DATA PAPER

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 Traflc 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 is a promising way to improve the quality of Websearch, 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 can not be released publicly for such reasons as privacy protection and information security. This leads to a serious lack of accessible public data sets in this field. The few publicly available data sets have not become widely used in academia because of the complexity of the processing process required to study personalized search methods. The lack of data sets 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 data set 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 data set 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.

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

1. INTRODUCTION

The search engine is one of the primary ways that people obtain useful information from the Internet. When given a query, search engine ranks documents according to the matching degree between the query and the document. Generally, the search engine without personalization always returns the same results for the same query from different users and ignores their varied hidden interests. However, the fact is that for the same query, the real intentions of different users are often different. This is especially obvious when the query is ambiguous. For example, for the query "Giant", some users want to find the information about the Giant Bike, while some other users want to access the content related to a film named Giant, and still some other users want to learn about the English word giant. That is to say, non-personalized search engines cannot distinguish such queries.

Personalized search, designed to return a personalized list of documents for users, is one of the approaches to the query ambiguity problem as described above. With the popularity of the Internet and big data, personalized search has become an important technology in search engines.

Although personalized search is receiving increasing attention and interest from researchers, accessing data sets for personalized search is not easy for many of them. Moreover, despite a large volume of data sets available in the field of information retrieval, most of them are not suitable for the study of personalized search. The reasons for this include: (1) the data sets from commercial search engines are not public; (2) the data sets lack such necessary information as that of long-term click behaviors of users, unified identifier for users or raw text of queries and documents; (3) the data sets have not been processed in a uniform way.

The first case in point is LETOR [1], a benchmark data collection for information retrieval, which lacks the time information of user historical behaviors. The second is SogouQ [2], query logs for short, from Sogou search engine, which lacks unified identifiers for users. Some available public data sets, e.g., Yandex[①] and SEARCH17 [3], have no raw text of queries or documents. For another famous data collection, AOL query logs [4, 5], there are no publicly available personalized search data sets.

In this paper, based on the work of [6], we proposed a complete and detailed data set construction process to construct a data set named AOL4PS in the field of personalized search from AOL query logs. When generating candidate documents for queries, we proposed an improved BM25 algorithm to improve computing efficiency and reduce storage space in the process. In addition, the statistics of AOL4PS are provided and the experiments are conducted on the AOL4PS to test the validity.

The remainder of the paper is structured as follows. Section 1 introduces the meaning of personalized search and the lack of data sets in personalized search. Section 2 reviews the current status of data sets in personalized search. Section 3 describes the complete process of data set construction, and the content and statistics of the proposed data set named AOL4PS. Section 4 analyzes the distribution of AOL4PS and describes how experiments are conducted to test its validity. Finally, we conclude our work in Section 5.

[①] https://www.kaggle.com/c/yandex-personalized-web-search-challenge

2. RELATED WORK

Crowdsourced data sets. Anikó et al. [7] collected two real-world data sets by posting two tasks on Amazon's Mechanical Turk. In total, they recruited 300 AMT workers, 200 for the Google Search experiment and 100 for the Bing experiment. Kumar and Sharan [8] used the browsing history of 10 different users for 50 queries. In this way, they simulated the search scenarios of several users. Generally, the query logs constructed via crowdsourcing suffer from a small number of users and queries and long time-consuming for personalized search where the number of users and queries play critical roles.

Nonpublic data sets. The data set from MSN query logs [9, 10], anonymized logs of the Microsoft Bing search engine [11, 12, 13] and the query logs obtained from Yahoo search engine [14] are based on commercial search engine data, which are either unpublished or no longer available.

Data sets without raw text. In 2013, Yandex released a large-scale anonymized data set for "Yandex Personalized Web Search Challenge". Similar with Yandex, Nguyen et al. [3] exposed a data set named SEARCH17 in 2019. Yandex data set consists of information on anonymized user identifiers, queries, query terms, URLs, URL domains, and clicks. Although Yandex provides a large-scale data set, its anonymous identifier processing of queries and URLs prevents researchers from accessing the raw text. Because of the lack of raw text of queries and documents, machine learning models which are not based on text information are widely used on the Yandex data set [15, 16]. However, the Yandex data set is not suitable in the era of deep learning, where text rich in semantic information is significantly needed to enhance the performance of various tasks in natural language processing through representation learning.

AOL query logs. In 2006, American Online[2] (AOL) released the AOL query logs which are suitable for information retrieval, query recommendation, and personalized search. One of the most important advantages of AOL query logs is that they contain the original corpus of queries and documents. Compared with Yandex and SEARCH17 both without raw text, the AOL query logs are more suitable for deep learning based-methods. Carman et al. [17, 18] improved topic models to handle the personalized search task. They did not re-rank the list of candidate documents but generated a new ranked list for users. Tyler et al. [19] proposed a re-ranking method which utilizes re-finding behaviors recognition for personalized information retrieval. Each query from the AOL query logs is resubmitted to the AOL search engine and the returned documents are crawled to generate candidate documents. Ahmad et al. [6] used the AOL query logs for document ranking and query suggestion. To address the problem that AOL query logs do not contain candidate documents from the search engine, they crawled the titles of documents and generated candidate documents using the BM25 algorithm.

In previous studies, the processing of AOL query logs is not sufficiently clear for the construction of personalized search data sets. The existing personalized search data sets based on AOL query logs in different scales are almost publicly unavailable [4, 5]. It is because there is no complete and clear data set

[2] https://www.aol.com

construction process and no publicly available personalized search data set, and there is a gap to fill in the research field of personalized search. Therefore, a unified approach to the construction of personalized search data set based on AOL query logs is necessary and significant. This paper presents the processing of AOL query logs to construct a personalized search data set which can be available to the public and test its validity on several classic and state-of-the-art personalized search models.

3. DATA SET CONSTRUCTION

3.1 Content of AOL Query Logs

In 2006, AOL Search released a collection of user query logs that include 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.

In particular, AnonID is an anonymous user ID for privacy protection. For example, the following is an entry taken from the AOL query logs: "142 westchester.gov 2006-03-20 03:55:57 1 http://www.westchestergov.com", which denotes the user with ID 142 submitted a query of "westchester.gov" on 2006-03-20 and clicked on http://www. westchestergov.com, which was the result ranked #1.

Table 1. Field and corresponding description of AOL query logs.

Field	Description
AnonID	An anonymous user ID number
Query	Search word
QueryTime	The query issued by the user, case shifted with most punctuations removed
ItemRank	The time at which the query was submitted for search
ClickURL	If the user clicked on a search result, the rank of the items on which he or she clicked is listed.
	If the user clicked on a search result, the domain portion of the URL in the clicked result is listed.

The statistical results of AOL query logs are shown in Table 2.

Table 2. Statistics of AOL query logs.

Item	Statistics
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 only contain the records of user clicked documents for the queries, and do not have the records of the candidate documents returned by the search engine. As a result, we generated the list of candidate documents for each query and annotated the documents that users are satisfied with. Since the AOL query logs are very large in scale and thus consume much time and space in the process of data set

construction, in the case of limited hardware resources, it is very unlikely to meet the time requirements of computation through traditional data processing. Therefore, we proposed an improved BM25 algorithm to solve this bottleneck problem of time consumption in data processing.

3.2.1 Data Preprocessing

Data preprocessing is to remove the wrong data and redundant data. Although the logs are structured, the log file is not a database, and the logs were not checked for completeness when they were generated, so there is no guarantee that the logs generated by the server is correct and complete. In addition, data redundancy is caused when log files are merged. A commercial search engine usually has many servers to handle a large number of query requests and the same record may appear twice or more times in the final merged query logs.

3.2.2 Data Crawling

We crawled the text of documents since the AOL query logs contain only the URLs of documents but not the text content. The purposes of crawling the text content are two-folded: first, in the process of data set construction, the relevance scores are calculated based on the matching degree of the text between queries and documents; second, in personalized search models, text content of both queries and documents is an important kind of feature.

Following [6], we crawled the title of Web pages that correspond to documents. For example, the corresponding title of "www.orlandosentinel.com" is "Orlando news, weather, sports, business | Orlando Sentinel". Crawling titles of the documents has two benefits: first, titles are more similar to queries than body texts [21]; second, using titles instead of body texts can significantly reduce the storage capacity.

The titles crawled from Web are divided into the following four categories by their text content (Table 3). The first category includes the titles that contain valid information; the second category contains those with no crawled text, being NAN or the white space character for example; the third category is composed of the titles with only invalid information, such as "404 Not Found", "403 Forbidden", and "502 Bad Gateway"; the fourth category comprises the titles in non-English characters such as Chinese, Japanese, and Russian.

Table 3. Category percentages and examples of the titles crawled by the crawler.

Category of the title texts	Percentage (%)	Examples
Valid text	53.92	New Cars, Used Cars For Sale, Car Prices & Reviews at Automotive.com
No text	34.08	NAN; white space
Invalid text	10.49	404 Not Found; 403 Forbidden; Access Denied
Non-English text	1.51	Tomaszów Mazowiecki, Łódź, mieszkania

Since the AOL query logs were released in 2006, many of the URLs no longer exist or the content has been updated. About 35.72% of the documents appear to have the same domain name with the title, e.g., "http://www.clanboyd.info" has the same domain name with the title "clanboyd.info". Considering that the textual content is missing in many of the documents, we take the domain names of the documents as their textual content.

3.2.3 Data Cleaning

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

Data Cleaning Process. The flow of data cleaning is divided into two stages: in the first stage, normalization, tokenization, and word segmentation are performed in sequence; in the second stage, according to the needs of different tasks, lemmatization or stemming is performed. For example, when calculating the similarity between queries and documents, stemming is performed to minimize the number of words and make it easier to match queries with documents. When calculating the representation of queries and documents, lemmatization is performed to reduce the memory and retain as much semantic information as possible.

Data Cleaning Effect. We use word coverage rate to evaluate the effect of data cleaning. We define the word coverage rate as Equation (1):

$$WordCoverageRate = \frac{n}{N} \tag{1}$$

where N represents the number of words in queries and documents of AOL query logs, and n represents the number of words in GloVe6b[3], an open source data set that includes 400,000 words in total. For the words extracted by stemming and lemmatization, the word coverage rate is 72.32% and 93.52%, respectively. Calculating the word coverage rate can be helpful to judge the effect of data cleaning. The rates over 70% prove that the data cleaning is effective. Stemming can reduce the number of words, but tends to generate tokens instead of words and might lose the original meaning of words. Lemmatization retains more different forms of words and thus has richer semantic information.

3.2.4 Document Similarity Calculation

Introduction to BM25. BM25 [22] is a classic information retrieval algorithm which is widely used by many mature search engines, such as Lucene[4] and Elasticsearch[5]. BM25 is proposed based on probabilistic retrieval model and used to evaluate the similarity between queries and documents with the relevance

[3] https://nlp.stanford.edu/projects/glove
[4] https://lucene.apache.org
[5] https://www.elastic.co

scores, which are obtained through calculation by measuring the matching degree of the words between queries and documents. We generated a series of candidate documents for queries by ranking the relevance scores. BM25 scores are calculated through Equation (2):

$$\mathrm{Score}(\mathrm{Q},\mathrm{d}) = \sum_{i}^{n} IDF(q_i) \times \frac{f_i \times (k+1)}{f_i + k \times \left(1 - b + b \times \frac{dl}{avgdl}\right)} \tag{2}$$

where Q stands for the query to be retrieved, the words of query Q are $\{q_1, q_2, \ldots\}$, n represents the number of words in query Q, d is a document that search engine returned, $f_i(q_i, \mathrm{d})$ represents the frequency of word q_i in d, dl is the length of d, and $avgdl$ is the average length of all documents. The function of parameters k and b is to adjust the effect of document length on the similarity between queries and documents, and k = 2 and b = 0.75. $IDF(q_i)$ stands for the inverse document frequency of q_i, calculated through Equation (3):

$$\mathrm{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \tag{3}$$

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

Improved BM25 Algorithm. BM25 is an efficient and stable algorithm in information retrieval. Following [6], we used BM25 to compute the relevance scores between queries and documents, and then we generated the candidate documents for all queries after ranking the relevance scores. Although many open source tools of natural language processing have provided interface of BM25, its calculation time does not meet the requirements for large-scale data. As shown in Table 2, AOL query logs have millions of queries and documents, and the magnitude of the calculation of the relevance scores is at the level of billions. In order to solve the problem of too long calculation time, we proposed an improved BM25 algorithm using matrix to improve the efficiency of computation. The improved BM25 algorithm accelerate the original algorithm by at least 6 times, where the relevance scores are calculated through Equation (4):

$$\mathbf{RevanceScores} = \mathbf{X}^{\mathrm{T}}\mathbf{IDF}\frac{(\mathrm{k}+1)\cdot \boldsymbol{F}}{\boldsymbol{F} + \mathrm{k}\cdot\left(1 - \mathrm{b} + \frac{\mathrm{b}}{avgdl}\cdot \mathbf{DL}\right)} \tag{4}$$

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

Block Matrix Multiplication. The improved BM25 algorithm we proposed above is a matrix implementation of the original BM25 algorithm. Although the use of matrices for computing can save time, the storage of matrices is a critical problem demanding solution. As shown in Table 2, the number of queries exceeds 3 million, the number of documents exceeds 1 million and the number of the words is about

80,000. Such large matrix requires hundreds of GB memory to store. The average length of queries is about 4-word-long and the length of documents is about 7-word-long, which indicate that the above matrices are very sparse and sparse matrices can be used to reduce the storage space. Only the matrices $\mathbf{X}^{\mathrm{T}}$, **IDF** and **F** are sparse matrices, but the relevance scores that are obtained by the improved BM25 algorithm is a dense matrix. The problem of insufficient memory still occurs in calculation. As a result, we used block matrix multiplication to reduce the memory usage. In addition, we sorted the relevance scores for extracting related documents, which was also optimized via matrix multiplication.

Extract Related Documents. After calculating the relevance scores by the improved BM25 algorithm, we extracted related documents from all documents for queries. Following [23], we treated the top 1,000 documents as relevant documents and the remaining documents as irrelevant. Under conditions when only relevant document of a query is clicked by users, the query is categorized as valid; otherwise the query is removed as invalid. This is mainly because we want to focus on the queries which have related documents. In this way, we deleted the queries that do not have related documents and generated related documents for the queries we saved. We put the clicked documents in the center of the window and selected the documents within the window as candidate documents. In order to reduce the data storage and speed up the training process, we set the window size to 10.

3.2.5 Data Annotation

SAT-click. A SAT-click is simply defined as a click whose dwelling time on the document extends over a predefined time threshold or the last click on a search session [24, 25, 26]. Usually the time threshold is set as 30s. The SAT-clicks of users on documents are used as an indicator of user true interests.

Session Partition. Session refers to a series of queries issued by the same user over a short period of time. It is believed that the queries submitted by the user within a period of time and query intention within a session are generally related. Following [27], we first used 30 minute as the time interval for session partition. We next used the cosine similarity of successive query to partition sessions. The queries are represented by the TF-IDF weighted sum of the embedding vectors of words in each query. Following [6], the threshold of cosine similarity for session partition is set as 0.5.

3.2.6 Data Division

We divided AOL query logs into four categories: historical data, training data, validation data and test data. The historical data are used to create the user profile. The training data, validation data and test data are used to train, validate and test the personalized search models. Based on [28], we divided the historical and other data according to the query time. For the data of 12 weeks, we chose the data of the first 9 weeks as historical data. The data in the last 3 weeks are divided into training data, validation data and test data with a ratio of 4:1:1. In addition, we filtered out the users who did not click in the first 9 weeks or who had less than 6 records in the last 3 weeks.

3.3 Contents and Statistics of AOL4PS

The contents of the processed data set named AOL4PS are shown in Table 4. The Statistics of AOL4PS are shown in Tables 5 and 6.

Table 4. Field and description of AOL4PS.

Field	Description
AnonID	An anonymous user ID number
Query	The query issued by the user
QueryTime	The time at which the query was submitted
ClickPos	The position in which the user clicks the candidate documents
ClickDoc	The document clicked by the user
CandidateList	A list of candidate documents returned by the search engine for the query
SessionNo	The session number of a user search action

Table 5. Basic statistics of AOL4PS.

Item	Value	Item	Value
Date range	2006/3/1-2006/5/31	#sessions/#users	73.88
#users	12,907	#SAT-clicks/#users	103.75
#queries	1,339,101	#queries/#sessions	1.40
#distinct queries	382,222	#SAT-clicks/#sessions	1.40
#distinct URLs	746,998	#SAT-clicks/#distinct queries	3.50
#sessions	953,592	#SAT-clicks/#distinct URLs	7.14
#SAT-clicks	1,339,101	average query length	4.05
#queries/#users	103.75	average document length	7.05

Table 6. Division of AOL4PS.

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

3.4 Comparison of Data Sets

Table 7 shows the comparison of AOL4PS with existing data sets Yandex and SEARCH17. The advantage of AOL4PS is that it has the original text information, such as the content of queries and the URL of documents, while there are no original text of queries and documents in Yandex and SEARCH17. In personalized search, a wider range of query time and more query records can help personalized search models to learn more accurate user interest features. At the same time, the moderate number of users and query records make the consumption of hardware resources and time resources in the process of model training more controllable. AOL4PS has advantages in personalized search due to its rich original content, a large number of users, long query time of users, and a large number of user query records.

Table 7. Comparison of AOL4PS with existing data sets.

Data set	#Days	#Users	#Queries	#Queries/#Users	Text Information
Yandex	30	5,736,333	64,693,054	11.3	No original text, only query IDs and document IDs
SEARCH17	15	106	8,016	75.6	No original text, only query IDs and document IDs
AOL4PS	92	12,907	1,339,101	103.7	Original text of query, and url of document

3.5 Usage of AOL4PS

As the current query moves backward in user query sequence, all queries that appear before the current query are regarded as the historical queries of the current query. Therefore, for a user query sequence, multiple samples can be constructed. As shown in Figure 1, it is assumed that a user has 15 query records, and the number of historical queries is 9, the number of training queries is 4, and the number of validation queries and test queries is 1. Therefore, 4 training samples, 1 validation sample and 1 test sample can be constructed. In each sample, as the current query moves back in the user query sequence, the number of historical queries used to generate the user interest increases. In AOL4PS, each user query sequence can satisfy the construction of training, validation and test samples.

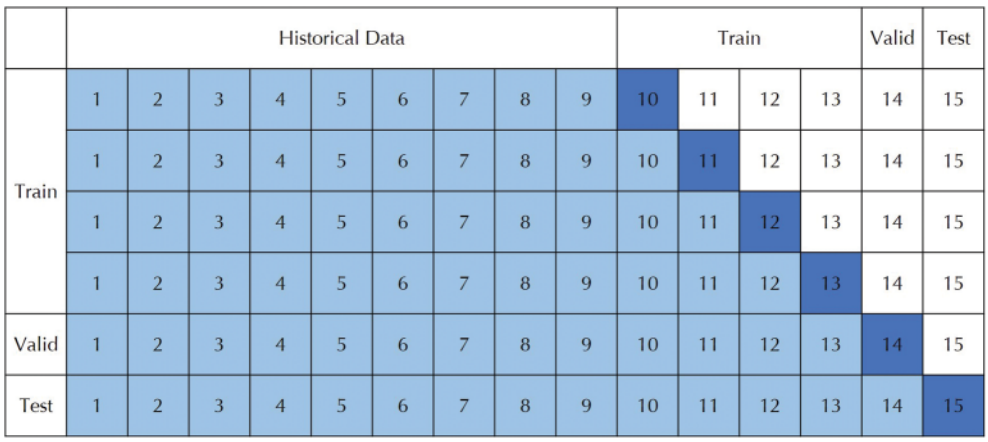


Figure 1. Sample construction description.

4. DATA SET ANALYSIS AND EXPERIMENTS

4.1 Distribution of Queries

In AOL4PS, more than 60% of the distinct queries are issued only once in the 3-month period and about 87% of the distinct queries are single-user queries. The statistics are similar with those given in [9] where the commercial query logs are used. About 46% of the queries in the test set appear in the training set. Furthermore, 35.75% of the repeated queries in the test set are repeatedly submitted by the same user.

4.2 Distribution of Query Click Entropy

In this paper, we utilized click entropy of query [29] to measure the ambiguity of query and the degree which the query needs to be optimized in personalization, as given in Equation (5):

$$\text{ClickEntropy}(q) = \sum_{p \in P(d)} -P(p|d) log_2 P(p|d) \tag{5}$$

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

$$P(p|d) = \frac{|Clicks(q,d,\cdot)|}{|Clicks(q,\cdot,\cdot)|} \tag{6}$$

where $|Clicks(q,d,\cdot)|$ represents the number of clicks on document d of query q and $|Clicks(q,\cdot,\cdot)|$ represents the total number of clicks on query q.

As shown in Figure 2, we calculated click entropy of the queries that are asked by more than one person. The higher click entropy indicates that the more queries need to be optimized by personalized search in AOL4PS. It can be seen that the click entropy of more than 30% of the queries is larger than 0. Therefore, it is obvious that AOL4PS requires a personalized search optimization.

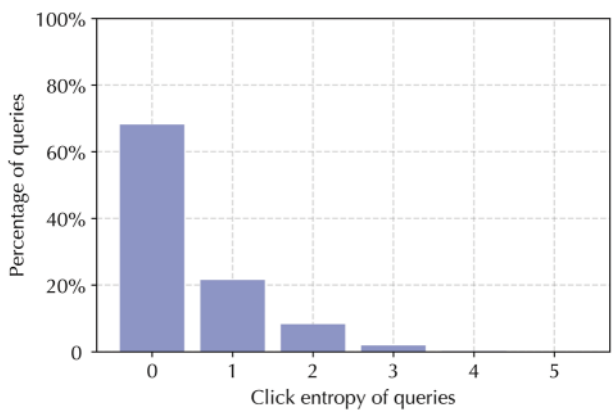


Figure 2. Distribution of query click entropy.

4.3 Distribution of Sessions

Figure 3 shows the distribution of the number of queries per session. More than 25% of the sessions contain at least two queries. This indicates that users sometimes submit several queries to fulfill an information need. This observation is consistent with that given in [9].

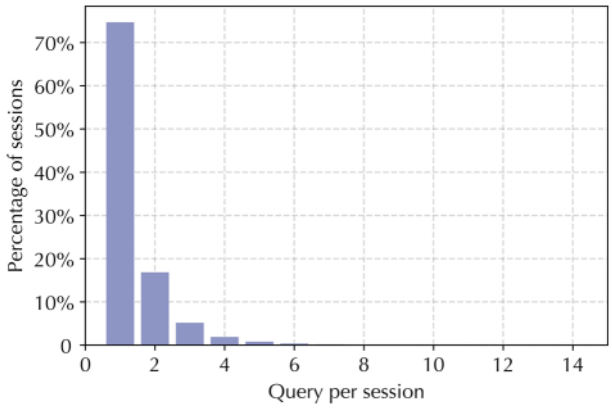
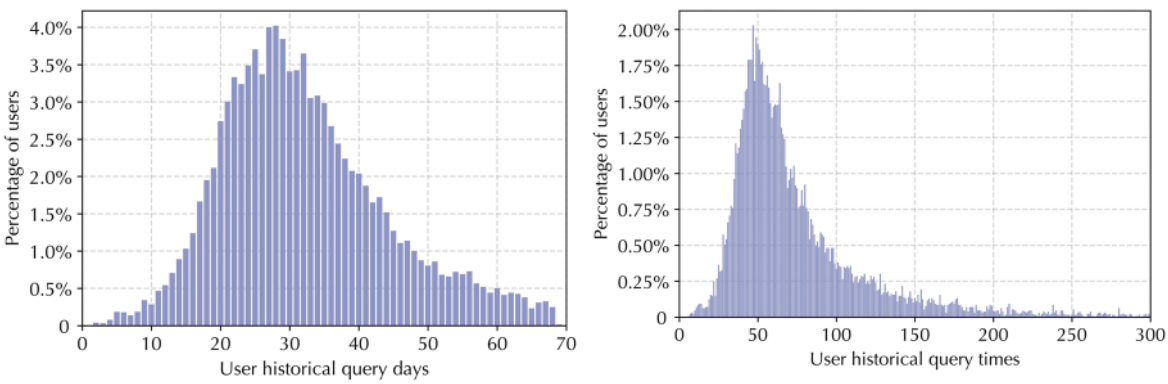


Figure 3. Distribution of query number of sessions.

4.4 Distribution of Users

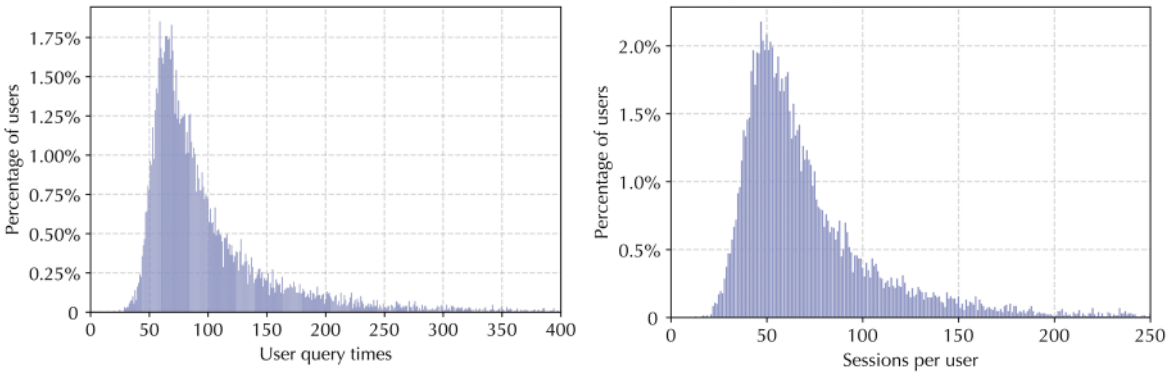
Figure 4(a) shows the distribution of historical days and Figure 4(b) shows the query times for users in search history. We found that the history of 68% of the users in test set over 25 days and about 68% of the users submit more than 50 queries during the historical period of users. Figure 5(a) shows the distribution of user query times in the 3-month period. More than 94% of the users sent at least 50 queries. Figure 5(b) shows the distribution of session numbers of users in the 3-month period. More than 70% of the users have at least 50 sessions. This indicates that AOL4PS which is built based on the long-term historical behaviors can provide sufficient data to learn the 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 users queries.

4.5 Evaluation Measures

We measured the personalized search accuracy by mean reciprocal rank, P@K and average click position.

4.5.1 MRR

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

$$\mathrm{MRR} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{7}$$

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 evaluation metric used in this paper is P@K, which measures the accuracy of the top K documents for a given query. As we all know, users are usually interested in the first few documents in the list of candidate documents. The computation formula is as Equation (8):

$$\mathrm{P@K} = \frac{n}{N} \tag{8}$$

where n represents that there are n satisfied documents whose rank in candidate documents are in the first K, N is the number of documents, and in this paper we set K as 1.

4.5.3 Avg.Click

We further use the average click position (Avg.Click) [30] of the SAT-click to evaluate the effect of re-ranking. Lower Avg.Click represents better personalized search ranking, which is defined as Equation (9):

$$\text{Avg.Click} = \frac{1}{N}\sum_{i=1}^{N} ClickPosition_i \tag{9}$$

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 application of learning to rank in information retrieval. Learning to rank methods can be generally classified into three categories, which are pointwise methods, pairwise methods, and listwise methods [31]. These methods transform the ranking task into regression task, classification task and so on. In AOL4PS, we labeled whether the user clicked on a document and the rank of the clicked document in the list of candidate documents. In personalized search, the ground truth is the relative ranks of two documents. In our experiments, we employed five classic or state-of-the-art personalized search models to test the effectiveness of AOL4PS. The following five methods are statistics-based methods (BM25, P-Click), listwise method (SLTB), pairwise methods (HRNN, GRADP), respectively.

BM25. As we mentioned before, BM25 proposed by Robertson et al. [22] is a document ranking algorithm without personalization, which determines the relevance between queries and documents by the matching degree of text. The original ranking of all candidate documents in AOL4PS is derived from BM25. We set adjustment factor k as 1.5, b as 0.75.

P-Click. Dou et al. [9] proposed the P-Click, which re-ranks the documents based on the number of clicks which a user makes under the same query. P-Click is bias to the repeated queries. We set the smoothing parameter β in the score calculation function as 0.5.

SLTB. Bennett et al. [11] analyzed user interests by extracting more than 100 features from the short-term and long-term historical behaviors. All features are used to train a LambdaMart [32] model to generate a personalized ranking list. SLTB is one of the best machine learning-based models in personalized search. The following parameters are used to implement the model: number of leaves = 70, minimum instances in a leaf node = 2000, learning rate = 0.3, and number of trees = 50.

HRNN. Ge et al. [28] used the hierarchical recursive neural network with query-aware attention to build the short-term and long-term user profile according to the current query dynamically. Then, documents are re-ranked based on the user profile and other relevant features. The following parameters are used to implement the model: word embedding size = 200, size of short-term interest vector = 200, size of long-term interest vector = 600, number of hidden units in attention MLP = 512, and learning rates = $1e^{-3}$.

GRADP. Zhou et al. [33] used the recurrent neural network and attention mechanism to capture the dynamicity and randomness of user interests. To build effective user profile, they extracted several query features including click entropy, topic entropy and so on. The following parameters are used to implement the model: word embedding size = 200, hidden size of GRU = 600, number of hidden units in attention MLP = 512, n in nCS = 2, n in nRS = 3 and learning rates = $1e^{-3}$.

4.7 Experiment Results and Analysis

We split the whole AOL4PS into four sub data sets according to the number of query records of users and each sub data set contains over 3,000 users. Besides, to reduce the storage and time in experiments, we chose only 5 candidate documents from the whole 10 candidate documents. We evaluated the performances of different models on the four sub data sets of AOL4PS and the results are shown in Table 8.

Table 8. Performance of different personalized search models.

Data set	Data1			Data2			Data3			Data4		
Method	MRR	P@1	Avg. Click	MRR	P@1	Avg. Click	MRR	P@1	Avg. Click	MRR	P@1	Avg. Click
BM25	0.4633	0.1792	2.6015	0.4845	0.2098	2.5332	0.4809	0.2056	2.5469	0.4786	0.1996	2.5540
P-Click	0.6554	0.4734	2.0295	0.7212	0.5718	1.8270	0.7437	0.6075	1.7647	0.8246	0.7297	1.5193
SLTB	**0.8427**	**0.7255**	**1.4272**	0.8288	0.6998	**1.4621**	0.8362	0.7116	**1.4387**	0.8306	0.7068	1.4813
HRNN	0.8096	0.7126	1.6822	0.8280	0.7424	1.6219	0.8472	0.7693	1.5458	**0.8926**	**0.8369**	**1.3811**
GRADP	0.8077	0.7099	1.6874	**0.8407**	**0.7596**	1.5674	**0.8556**	**0.7811**	1.5118	0.8894	0.8318	1.3918

From Table 8, we observed that all personalized models (P-Click, SLTB, HRNN, and GRADP) perform well on the four sub data sets. Specifically, all personalized models significantly outperform non-personalized model (BM25) on MRR, P@1 and Avg.Click, showing the effect of personalization. SLTB outperforms P-Click, which means that the complex machine learning models are promising. P-Click is designed with only one feature, but SLTB is designed with more than 100 features. Machine learning models rely heavily on the quality and quantity of artificial designed features.

HRNN and GRADP outperform traditional machine learning models including P-Click and SLTB especially on data3 and data4, which are similar to how these models perform on another personalized search data set [28]. HRNN and GRADP are similar in structure, both of them are based on recurrent neural network and attention mechanism. HRNN is proposed to model the impact of short- and long-term historical behaviors on the current query. GRADP is proposed to solve the problem of dynamicity and randomness of queries. HRNN and GRADP can learn accurate user interest features from long-term historical behaviors, which shows the superiority of deep learning model in personalized search. Deep learning models rely on the modeling of user query sequences, while traditional machine learning models are heavily influenced by artificial designed features. From the experimental results, it can be seen that deep learning models have more advantages in the case of long query records, and have become the mainstream models of personalized search.

Figure 6 shows the MRR statistics of different models on AOL4PS. The results of the BM25 algorithm are consistent on the four sub data sets of AOL4PS. SLTB steadily improves the original ranking but does not show an incremental trend on the four sub data sets. The results of P-Click, HRNN and GRADP on the four sub data sets are incremental, because the total number of user queries increases sequentially across the four sub data sets, and these models can capture user interests accurately based on more historical data. Therefore, for personalized search methods based on deep learning, more data help to obtain more accurate information of user interests. AOL4PS has a large number of users and query records, which can effectively test the effect of personalized search models.

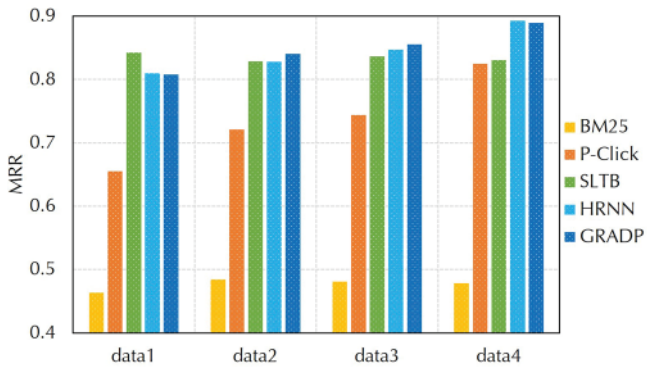


Figure 6. The results of different sub data sets.

In summary, AOL4PS is a large data set for personalized search. First, the data set has more than 10,000 users and more than 1 million query records. Second, the history of user is long enough which can support personalized search models to learn accurate user interests. In conclusion, AOL4PS can effectively test the effectiveness of personalized search models.

5. CONCLUSION AND FUTURE WORK

This paper presents a solution to the lack of public data sets in the research field of personalized search. We proposed a complete and detailed data processing process based on AOL query logs. We constructed a large-scale data set with high quality, AOL4PS, for the personalized search. AOL4PS contains the data of large amount of users with long-term historical behaviors. In addition, we examined the performance of several typical personalized search models on AOL4PS, demonstrating the applicability and superiority of AOL4PS for personalized search task.

However, there are still many areas for improvement in our future work. First, when crawling document content, many documents were not crawled because their content had been updated or were not available, so we removed such documents from AOL4PS. In the future, we can get document content through historical website information stored at the Internet Archive or from existing information retrieval data sets. Second, we constructed a list of candidate documents for each user click. In a real retrieval scenario, a

user might have multiple clicks on the same query. In the future, we can use methods such as merge queries to generate the list of candidate documents with multiple clicks.

ACKNOWLEDGEMENTS

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

AUTHOR CONTRIBUTIONS

This work was a result of collaboration among all of the authors. H. Wan (hywan@bjtu.edu.cn, corresponding author) led the whole work and organized the content of this paper. Q. Guo (qianguo@bjtu.edu.cn) collected and constructed the data set, ran the experimental results, and wrote the paper. W. Chen (w_chen@bjtu.edu.cn) performed the data statistics and experimental analysis. All the authors have made meaningful and valuable contributions by revising and proofreading the resulting manuscript.

DATA AVAILABILITY STATEMENT

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

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 Websearch. Computing Research Repository abs/1706.05011 (2017)
[8] Kumar, R., Sharan, A.: Personalized Websearch 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 Websearch. 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 Websearch. In: Proceedings of the Fifth International Conference on WebSearch and WebData 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 Websearch. 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 Websearch 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 Websearch. ACM Transactions on Information Systems 23, 147–168 (2005)
[26] Gao, J., et al.: Smoothing clickthrough data for Websearch 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 Webapplications. 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 WebSearch and WebData 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 working towards 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 in 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 current research interests include spatial-temporal data mining, social network mining, and information extraction.