A Personalized Search Query Generating Method for Safety-Enhanced Vehicle-to-People Networks

Xiaodan Yan, *Member, IEEE*, Jiwei Zhang, *Member, IEEE*, Haroon Elahi, Meiyi Jiang, and Hui Gao

***Abstract*—Distracted driving due to smartphone use is one of the key reasons for road accidents. However, the 6G super-heterogeneous network systems and highly differentiated application scenarios require highly elastic and endogenous information services involving the use of smart apps, and related information retrieval by drivers in modern Vehicle-to-People (V2P) Networks. The tension raised due to the conflicting attention requirements of driving and information retrieval can be resolved by designing information retrieval solutions that demand minimal user interaction. In this paper, we construct a Personalized Search Query Generator (PSQG) to reduce driver-mobile interaction during information retrieval in the 6 G era. This system has a query generator and a query recommendation component that update two sets of relationships dynamically: one is the query and the title, another is search and recommendation. The proposed system learns a user's intent based on historical query records and recommends personalized queries, thus reducing the driver-mobile interaction time. We deploy the system into a real search engine and conduct several online experiments. These experiments are conducted using a custom constructed dataset comprising ten million samples. We use the BLEU-score metric and perform A/B testing. The results demonstrate that our system can assist users in making precise queries efficiently. The proposed system can improve drivers' safety if used in smartphones and other information retrieval systems in vehicles.**

***Index Terms*—Vehicular networks, search query, personalization, recommendation, safety.**

I. Introduction

According to the World Health Organization (WHO), yearly, approximately 1.3 million people die in road accidents [1]. WHO identifies distracted driving due to smartphone use as one of the key contributing factors for road accidents. Activities like phone conversations and texting are known to reduce drivers' situation awareness and increase their reaction times towards unforeseen events [2]. Research finds positive associations between road accident rates and the number of calls, texts, and Internet connections [3]. Simultaneously, more and more smart technologies are being integrated into modern vehicles in the era of 6 G. The use of smart apps and related information retrieval by drivers is going to be inevitable in 6G-supported vehicular systems.

Whereas different multi-modal techniques based on V2P [4] can be used for in-car information retrieval, users mostly query information through active search and use the search engine's text search function [5]. This is a bi-directional interaction, where users [6] input queries and search engines present results accordingly. This interaction becomes complex in cases when users do not know the query requirements. For example, if users want to travel by themselves, there may be unclear search needs, leading to users not knowing what information [7] to search for better. Such interaction could involve attention requirements same as the simple or complex texting, also bi-directional interactions. Simultaneously, it can affect search engine results and result in prolonged interactions that could be dangerous in the case of vehicle drivers [2], [3], [8].

In the 6G era, people's lives are facing more highly differentiated application scenarios [9], [10]. In order to solve the tension caused by the distraction needs of driving and information retrieval, we can design a search solution that needs the least user interaction. For example, complexity due to a lack of understanding of clear query requirements can be eliminated by predicting user intent. Most search engines, such as Google, Baidu, and Bing, use the query-suggestion method to detect users' intention based on their input before information retrieval. This process involves addressing other related issues like correcting spelling mistakes, learning text semantics, explaining abbreviations, dealing with specialized terms, and other errors [11]. Also, query expansion (QE) is proposed to improve the search efficiency of Internet information, which can be applied to various information retrieval fields [12]. Query auto-completion is another essential feature of the search engine that makes query submission efficient [13].

The personalized recommendation suggests content to users that may match their interests. By suggesting content to users, recommender systems can help them with decision-making while interacting with different content repositories on the Internet [14]. These recommendations are usually personalized, and different suggestions are made to different users. For example, in social networks, comments, tags, attributes, and other content information [15] can be related to user interests [16]. Content

Manuscript received August 12, 2020; revised January 14, 2021 and March 19, 2021; accepted April 5, 2021. Date of publication April 27, 2021; date of current version July 8, 2021. This work was supported in part by the Action Plan Project for Basic Scientifific Research Business Fee of Central University to Improve Scientifific and Technological Innovation Ability (No. 2019XD-A12) and the Beijing Municipal Natural Science Foundation-Haidian Original Innovation Joint Fund (No. L182034). The review of this article was coordinated by the Guest Editors of the Special Section on Vehicular Networks in the Era of 6G: End-Edge-Cloud Orchestrated Intelligence. *(Corresponding author: Jiwei Zhang.)*

Xiaodan Yan is with the Beihang University, Beijing 100191, China (e-mail: xdyan17@163.com).

Jiwei Zhang, Meiyi Jiang, and Hui Gao are with the Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: jwzhang666@bupt.edu.cn; kingnothing@bupt.edu.cn; gaohui786@bupt.edu.cn).

Haroon Elahi is with the Guangzhou University, Guangzhou 510006, China (e-mail: haroon.elahi@e.gzhu.edu.cn).

Digital Object Identifier 10.1109/TVT.2021.3075626

can be assigned different relevance and interest values, and the interesting content can be sorted or filtered using different techniques, and then recommended to users. However, the increasing number and variety of data [17], [18] make it difficult for users to make the right choice. In this regard, association rule mining can help find related items and shorten the time to find the required items [19].

The integration of search engines and personalized recommendation is not only important from a user-experience point of view, but it has essential commercial value [20]. Although there are many differences between search and recommendation technologies, they are application branches of big data technology [21], [22], and there are many overlaps. Consequently, in recent years, search engines have gradually integrated features, such as displaying the relevant recommendations and search terms [23]. For example, in the platform type e-commerce websites, due to a large number of results and no apparent difference in correlation, the personalized ranking of search results helps users. The personalization and recommendation technology [24] used here also helps to promote transactions. Besides, the click feedback algorithm is also widely used in both to improve the effect.

Although the existing methods may help people formulate queries to describe their intent, none of them are trying to unveil user interests, which is of paramount importance to users. Imagine the following scenario: a user uses the routing application in his car to search for the shortest path to a hospital. However, the user is not aware of the fact that the shortest path is undergoing repairs. Since conventional route planning may involve factors like the distance and road blockages due to traffic congestion but not the repairs, it is unlikely for him/her to conduct related searches. In such scenarios, interest discovery intuitively facilitates the users in retrieving new or niche information [25].

This research aims to propose a solution that considers users' interests and predicts their intent for generating query suggestions for use in 6G-supported information retrieval solutions in vehicles. In this paper, we propose the PSQG mechanism for generating personalized search queries based on the users' historical searches in vehicles for reducing driver-mobile interaction. In this system, the search engine uses information that motivated a user to search in the first place (user's interest), predicts his/her intent, and generates personalized query recommendations, accordingly. The contributions of this work can be summarized as follows:

- We propose a novel mechanism called PSQG that suggests the query by predicting user intent and interests. The proposed model combines the extraction, generation, and recall modules to generate the query for each article. We consider information including the article's profile, user's profile, query features, and user's search history to suggest the query to the user.
- We construct a history query corpus, which contains 10 million title query pairs. The experimental results on the dataset demonstrate that our proposed framework can generate high-quality query suggestions to support personalized recommendations.
- To the best of our knowledge, this is the first work proposing the generation of personalized query suggestions for reducing information retrieval's attention requirements by generating personalized queries after mining users' potential intent.

The rest of the paper is organized as follows. We briefly review the related work in Section II. In Section III, we describe the PSQG framework. Section IV provides detailed information about personalized recommendation system. Experimental evaluations are given in Section V. Conclusions and future work are discussed in the last section.

II. Related Work

User interaction with search engines is a bi-directional, complex, and attention-seeking phenomenon. However, in vehicles, particularly while driving fragmented attention can affect both driving and searching. Previously, none of the studies has focused on improving search to reduce human interaction and attention requirements in vehicles. Yet, many 6G techniques have been proposed to improve human-search engine interaction. This section provides an overview of some prominent studies.

A. Query Auto-Completion

Many researchers have focused on Query Auto-Completion (QAC). QAC suggests queries that complete users' text as they enter each character to speed up their searches [26]. It is reported that more than 50% keystrokes of global Yahoo! Searchers had been saved in 2014 by using QAC to search the suggested query completions [27] when submitting queries.

There are two conventional QAC approaches, including query-log-based and document-based. These approaches use query log data. The use of log data for auto-completion is not new. Longuet-Higgins *et al.* [28] describe a method that helps reduce the number of keystrokes required to complete a word, in this case, identifiers and commands are entered by developers. Recently, query frequency [29], term co-occurrence, query chains, query click through [30] and hitting time have become the most commonly-used types of information [31] in the query log. Term co-occurrences often appear in the search and can be predicted by using the probability of co-occurrences in different feature spaces according to the clustering or dispersion trend between terms. According to the statistical method and spatial point pattern, the detection can be completed without using external data sources [32].

B. Query Correction and Expansion

In many fields, a considerable part of the information is presented in text, and the quality of data analysis [33] results largely depends on the quality of input data. Therefore, query spelling correction is of great significance [34]. The characteristics of different texts can be used to develop the spelling correction module to deal with the problems more accurately.

Query expansion means adding terms to expand the given query, and then better information retrieval (IR) performance comes along [35]. Queries often contain polysemy words. We can read the semantic relationship between polysemous words, and score the semantic relevance significantly, and distinguish them according to different scores. There are also spelling errors in search engine queries. Therefore, the research of query completion to correct spelling errors is of great value.

Online spelling correction provides correct spelling suggestions incrementally at the runtime when a query is entered. Generative modules based on noisy channel transformation and unsupervised learning have been used for this purpose [36].

C. Personalization and Recommendation

The personalized recommendation presents information and products of interest to users based on their interests and purchasing behavior [37]. When users query on the Internet, there are often a series of queries. Therefore, we can use the query chain to learn user preferences from the log of the search engine and then use this information to generate a query intelligently. Support vector machines have been used for comparing preferences and explicit association in this regard [38]. Another method based on document information includes generating or extracting related words or phrases by processing related documents containing queries and then using these words or phrases for query suggestions.

There are three distinctive methods to achieve personalized recommendations: collaborative filtering recommendation algorithm [14], content-based recommendation [19], and social network-based recommendation algorithm [16]. For the recommendation works, the cold start has been a key problem, which appears when new users or new projects are added to the system [39]. The cold start problems could be distinguished into three types: new user problems, new item problems, and new system problems where the new user is especially important [40]. In such cases, it is challenging to provide a recommendation. Collaborative filtering offers solution for this problem [16]. In case of query recommendation, the easiest solution is recommending popular queries to new users, and then switch to personalized recommendations when user data is collected [41].

A user content-based algorithm will be developed on the user's profile base (e.g. age, gender, region) to find similar users and recommend the query they interacted positively. Of course, all the information is gained during the registration process, including the instantly-input data and already available data e.g. from his other social media accounts [42]. Ghosting is the process of autocompleting a search recommendation. One of the recent works proposes a behavior-based ghosting model for retail markets that considers a user's context to ghost on high-confidence queries [43].

The recommendation systems also make use of search engine technology. For example, an inverted index is an important data structure used by the search engine to solve computing performance. The content-based recommendation also makes frequent use of the inverted index, query, result merging, and other methods [44].

D. Task Optimization

Delays in search can negatively affect user experience. Such delays can also mean a prolonged user engagement. One of the recent works proposes a framework for generating query suggestions to support users in completing their basic tasks, thereby reducing the total number of such tasks performed by users [45].

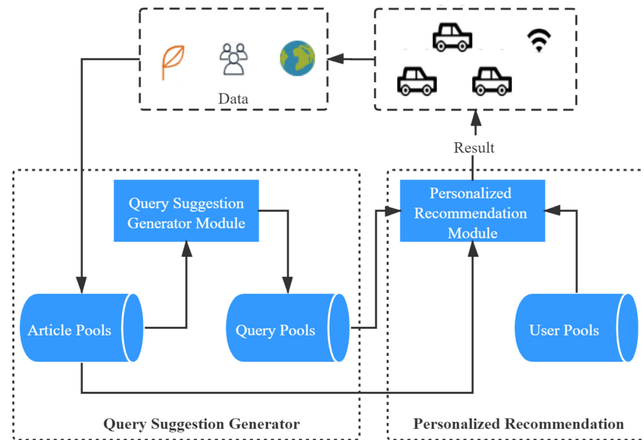

Fig. 1. An overview of query suggestions generator to support personalized recommendations.

Contrary to the previous studies reviewed in this section, we address the problem of reducing user interaction while retrieving information in the vehicles. We propose a framework for actively recommending query suggestions for this purpose. The proposed framework can predict and generate target and personalized query suggestions based on the user's attributes and a document's profile based on the query log.

III. Query Suggestions Generator Unified Model

As shown in Figure 1, our query suggestions generator to support personalized recommendations consists of two modules, i.e., query suggestion generator and personalized recommendation. Each query candidate is represented by q_i which is produced by generation module. Given the information of the article (we use the title) X, the query suggestion generator module gets a series of query candidates $C = \{q_i\}$. The personalized recommendation module recommends query suggestions $Q = \{q_i\}$ to the user according to the user profile, article profile, and the query profile.

As shown in Figure 2, query suggestion generator unified model is a unified framework combining the generation, extraction, and recall to generate query candidates. The input is a title sequence $X = \{x_1, x_2, \ldots, x_n\}$ consisting of n-words, where x_i is the i_{th} word. The query candidate q_1 is produced by generation module. We operate this module with the framework of sequence-to-sequence. Then, we extract the importance term based on high-frequency search and the key part of speech terms from X to get the query candidates q_2 and q_3 (shown in the yellow box). Given the query suggestion $\{q_i | 1 \leq i \leq 3\}$, the recall module (shown in the green box) recalls the related query candidates $\{q_i | 4 \leq i \leq m\}$, where $8 < m < 20$.

As mentioned above, our target is to generate query suggestion candidates with high correlation and high fluency. To achieve this goal, we construct a unified evaluation module considering the

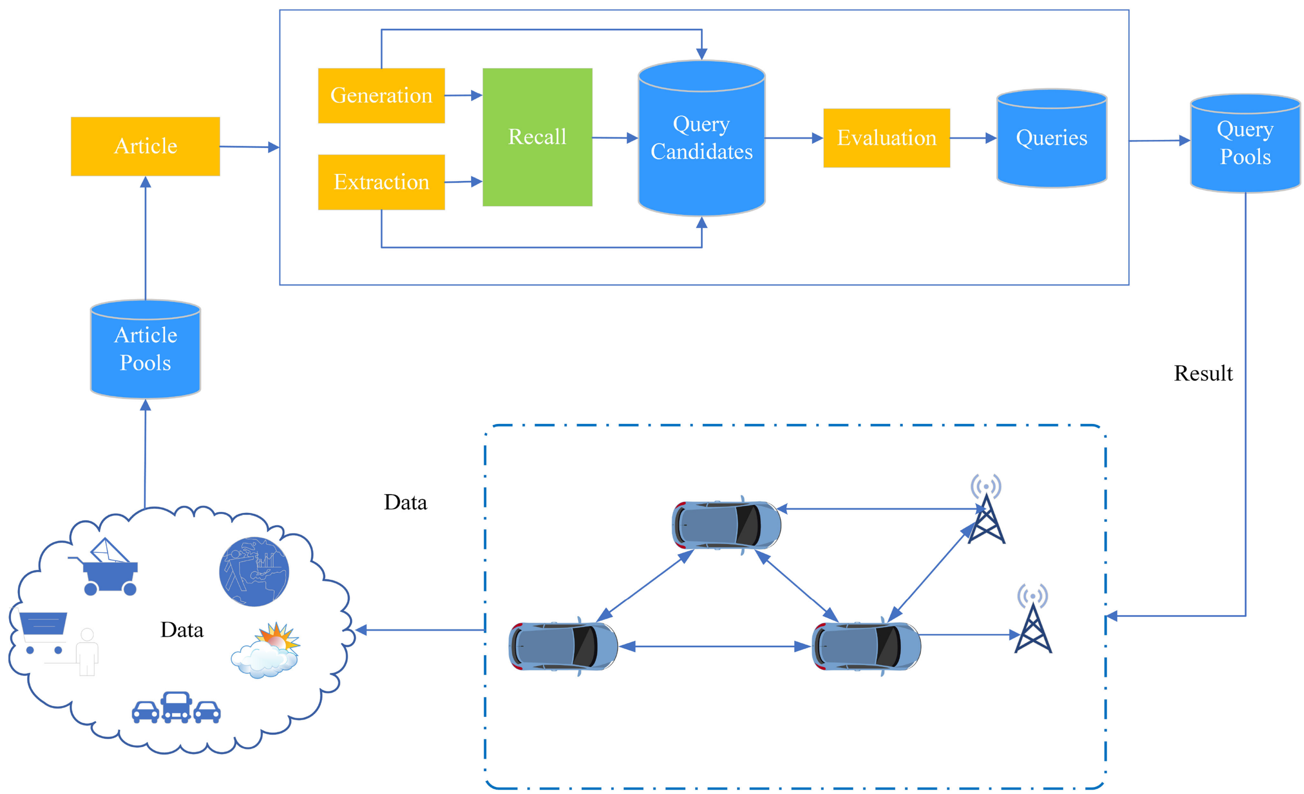

Fig. 2. Query suggestions generator unified model applied in a V2P network.

Algorithm 1: Query Candidate Based on R-LSTM.

Input: X, i, m, s_i, c_i.
Output: Y'.

1) **Initialization**: Initialize the module, initialize $X = x_1, x_2, \ldots, x_n$, s_i, and c_i.
2) **for** $i = 1$ to m **do**
3) Define $\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}$ where $e_{ij} = a(s_{ij}, h_i)$.
4) $c_i = \sum_{j=1}^{n} \alpha_{ij} h_j$.
5) $s_i = f(s_{i-1}, y'_{i-1}, c_i)$.
6) Obtain $G(y'_i|Y'_{1:i-1}, X) = g(y'_{i-1}, s_i, c_i)$.
7) Using $\overleftarrow{h^x_{i+1}}$=R-LSTM$(x_i, \overleftarrow{h^x_i})$.
8) **end**

---

relevance and fluency query to filter and sort the queries from $\{q_i|i \leq m\}$.

1) Generation Module: The goal of generation module, which is called G for short, is to produce a query Y'. And $Y'_{1:i}$ denotes the generated partial title at i_{th} step and $i \leq m'$. Then we use $y'_i = G(Y'_{1:i-1}|X_{1:n})$ to denote the ralationship between a given title X and a query Y'.

Seq2Seq is a neural network model for solving machine translation problems.The most basic Seq2Seq model contains three parts, namely encoder, decoder and intermediate state vector C connecting the two. Encoder is an Recurrent Neural Network (RNN)/Long Short-Term Memory (LSTM) model that converts input sentences into fixed-length vector expressions. Decoder is the reverse process of encoder, which outputs corresponding sequences through the learning of state vector C. Its input and output are a sequence. However, decoder can only convert sequences of any length to vectors of fixed length. This means that the longer the sequence, the more information will be lost.

The attention mechanism effectively solves the problem of using Seq2Seq model to deal with long sentences. This mechanism provides a way to keep in touch with distant words, which solves the problem of insufficient information saved by a vector. Therefore, we combine Seq2Seq model with the attention mechanism. First, we calculate the weights according to the feature of encoder and decoder. Then, the weighted sum of the encoder's features is calculated as the input of decoder. Not all contexts have an impact on the generation of the next state. The attention mechanism helps the Seq2Seq model to select the appropriate context and use it to generate the next state.

We apply the standard LSTM encoder-decoder on the base of attention (seq2seq-attention) in this task. We define a conditional probability for generation module as follow:

$$G(y'_i|Y'_{1:i-1}, X) = g(y'_{i-1}, s_i, c_i) \tag{1}$$

Where s_i denotes the hidden status unit in formula 1, and the context vector is indicated by the letter c. The term c_i is at step i. The hidden status s_i for standard LSTM decoder can be defined as formula 2. Where s_{i-1} denotes the previous step status, y'_{i-1} denotes the previous step output, and c_i denotes context vector:

$$s_i = f(s_{i-1}, y'_{i-1}, c_i) \tag{2}$$

All input information of X can be used to produce the i_{th} word of the query by generation module. Each word can be acquired by the decoder with a unique context vector. The c_i is defined as follows:

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j \tag{3}$$

The weight α is expressed as formula 4, where $e_{ij} = a(s_{ij}, h_i)$ is called the alignment module. We use e_{ij} to evaluate the matching degree between text (j_{th} word) and query (i_{th} word). The decoder with a unique context vector is a traditional attention mechanism.

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} \tag{4}$$

Some cases show that information word required by the query may be in any position of the title, which means the information word in the title may be from right to left or may be scattered in different positions. To this end, we introduce a bidirectional LSTM (Bi-LSTM) module for the encoder. Take the title X as an example. Its hidden state of the forward LSTM at timestamp i can be represented by:

$$\overleftarrow{h_i^x} = LSTM(x_i, \overleftarrow{h_{i-1}^x}) \tag{5}$$

The Bi-LSTM consists of a forward LSTM and a backward LSTM. Suppose the corresponding outputs are denoted as $[\overleftarrow{h_1^x}, \ldots, \overleftarrow{h_{-1}^x}]$ and $[\overleftarrow{h_{-1}^x}, \ldots, \overleftarrow{h_1^x}]$ respectively, where the index that -1 stands for the last element. Then, the composite hidden state of a word is the concatenation of the two LSTM representations, i.e., $h_i^x = [\overleftarrow{h_i^x}, \overrightarrow{h_i^x}]$. In our model construction, the Bi-LSTM is applied only on the first layer of the encoder, and the remaining layers are still unidirectional LSTM.

As mentioned above, the title is so short that a simple stacking of the model can result in slow training and is susceptible to gradient vanishing and gradient exploding. The problem of gradient disappearance and gradient explosion generally becomes more and more obvious as the number of network [46]–[48] layers increases. When we set up a 4-layer LSTM on the (vertical) single node, the model becomes un-robust. We introduce a simple but effective modification to Bi-LSTM in the encoder module, known as Residual LSTM (R-LSTM) [49]. In the temporal domain, R-LSTM provides a mechanism to avoid exploding gradients or vanishing gradients by using temporal shortcut path.

The R-LSTM target is to establish a difference between the middle and the target layer, combine the input of the previous layer with the current output as the input to the next layer. Then the i_{th} layer $LSTM_i$ interacts with the $i+1_{th}$ layer $LSTM_{i+1}$ as follows:

$$\overleftarrow{h_i^x} = LSTM(x_i, \overleftarrow{h_{i+1}^x}) \tag{6}$$

2) Extraction Module: In the extraction module, called E, we use two strategies based on document features and part-of-speech features. The document-based feature strategy refers to the selection of important terms from the title. The strategy based on the part-of-speech features refers to keyword selection from the title.

In the document-based feature strategy, we extract keywords according to term weighting. In addition to TF $\times$ IDF (Term Frequency-inverse Document Frequency), there are also some

Algorithm 2: Term Frequency-Inverse Document Frequency.

Input: X, i, m
Output: w
1) **Initialization**: Initialize the module, initialize $X = x_1, x_2, \ldots, x_m$, and $q_i = t_1, t_2, \ldots, t_j$.
2) Calculate term frequency of word t in query q_i $TF(t, q_i) = \frac{n_{t,i}}{\sum_{k=1}^{|T|} n_{k,i}}$.
3) Calculate inverse document frequency $IDF_t = log\frac{M}{m_t + 0.01}$.
4) $w(t, q_i) = TF(t, q_i) \times IDF_t$.
5) **end**

other main methods of keyword extraction, including NGL coefficient, mutual information, chi-square, and information gain [50]. TF $\times$ IDF and mutual information are conventional methods. Term frequency (TF) refers to the number of times a word appears in the file. However, some common words (eg. and) do not play an important role in determining the theme of the article. On the contrary, some words with less frequency can express the theme of the article. Therefore, it is not appropriate to use TF alone. The design of weight must meet the following requirements: the stronger the ability of a word to predict the theme, the greater the weight; otherwise, the smaller the weight. In all statistical articles, if some words only appear in a few articles, then such words have a great effect on the theme of the article. The weight of these words should be designed larger, and the role of IDF is just like this.

TF-IDF combines the advantages of TF and IDF to evaluate the importance of words to a document. The importance of a word increases in proportion to the number of times it appears in the document, but at the same time it decreases inversely with the frequency of its appearance in the corpus. Therefore, the more times a word appears in an article and the fewer times it appears in all documents, the more likely it is to represent the article and distinguish it from other articles.The conventional, TF $\times$ IDF weight scheme- developed by [51] is as follows:

$$TF(t, q_i) = \frac{n_{t,i}}{\sum_{k=1}^{|T|} n_{k,i}} \tag{7}$$

Here, $TF(t, q_i)$ denotes term-frequency of word t in a query q_i, $n_{t,i}$ denotes number of accurances of term t in q_i and $n_{k,i}$ means number of accurances of all terms in q_i. Where IDF is as follows:

$$IDF_t = log\frac{M}{m_t + 0.01} \tag{8}$$

M denotes the total number of queries in the corpus, and m_t denotes the total number of queries in the corpus where the term appears. So we calculate the weight as below.

$$w(t, q_i) = TF(t, q_i) \times IDF_t \tag{9}$$

We use the word segmentation system ICTCLAS to divide the massive users queries (100 million levels) into terms as $q_i =$

TABLE I
THE CHOSEN POS TAGS IN THE STANFORD CORENLP TOOLKIT

Tag	Comments
NR	Proper Noun
NN	Common Noun
VV	Verb
FW	Foreign Word
PN	Pronoun

$t_1, t_2, \ldots, t_j$. We then use Equation (7) to calculate each term weight, having (t_j, w_j). Meanwhile, we use the ICTCLAS to segment the title X, and we have $X = x_1, \ldots, x_m$. We can get the title's term weight based on query q_2. Although there are some differences between the terms of the query and the title, owing to a large amount of query data, the simple statistical scheme can effectively capture the characteristics of the keyword without considering this small error.

Compared with previous statistical NLP approaches, which rely on superficial statistical data such as word counts and co-occurrences, we introduce the Part-Of-Speech (POS). A large amount of data indicates that the query's key information is mainly contained in some terms. We compare the popular and open-source tools for POS in the Chinese language, Language Technology Platform (LTP) [52], The Stanford CoreNLP toolkit [53], ICTCLAS, and THULAC [54]. In our model, we chose The Stanford CoreNLP toolkit in consideration of many factors as the coverage of part of speech, processing efficiency, and accuracy, etc.

After experimentation, we chose the POS tags shown in Table I. Obviously, Table I is consist of tag and comments.

3) Recall Module: To make our query suggestions richer, we recall the similar user's queries semantically from search log. The existing query suggestions, generated by the generation and the extraction module, are the input to our recall module. Since the number of words in the query is very small, we calculate the similarity between the queries semantically.

We usually spend a lot of time looking for specific information in a large number of documents, and semantic search is one of the most important search methods. For the same word, if put in different contexts and times, it will have the different meanings. It is usually possible to capture something called "embeddings" in the context of a word, and two words with similar meanings will have similar vectors, allowing us to calculate the similarity of vectors. In vector space, we should be able to calculate the similarity between any two sentences. This is what the sentence embedding model can achieve. These models convert any given sentence into a vector, which can quickly calculate the similarity or difference of any pair of sentences.

We represent the query as a concatenation of terms $q_i = t_1, t_2, \ldots, t_m$, generally $m \leq 5$. Each term has and has only one-word embedding, which we represent as te. So our query embedding can be expressed as follows:

$$qe = \frac{1}{m}\sum_{i=0}^{m} te_i. \tag{10}$$

$$sim - score = \frac{\sum_{i=1}^{n}(qe'_i \times qe''_i)}{\sqrt{\sum_{i=1}^{n}(qe'_i)^2}\sqrt{\sum_{i=1}^{n}(qe''_i)^2}} \tag{11}$$

Here, qe' and qe'' denote the embedding of two queries, i denotes the i_{th} dimension, and n denotes the total dimension. Then the recall module recalls the related query candidates $q_i|4 \leq i \leq m$, where $8 < m < 20$.

4) A Unified Evaluation Function: For each title, we have the query candidates $q_i|i \leq m$. Unfortunately, we found that some queries are not complete in semantics or not related to the title. To this end, we filter and sort existing queries from the language fluency and semantic similarity. Firstly, we evaluate language fluency by the language model (LM). Generally speaking, LM is to judge whether a sentence is smooth from grammar. With the LM, queries that do not conform to the grammar rules or look like sentences can be eliminated directly, and the recognition errors will be greatly reduced. The LM is usually constructed as the probability distribution of strings, which actually reflects the probability of occurrence as a smooth sentence.

The goal of LM is to estimate the probability distribution of various linguistic units, e.g., words, sentences, etc. The LM probability $p(w_1, w_2, w_n)$ is a product of word probabilities based on a history of preceding words. Therefore, we can calculate the probability of a query as:

$$p(q) = p(t_1, t_2, \ldots, t_m) = p(w_1)\prod_{i=2}^{m}(t_i|t_1 t_2 \ldots t_{i-1}) \tag{12}$$

where t_i denotes i_{th} term in query. We set $p(q)$ as $score_1$. We calculate the $score_1$ of each query in the query candidates. If $score_1$ is less than the threshold A, we filter out this query.

On the other hand, we set the sim-score in 9 to $score_2$, which is used to measure the degree of the relation of the query to the title. If $score_2$ is less than the threshold B, we filter out this query too.

Finally, we synthesize $score_1$ and $score_2$ to sort the remaining query candidates, as shown in 13.

$$final - score = \lambda score_1 + (1 - \lambda) score_2 \tag{13}$$

IV. PERSONALIZED RECOMMENDATION

As shown in Figure 3, the personalized recommendation system consists of two modules, a preliminary screening module and a ranking module. For the primary screening module, we need to select user-related query suggestions from the query pools.

Owing to the massive queries, a simple Factorization Machine (FM) [55] is enough.The main goal of FM is to solve the problem of data feature combination in the case of sparse data. In the general linear model, each data feature is considered independently, and the relationship between data features is not considered. But in fact, a large number of features are related. Taking e-commerce as an example, female users tend to see more advertisements such as cosmetics and clothing, while men prefer various ball equipment. If we can find out these related features, it is obviously very meaningful. The module equation for an FM of degree $d = 2$ is defined as:

$$y = w_0 + \sum_{i=0}^{n} w_i x_i + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{ij} x_i x_j \tag{14}$$

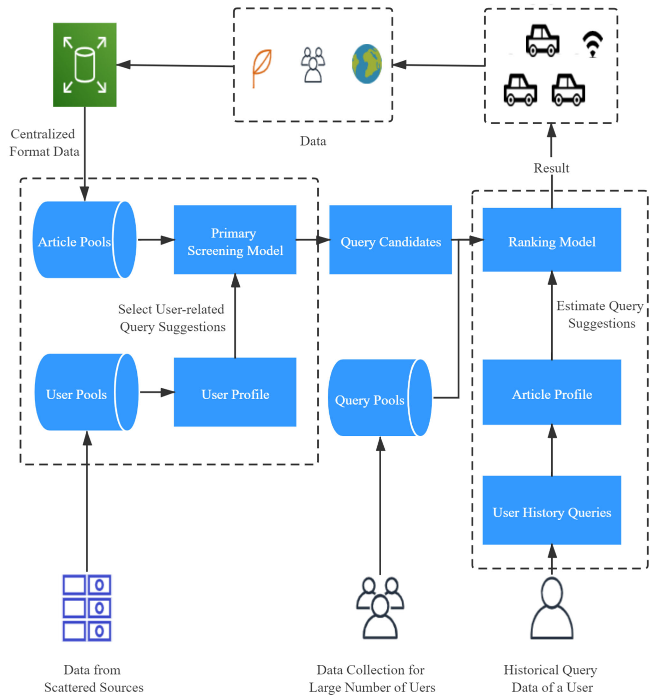

Fig. 3. The overall framework of personalized recommendation.

where n denotes the number of features of the sample, and x_i denotes the i_{th} feature. w_0 and w_i represent the weights of the network that weight is a concept in statistics. It mainly reflects the individual's position or proportion in the whole, and the specific value is the weight value.

The most critical task in the ranking module is to estimate the user's preference for query suggestions. In addition to the query suggestions for the primary screening module, we also introduce additional candidate sets, such as user historical queries. Now we treat it a classification problem: at time t, the user U (context information C) accurately predicts the category of a query in the query pools (each query has a category i), so we can fit it with a DNN model.

The personalized recommendation module's performance is mainly reflected in the article profile, user profile, and query embedding. In the article profile, we introduce 200-dimensional features such as titles, categories, keywords, and tags. The user profile will be more, including demographic information such as gender, age, region,etc., and other context information such as device, login status. Of course, the historical search query will also be introduced. We weight-average the embedding vector of the token after the query of the historical search, which reflects the user's overall search history. It is worth mentioning that even if a query is related, the user will not repeatedly click the same query. Therefore, we introduce a time penalty factor as one of the features.

For the cold start problem of new users, we will use some of the user's context information such as login authorized account, login location and find the closest user as the initial embedding of the new user instead of using random values. Of course, some high-frequency hot and non-personalized queries will be suggested more frequently among new users.

It is worth mentioning that since each of our articles corresponds to a large number of queries, we can update the query suggestions in real-time.

V. Experimental Evaluation

In this section, we first introduce the dataset. Then we present validation settings and use the bleu matrix in natural machine translation to analyze the query's information. We also perform an online A/B testing in a commercial search engine. Finally, we evaluate the fluency and the relevance of the human query to test and verify secondarily.

A. Datasets and Evaluation Metrics

To the best of our knowledge, there is no benchmark dataset for the PSQG task, and we construct a corpus. Each sample in our corpus is a pair (title, query) in which query comes from the user's real search query based on query-log. As a result, we collect ten million and one million queries paired with one title each non-repetitively. And another, we randomly split our corpus as a training set with ten million samples, a test set with 500 000 samples, and a development set with 500 000 samples.

We evaluate our generation modules using the common assessment metric in neural machine translation BLEU-score. BLEU-score is the metric to claim a high correlation with human judgments of quality. Besides, it uses a modified form of precision to compare a candidate translation against multiple reference translations.

Since personalized recommendations rely on user-profiles and article profiles, we deploy the model on modern search engines and use A/B testing to evaluate the effectiveness of our PSQG tasks.

B. Implementation Details

In the generation query suggestion module, the three modules' implementation details, and an evaluation function are as follows:

Generation: In the generation query suggestion, the structure of the generation module G is an RNN [56] based encoding-decoding framework with an attention mechanism. We use a four-layer LSTM setup, and the first layer is a bidirectional LSTM and residual structure. The LSTM hidden state and word embedded dimensions are set to 256. In the case of characters, the structure of encoder and decoder shares a vocabulary which is 50k-size. The initial learning rate is chosen from cross-validation on dev set (0.02), and we halve it every 80 k iterations.

The deep learning framework we adopted is TensorFlow 1.4. Two Tesla V100 GPU is used for training the model and performing experiments. In contrast to the comparative methods converged within 20 epochs, our models cost about 4 hours per epoch.

Extraction: For the extraction module, we use a document-based feature and a part-of-speech feature-based approach to achieve. Firstly, we drew massive queries from the search log and segmented them. Then, we calculate the term importance of this massive queries according to 9 mentioned in Section 3.1.2.

Finally, we segment the title by the same segmentation tool to get the score of term importance for each term. If the score of term importance is below the threshold (we set it to 0.52 in this paper), we filter out this term. If there are too many alternative terms (more than 5, because the query is limited to 10 words), we extract the five terms with the highest score of term importance. Of course, our extraction module guarantees that the order is the same as in the title.

On the other hand, we use the Stanford CoreNLP toolkit to perform part-of-speech tagging on the title, and extract the part-of-speech terms as NR, NN, VV, FW and PN terms.

Recall: Firstly, we use the Gensim tool 7 to train the massive queries mentioned in the extraction module to getting word embedding. Then, calculate the similarity between the query, including the generation module and the extraction module and the massive query. For the detail, see 10 mentioned in Section 3.1.3.

At this point, each title corresponds to some query suggestions. We will filter and sort these query suggestions. Firstly, we will use the KenLM tool 8 to train the massive queries to get the language module. Then, we use the language module to calculate the language-smooth $score_1$ of the query, and the word embedding module to calculate the similarity $score_2$ of each candidate query and title. If the score is $score_1 \leq 0.42$ or $score_2 \leq 0.5$, we move it out of the candidate pool and sort the existing query using the unified evaluation function mentioned in Section 3.1.4, where $\lambda = 0.3$.

In the personalized recommendation module, our word embedding, article profile embedding, and user profile embedding are both 256-dimensional. The article profile includes topics, keywords, professional thesaurus. And the user profile, including the user's operations in the app, click on videos, articles, browsing, favorites, etc. Then we use FM preliminary screening to get the query candidates according to the article profile embedding and user profile embedding. The size of the matrix is the number of articles multiplied by the users. The loss function uses the squared loss and the L_2 regular term. And the learning rate here is 0.01. And our ranking model is a typical DNN model.

C. *Evaluation Results of Generation Module*

To evaluate the performance of our generation module, we compare our model with the baselines and state-of-the-art methods: (i) Seq2Seq (word): is a baseline, seq2seq with attention mechanism [57], which is a word-based. (ii) Seq2Seq (char): is the same as the (i) in the model, different in character-based. (iii) Seq2Seq+Bi: Bi-LSTM is introduced on (ii); (iv) Seq2Seq+R: RLSTM is introduced on (ii). (v) Seq2Seq+Bi+R(512): Bi-LST and R-LSTM are both introduced on (ii), and The word embedding dimension is 512. (vi) Seq2Seq+Bi+R(256): is the same as the (v) in the model, but the word embedding dimension is 256. it is worth mentioning that, except for Seq2Seq (word), other models are based on characters. In order to ensure the consistency of our evaluation, we re-segment the results of Seq2Seq (word) into characters and calculate BLEU score.

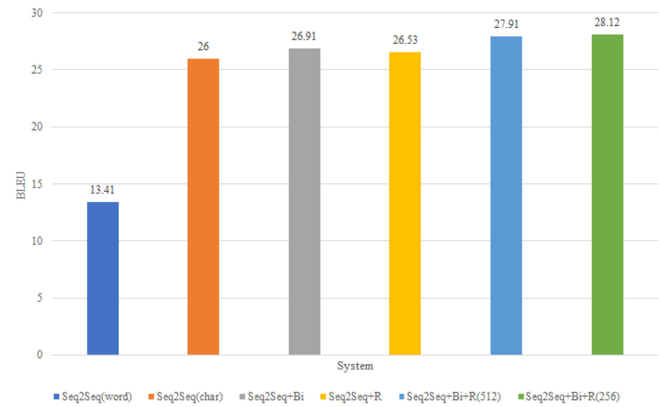

Fig. 4. BLEU-score of generation module.

From Figure 4, we can observe the following results. Firstly, comparing Seq2Seq (word) and Seq2Seq (char), we found that the character-based model has tremendous advantages. This may be because our number of query words is limited, so the model does not fit well. On the other hand, the character-based model greatly reduces our vocabulary and the appearance of OOV. Secondly, comparing models (ii), (iii), (iv), (v) and (vi), we found that the introduction of BI-LSTM and R-LSTM structures is effective. In particular, th introduction of the Bi-LSTM network, BLEU-score rises by 0.91. Finally, comparing (v) and (vi), a surprise finding is that word embedding is 256 which is more suitable for this task. We analyzed some cases and found that in the character-based model, higher dimensional word embedding may not effectively improve the effect, and even decrease.

D. *A/B Testing*

In our production system, we use live evaluation via A/B testing as the primary method for evaluating the performance of the PSQG system. The main principle of A/B testing is to provide two different design versions to different users at the same time, and to compare the effects of the two versions. We take a three-step analysis of the performance of our PSQG model, which is (1) the performance of the generation module (2) the recommendation module (3) the effect of overall framework PSQG.

Figure 5 shows the results of the A/B testing on the generation module. Under this experimental setup, we introduced the generation module without personalized recommendations. V1 and V2 are only user historical queries without generation modules, where V3 and V4 are with generation modules. Such AA/BB is set to reduce the impact of user turbulence. "Query with clicks" denotes the proportion of clicks on the query suggestions we present to the user. "Position $\leq$ 1" denotes that the user's first click of the query is ranked first, and other indicators have similar meanings, where 3, 5 and 10 respectively denote different positions.

From Figure 5, we can see that "position $\leq$ 1," "position $\leq$ 3," and "position $\leq$ 5" have some improvement, but "Query with clicks" and "position $\leq$ 10" are not obvious or even degraded. The user's query suggestions do not match all the users, which could not cause the user to click.

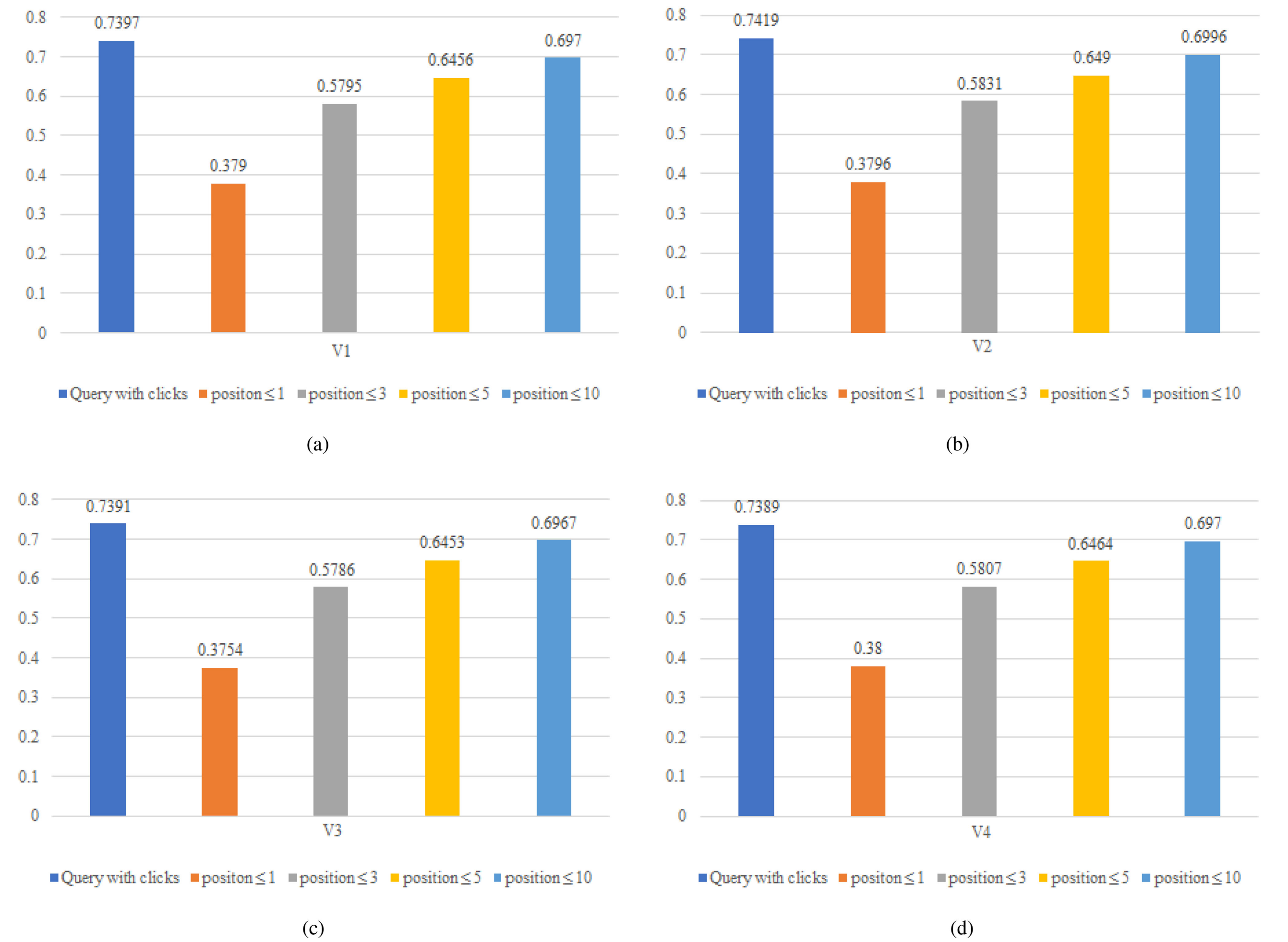

Fig. 5. A/B testing on generation module.

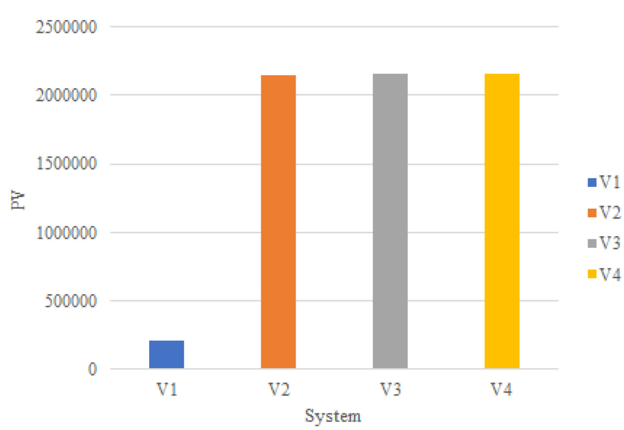

Fig. 6. A/B testing on personalized recommendation module for PV score.

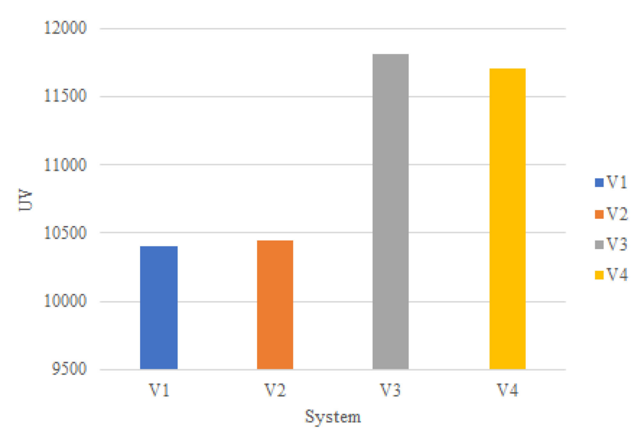

Fig. 7. A/B testing on personalized recommendation module for UV score.

Figures 6, 7, and 8 show the results of our A/B testing of the personalized recommendation module. V1 and V2 are introduced to the generation module without personalized recommendations, where V3 and V4 are introduced to the generation module with personalized recommendations.

PV denotes that the sum of the clicked query from all the query suggestions. UV denotes that are added as long as it is clicked during a query session. Besides, CTR denotes the proportion of all queries that are clicked, also using the AA/BB comparison experience.

Figures 6 and 7 show that both PV and UV increase by about 12%, which proves that we have been able to attract users' clicks to a large extent when we introduce personalized recommendations. The same trend can be seen in Figure 8.

In Figures 9, 10, and 11 we evaluated our overall framework. V1 is the generation module with the personalized

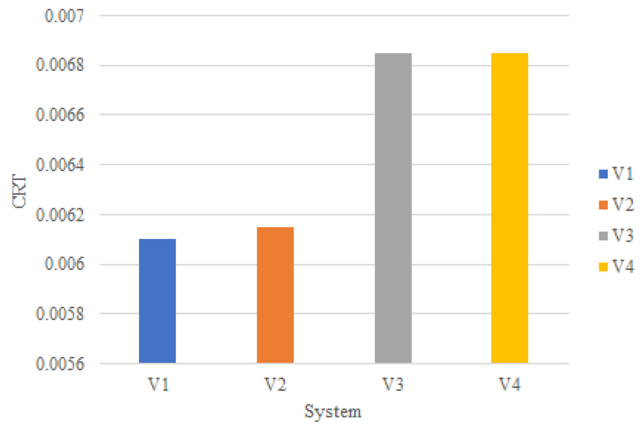

Fig. 8. A/B testing on personalized recommendation module for CRT rate.

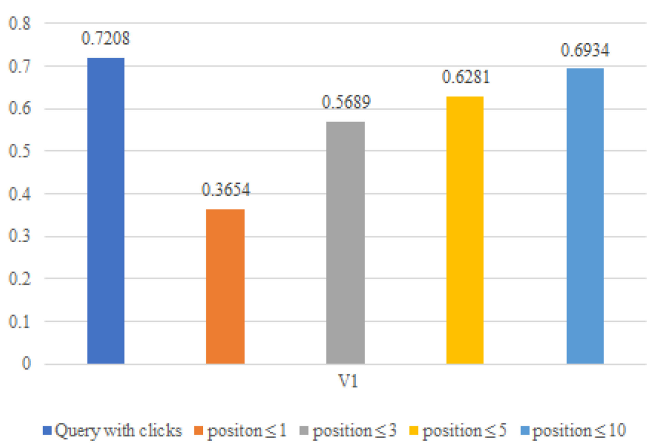

Fig. 9. A/B testing of PSQG model for V1.

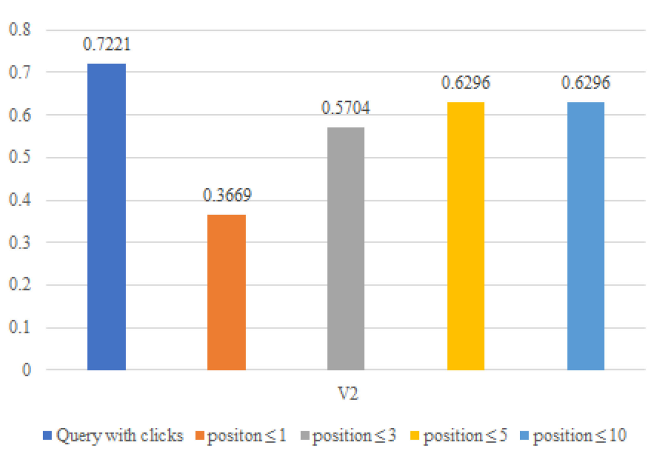

Fig. 10. A/B testing of PSQG model for V2.

recommendation module; V2 is a joint generation, extraction, and recall unified module with personalized recommendations; V3 denotes V2 as a supplement to V1.

From Figures 9, 10, and 11, we can see that our unified module improves significantly on all indicators. Even if it is only a supplement to V1, there is a certain improvement.

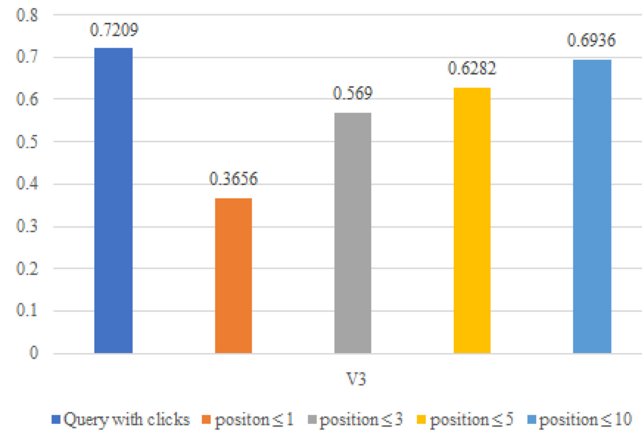

Fig. 11. A/B testing of PSQG model for V3.

TABLE II
THE FIRST SAMPLE QUERY SUGGESTIONS

Item	Description
A(1)	Huo Qigang Guo Jingjing's one-year-old daughter was exposed. The little girl's skin was white and her eyes were very beautiful
G	Huo Qigang Guo Jingjing Huo Qigang Guo family good Daughter
U	Huo Qigang Guo Jingjing Daughter Exposure Huo Qigang, daughter Huo Qigang and Guo Jingjing are at home Huo Qigang Guo Jingjing Wedding Banquet Singing

TABLE III
THE SECOND SAMPLE QUERY SUGGESTIONS

Item	Description
A(2)	Yin Jinbao, the chairman of Tianjin Rural Commercial Bank, cut his wrist and died. He just got into the fast lane and why is he braking? What is behind the three banks?
G	Tianjin Rural Commercial Bank Tianjin Rural Commercial good Farmers' bank
U	Death of chairman of Tianjin Rural Commercial Bank Chairman of Tianjin Rural Commercial Bank Tianjin Rural Commercial Bank New Chairman

E. *Result Analysis*

To better evaluate the effects of our joint generation, extraction, and recall unified module with personalized recommendations, we sampled several cases for further analysis. "A" denotes the title of the article; "G" denotes query suggestions sampled from the generation module with the personalized recommendation module; "U" denotes query suggestions sampled from a joint generation, extraction, and recall unified module with a personalized recommendation.

From Table II and Table III, we can see that the number of query suggestions by our unified module are significantly higher than the generation module. When we update the query suggestions in real-time, our unified module can provide diverse query suggestions to drive more users to click. For example, for A(1) in Table II, which not only provides the query of "GuoJingjing's daughter" but also generates "Huo Qigang and Guo Jingjing's wedding," and other query suggestions. From the quality of the query, our joint module's query contains more information. For example, for A(2) in Table III, we capture death's information instead of only one entity of the "Rural Commercial Bank".

To be more specific, we sample the query suggestions from our unified module and only generation modules individually on the same random 500 titles. The choice of query suggestions from our unified module is 65%.

VI. Conclusion

Information retrieval based on V2P by drivers becomes inevitable in modern vehicles running 6G-supported information systems and yet can be dangerous due to fragmenting the attention of drivers. Therefore, modern information retrieval systems can become a safety hazard. Scenarios in which drivers do not know their clear search requirements prolong driver-mobile interaction and increase safety risks. In this paper, a PSQG mechanism is proposed combining the extraction, generation, and recall to generate query candidates based on the user's historical search and interests to support personalized recommendations. Experiments show that, in terms of quality, the proposed mechanism generates high-correlation and high-fluency query candidates. This mechanism can reduce the driver-mobile interaction time and its attention requirements, thereby improving the overall safety of a 6G-supported vehicle information system.

In the future, we will estimate the influence of the PSQG model to determine the system's search and recommendation quality. Moreover, a real system also needs to be built to assist users in making precise queries efficiently.

References

[1] World Health Organization, "Road Traffic Injuries," 2020. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[2] P. Choudhary and N. R. Velaga, "Modelling driver distraction effects due to mobile phone use on reaction time," *Transp. Res. Part C. Emerg. Technol.*, vol. 77, no. 4, pp. 351–365, 2017.

[3] C. Gariazzo, M. Stafoggia, S. Bruzzone, A. Pelliccioni, and F. Forastiere, "Association between mobile phone traffic volume and road crash fatalities: A population-based case-crossover study," *Accid. Anal. Prev.*, vol. 115, no. 6, pp. 25–33, 2018.

[4] K. Sonklin, C. Wang, D. Jayalath, and Y. Feng, "Connected-vehicle data exchanges and positioning computing based on the publish-subscribe paradigm," *J. Comput. Commun.*, vol. 7, no. 10, pp. 82–94, 2019.

[5] Q. Hu, S. Wang, C. Hu, J. Huang, W. Li, and X. Cheng, "Messages in a concealed bottle: Achieving query content privacy with accurate location-based services," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7698–7711, Aug. 2018.

[6] S. Tu *et al.*, "Mobile fog computing security: A user-oriented smart attack defense strategy based on DQL," *Comput. Comm.*, vol. 160, pp. 790–798, 2020.

[7] M. Waqas, M. Ahmed, J. Zhang, and Y. Li, "Confidential information ensurance through physical layer security in device-to-device communication," in *Proc. IEEE Global Comm. Conf.*, 2018, pp. 1–7.

[8] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo, and C. Li, "Learning URL embedding for malicious website detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6673–6681, Oct. 2020.

[9] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An efficient privacy-enhanced attribute-based access control mechanism," *Concurr. Comp. Pract. E.*, vol. 32, no. 5, pp. 1–10, 2020.

[10] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, "A blockchain-based multi-cloud storage data auditing scheme to locate faults," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2021.3057771.

[11] C. He, Z. J. Wang, and C. Miao, "Query diversity schemes for backscatter RFID communications with single-antenna tags," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6932–6941, Aug. 2017.

[12] H. K. Azad and A. Deepak, "Query expansion techniques for information retrieval: A survey," *Inf. Process. Manag.*, vol. 56, no. 5, pp. 1698–1735, 2019.

[13] M. Shokouhi, "Learning to personalize query auto-completion," in *Proc. 36th Int*. ACM SIGIR *Conf. Res. Dev. Inf. Ret., Dublin*, Ireland, 2013, pp. 103–112.

[14] T. Pradhan and S. Pal, "A hybrid personalized scholarly venue recommender system integrating social network analysis and contextual similarity," *Future Gener. Comput. Syst.*, vol. 110, no. 9, pp. 1139–1166, 2020.

[15] M. Waqas *et al.*, "A comprehensive survey on mobility-aware D2D communications: Principles, practice and challenges," *IEEE Commun. Surv. Tut.*, vol. 22, no. 3, pp. 1863–1886, Jul.–Sep. 2020.

[16] W. Li, J. Cao, J. Wu, C. Huang, and R. Buyya, "A collaborative filtering recommendation method based on discrete quantum-inspired shuffled frog leaping algorithms in social networks," *Future Gener. Comput. Syst.*, vol. 88, no. 11, pp. 262–270, 2018.

[17] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Trans. Serv. Comput.*, vol. 13, no. 2, pp. 289–300, Mar./Apr. 2020.

[18] X. Zhou, W. Liang, I. Kevin, K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly big data," *IEEE Trans. Emerg. Top. Com.*, vol. 9, no. 1, pp. 246–257, 1 Jan.–March 2021.

[19] H. Jafarkarimi, A. T. H. Sim, and R. Saadatdoost, "A naive recommendation model for large databases," *Int. J. Info. Edu. Tech.*, vol. 2, no. 3, pp. 216–220, 2012.

[20] Q. Guo, Z. Sun, J. Zhang, and Y. Theng, "An attentional recurrent neural network for personalized next location recommendation," in *Proc. AAAI Conf. Artif. Intell*., 2020, pp. 83–90.

[21] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, "A blockchain-enabled deduplicatable data auditing mechanism for network storage services," *IEEE Trans. Emerg. Top. Comput*., to be published, doi: 10.1109/TETC.2020.3005610.

[22] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, "Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6182–6192, Sep. 2020.

[23] Y. Li, X. Tang, W. Cai, J. Tong, X. Liu, and G. Wang, "Resource-efficient index shard replication in large scale search engines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 12, pp. 2820–2835, Dec. 2019.

[24] X. Zhou, Y. Li, and W. Liang, "Cnn-rnn based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 18, no. 3, pp. 912–921, May-Jun. 2021.

[25] W. Niu, Z. Liu, and J. Caverlee, "On local expert discovery via geo-located crowds, queries, and candidates," *ACM Trans. Spat. Algorithms Syst.*, vol. 2, no. 4, pp. 1–24, 2016.

[26] F. Cai and M. de Rijke, "A survey of query auto completion in information retrieval," Found. *Trends Inf. Ret.*, vol. 10, no. 4, pp. 273–363, 2016.

[27] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7729–7739, Sep. 2016.

[28] H. C. Longuet-Higgins and A. Ortony, "The adaptive memorization of sequences," *Mach. Intell.*, vol. 3, no. 1, pp. 311–323, 1968.

[29] K. W. Church and B. Thiesson, "The wild thing," in *Proc. 43 rd Annu*. Meet. Assoc. Comput. Linguist., University of Michigan, USA, 2005, pp. 93–96.

[30] D. Dimitrov, F. Lemmerich, F. Flöck, and M. Strohmaier, "Query for architecture, click through military: Comparing the roles of search and navigation on wikipedia," in *Proc. 10th ACM Conf. Web Sci.*, Amsterdam, The Netherlands, 2018, pp. 371–380.

[31] Y. Xu, C. Zhang, Q. Zeng, G. Wang, and Y. Zhang, "Blockchain-enabled accountability mechanism against information leakage in vertical industry services," *IEEE Trans. Netw. Sci. Eng*., vol. 8, no. 2, pp. 1202–1213, Apr.-Jun. 2021.

[32] O. Ozdikis, H. Ramampiaro, and K. Nørvåg, "Spatial statistics of term co-occurrences for location prediction of tweets," in *Proc. 40th Eur. Conf. IR Res.*, Grenoble, France, 2018, pp. 494–506.

[33] C. Feng *et al.*, "Efficient and secure data sharing for 5G flying drones: A blockchain-enabled approach," *IEEE Netw.*, vol. 35, no. 1, pp. 130–137, Jan./Feb. 2021.

[34] K. Balabaeva, A. A. Funkner, and S. V. Kovalchuk, "Automated spelling correction for clinical text mining in Russian," in *Proc. Med. Info. Eur.*, Geneva, Switzerland, 2020, pp. 43–47.
[35] X. Yan, B. Cui, Y. Xu, P. Shi, and Z. Wang, "A method of information protection for collaborative deep learning under GAN model attack," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 3, pp. 871–881, May-Jun. 2021.
[36] H. Duan and B. P. Hsu, "Online spelling correction for query completion," in *Proc. 20th Inter. Conf. World Wide Web*, Hyderabad, India, 2011, pp. 117–126.
[37] Y. Wang, C. Guo, Y. Chu, J. Hwang, and C. Feng, "A cross-domain hierarchical recurrent model for personalized session-based recommendations," *Neurocomputing*, vol. 380, no. 3, pp. 271–284, 2020.
[38] F. Radlinski and T. Joachims, "Query chains: Learning to rank from implicit feedback," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Chicago, USA, 2005, pp. 239–248.
[39] M. Jahrer, A. Toscher, and R. Legenstein, "Combining predictions for accurate recommender systems," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min*., 2010, pp. 693–702.
[40] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. 36th Int. ACM SIGIR Conf. Res. Dev. Inf. Ret.*, Boston, USA, 2009, pp. 203–210.
[41] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. ACM Conf. Recommender Syst.*, Barcelona, Spain, 2010, pp. 39–46.
[42] Z. Zhang, C. Liu, Y. Zhang, and T. Zhou, "Solving the cold-start problem in recommender systems with social tags," *EPL* (*Europhysics Letters)*, vol. 92, no. 2, pp. 28002–280 10, 2010.
[43] L. Ramachandran and U. Murthy, "Ghosting: Contextualized inline query completion in large scale retail search," in *Proc. 13th ACM Conf. Recommender Syst.*, Copenhagen, Denmark, 2019, pp. 196–200.
[44] D. Beleveslis and C. Tjortjis, "Promoting diversity in content based recommendation using feature weighting and LSH," in *Proc., Artif. Intell. Appl. Innov*., 2020, pp. 452–461.
[45] D. Garigliotti and K. Balog, "Generating query suggestions to support task-based search," in *Proc. 36th Int*. ACM SIGIR Conf. Res. Dev. Inf. Ret., Tokyo, Japan, 2017, pp. 1153–1156.
[46] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Sur.*, vol. 52, no. 6, pp. 1–36, 2019.
[47] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1227–1240, Jun. 2020.
[48] M. Waqas, M. Zeng, Y. Li, D. Jin, and Z. Han, "Mobility assisted content transmission for device-to-device communication underlaying cellular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6410–6423, Jul. 2018.
[49] J. Kim, M. El-Khamy, and J. Lee, "Residual LSTM: Design of a deep recurrent architecture for distant speech recognition," in *Proc. 18th Annu. Conf. Int. Speech Commun. Assoc., Stockholm*, Sweden, 2017, pp. 1591–1595.
[50] R. Chakraborty, "Domain keyword extraction technique: A new weighting method based on frequency analysis," *Comput. Sci. Info. Tech.*, vol. 109, no. 1, pp. 109–118, 2013.
[51] J. H. Paik, "A novel TF-IDF weighting scheme for effective ranking," in *Proc. 36th Int*. ACM SIGIR Conf. Res. Dev. Inf. Ret., Dublin, Ireland, 2013, pp. 343–352.
[52] W. Che, Z. Li, and T. Liu, "LTP: A Chinese language technology platform," in *Proc. Int. Conf. Comput. Linguist.*, Beijing, China, 2010, pp. 13–16.
[53] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The Stanford Corenlp natural language processing toolkit," in *Proc. 52rd Annu*. Meet. Assoc. Comput. Linguist., Baltimore, USA, 2014, pp. 55–60.
[54] H. Peng, E. Cambria, and A. Hussain, "A review of sentiment analysis research in chinese language," *Cogn. Comput.*, vol. 9, no. 4, pp. 423–435, 2017.
[55] N. Li, B. Li, and L. Gao, "Transient stability assessment of power system based on xgboost and factorization machine," *IEEE Access*, vol. 8, no. 1, pp. 28 403–28414, 2020.
[56] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, and J. Zhang, "Hierarchical bidirectional rnn for safety-enhanced b5G heterogeneous networks," *IEEE Trans. Netw. Sci. Eng*., to be published, doi: 10.1109/TNSE.2021.3055762.
[57] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

Xiaodan Yan (Member, IEEE) received the Ph.D. degree in information security from the Beijing University of Posts and Telecommunications, Beijing, China, in 2020. He has authored or coauthored ten articles in international journals and conferences, including IEEE Transactions on Industrial Informatics and Transactions on Computational Biology and Bioinformatics. His research interests include network and systems security, networking, and distributed systems.He is currently engaged in mobile Internet, big data security analysis technology, and Internet of Things security technology research. He is a Member of China Computer Federation. He is a Reviewer of more than four international conferences.

Jiwei Zhang (Member, IEEE) received the bachelor's degree in information and computing science from Yantai University, Yantai, China and the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China. He is currently a Postdoctoral with the Beijing University of Posts and Telecommunications. He has authored or coauthored more than 17 articles in multiple international journals and conferences. He has owned more than three patents and software copyrights. His research interests include artificial intelligence, computer vision, and internet of things. He is a Reviewer of six international conferences and journals.

Haroon Elahi received the Doctoral degree from Guangzhou University, Guangzhou, China. Before that, he studied computer science and ubiquitous computing with the Blekinge Institute of Technology, Karlskrona, Sweden. He also received the degrees in IT and commerce. At the beginning of the Ph.D. career, his position in Pakistan was an Assistant Professor. He has also worked with project design, professional-trainings, software development, and product design. His research interests include the state of privacy and security of cyber-physical systems in the rapidly changing threat landscape. Besides, he is a preacher of ubiquitous computing design techniques and methods.

Meiyi Jiang is currently working toward the master's degree with the National Engineering Laboratory of Mobile Internet Security Technology, Beijing University of Posts and Telecommunications, Beijing, China, with a major in network information security. Her research interests include network security, computer vision, and natural language processing. She is currently mainly engaged in the field of computer vision and information security research.

Hui Gao received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China. He also has studied with the State Key Laboratory of Networking and Switching Technology. He is a master's Supervisor with the School of Computer Science (National Pilot Software Engineering School), at the same university. He has authored or coauthored more than 11 articles in multiple international journals and conferences. His research interests include Internet-of-Things and participatory sensing.