126	1.6822
	GRADP	0.8077	0.7099	1.6874
Data2	BM25	0.7212	0.5718	1.8270
	P-Click	0.7437	0.6075	1.7647
	SLTB	0.8246	0.7297	1.5193
	HRNN	0.8280	0.7424	1.6219
	GRADP	0.8407	0.7596	1.5674
Data3	BM25	0.7437	0.6075	1.7647
	P-Click	0.8246	0.7297	1.5193
	SLTB	0.8472	0.7693	1.5458
	HRNN	0.8472	0.7693	1.5458
	GRADP	0.8556	0.7811	1.5118
Data4	BM25	0.8246	0.7297	1.5193
	P-Click	0.8472	0.7693	1.5458
	SLTB	0.8926	0.8369	1.3811
	HRNN	0.8926	0.8369	1.3811
	GRADP	0.8894	0.8318	1.3918

All personalized models (P-Click, SLTB, HRNN, GRADP) perform well across sub-datasets, significantly outperforming the non-personalized BM25 on MRR, P@1, and Avg.Click, demonstrating personalization effectiveness. SLTB outperforms P-Click, indicating that complex machine learning models are promising. P-Click uses only one feature, while SLTB employs over 100 features, showing that machine learning models rely heavily on feature quality and quantity.

HRNN and GRADP outperform traditional machine learning models (P-Click, SLTB), especially on Data3 and Data4, consistent with their performance on another personalized search dataset [28]. Both models use recurrent neural networks and attention mechanisms. HRNN models the impact of short- and long-term historical behaviors on the current query, while GRADP addresses query dynamicity and randomness. They learn accurate user interest features from long-term histories, demonstrating deep learning superiority in personalized search. Deep learning models rely on user query sequence modeling, whereas traditional models depend heavily on manually designed features. Results show deep learning models excel with long query records and have become the mainstream approach.

Figure 6 [Figure 6: see original paper] shows MRR statistics across sub-datasets. BM25 results remain consistent across all four sub-datasets. SLTB steadily improves original rankings but shows no incremental trend. P-Click, HRNN, and GRADP show incremental improvements across sub-datasets because total user queries increase sequentially, enabling these models to capture user interests more accurately with more historical data. For deep learning-based personalized search methods, more data yields more accurate user interest information. AOL4PS' s large user base and query records effectively test personalized search models.

Figure 6. Results on different sub-datasets.

In summary, AOL4PS is a large-scale personalized search dataset with over 10,000 users and more than 1 million query records. User histories are sufficiently long to support accurate user interest learning. AOL4PS can effectively evaluate personalized search model performance.

---

5. CONCLUSION AND FUTURE WORK

This paper addresses the lack of public datasets in personalized search research by proposing a complete, detailed processing pipeline for AOL query logs. We constructed AOL4PS, a high-quality large-scale dataset for personalized search containing extensive users with long-term historical behaviors. We validated AOL4PS through experiments with several typical personalized search models, demonstrating its applicability and superiority for personalized search tasks.

Future improvements include: First, many documents were unavailable during crawling due to content updates or removal; we excluded these from AOL4PS. Future work could retrieve document content through the Internet Archive' s historical website information or existing information retrieval datasets. Second, we constructed candidate document lists for each user click, whereas real retrieval scenarios may involve multiple clicks per query. Future work could merge queries to generate candidate lists with multiple clicks.

ACKNOWLEDGEMENTS

This work was supported by the National Key R&D Program of China (No. 2018YFC0830200).

AUTHOR CONTRIBUTIONS

This work resulted from collaboration among all authors. H. Wan (hywan@bjtu.edu.cn) led the project and organized the paper content. Q. Guo (qianguo@bjtu.edu.cn) collected and constructed the dataset, ran experiments, and wrote the paper. W. Chen (w_{chen}@bjtu.edu.cn) performed data statistics and experimental analysis. All authors contributed through manuscript revision and proofreading.

DATA AVAILABILITY STATEMENT

All data are available in the Science Data Bank repository under Attribution 4.0 International (CC BY 4.0), http://www.doi.org/10.11922/sciencedb.j00104.00093.

REFERENCES

[1] Qin, T., et al.: LETOR: A benchmark collection for research on learning to rank for information retrieval. Information Retrieval 13, 346–374 (2010)

[2] Liu, Y., et al.: How do users describe their information need: Query recommendation based on snippet click model. Expert Systems with Applications 38, 13847–13856 (2011)

[3] Nguyen, D.Q., et al.: A capsule network-based embedding model for knowledge graph completion and search personalization. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2180–2189 (2019)

[4] Yao, J., Dou, Z., Wen, J.: Employing personal word embeddings for personalized search. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1359–1368 (2020)

[5] Lu, S., et al.: Knowledge enhanced personalized search. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 709–718 (2020)

[6] Ahmad, W.U., Chang, K., Wang, H.: Context attentive document ranking and query suggestion. In: Proceedings of the 42nd International ACM SIGIR

Conference on Research and Development in Information Retrieval, pp. 385–394 (2019)

[7] Anikó, H., et al.: Measuring personalization of Web search. Computing Research Repository abs/1706.05011 (2017)

[8] Kumar, R., Sharan, A.: Personalized Web search using browsing history and domain knowledge. In: Proceedings of 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), pp. 493–497 (2014)

[9] Dou, Z., Song, R., Wen, J.: A large-scale evaluation and analysis of personalized search strategies. In: Proceedings of the 16th International Conference on World Wide Web, pp. 581–590 (2007)

[10] Dou, Z., et al.: Evaluating the effectiveness of personalized Web search. IEEE Transactions on Knowledge and Data Engineering 21, 1178–1190 (2009)

[11] Bennett, P.N., et al.: Modeling the impact of short- and long-term behavior on search personalization. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 185–194 (2012)

[12] Sontag, D.A., et al.: Probabilistic models for personalizing Web search. In: Proceedings of the Fifth International Conference on Web Search and Web Data Mining, pp. 433–442 (2012)

[13] Lu, et al.: PSGAN: A minimax game for personalized search with limited and noisy click data. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 555–564 (2019)

[14] Wedig, S., Madani, O.: A large-scale analysis of query logs for assessing personalization opportunities. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 742–747 (2006)

[15] Zhou, L.: Personalized Web search. Computing Research Repository abs/1502.01057 (2015)

[16] Yoganarasimhan, H.: Search personalization using machine learning. Management Science 66, 1045–1070 (2020)

[17] Carman, M.J., et al.: Towards query log based personalization using topic models. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, pp. 1849–1852 (2010)

[18] Harvey, M., Crestani, F., Carman, M.J.: Building user profiles from topic models for personalised search. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 2309–2314 (2013)

[19] Tyler, S.K., Wang, J., Zhang, Y.: Utilizing re-finding for personalized information retrieval. In: Proceedings of the 19th ACM Conference on Information and Knowledge Management, pp. 1469–1472 (2010)

[20] Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: Proceedings of the 1st International Conference on Scalable Information Systems, pp. 1–7 (2006)

[21] Huang, P., et al.: Learning deep structured semantic models for Web search using clickthrough data. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 2333–2338 (2013)

[22] Robertson, S.E., et al.: Okapi at TREC-7: Automatic ad hoc, filtering, vlc and interactive track. NIST Special Publication SP, 253–264 (1999)

[23] Dehghani, M., et al.: Neural ranking models with weak supervision. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 65–74 (2017)

[24] Harvey, M., Crestani, F., Carman, M.J.: Building user profiles from topic models for personalised search. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 2309–2314 (2013)

[25] Fox, S., et al.: Evaluating implicit measures to improve Web search. ACM Transactions on Information Systems 23, 147–168 (2005)

[26] Gao, J., et al.: Smoothing clickthrough data for Web search ranking. In: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 355–362 (2009)

[27] Luo, X., Ping, F., Chen, M.: Clustering and tailoring user session data for testing Web applications. In: Proceedings of the 2nd International Conference on Software Testing Verification and Validation, pp. 336–345 (2009)

[28] Ge, S., et al.: Personalizing search results using hierarchical RNN with query-aware attention. Computing Research Repository abs/1908.07600 (2019)

[29] Mei, Q., Church, K.W.: Entropy of search logs: How hard is search? with personalization? with backoff? In: Proceedings of the International Conference on Web Search and Web Data Mining, pp. 45–54 (2008)

[30] White, R.W., et al.: Enhancing personalized search by mining and modeling task behavior. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 1411–1420 (2013)

[31] Cao, Z., et al.: Learning to rank: From pairwise approach to listwise approach. In: Proceedings of the 24th International Conference on Machine Learning, pp. 129–136 (2007)

[32] Wu, Q., et al.: Adapting boosting for information retrieval measures. Information Retrieval 13, 254–270 (2010)

[33] Zhou, Y., et al.: Dynamic personalized search based on RNN with attention mechanism. Chinese Journal of Computer 42, 812–826 (2019)

AUTHOR BIOGRAPHY

Qian Guo received her B.E. degree in computer science and technology from Beijing Jiaotong University, China, in 2018. She is currently pursuing a Master's degree at the School of Computer and Information Technology, Beijing Jiaotong University. Her research interests include personalized search and knowledge representation learning.

Wei Chen received his Master' s degree in Computer Science and Technology from Guilin University of Electronic Technology, China, in 2020. He is currently pursuing a PhD degree at the School of Computer and Information Technology, Beijing Jiaotong University. His research interests include knowledge graph reasoning and recommendation systems.

Huaiyu Wan received his PhD degree in Computer Science and Technology from Beijing Jiaotong University, China. He is currently an Associate Professor with the School of Computer and Information Technology, Beijing Jiaotong University. His research interests include spatial-temporal data mining, social network mining, and information extraction.

ORCID: 0000-0003-1747-3472

Note: Figure translations are in progress. See original paper for figures.