Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Cohort-based personalized query auto-completion*

Dan-yang JIANG‡, Hong-hui CHEN

Science and Technology on Information Systems Engineering Laboratory,
National University of Defense Technology, Changsha 410073, China
E-mail: danyangjiang@nudt.edu.cn; chenhonghui@nudt.edu.cn

Received Jan. 5, 2018; Revision accepted Aug. 5, 2018; Crosschecked Sept. 4, 2019

Abstract: Query auto-completion (QAC) facilitates query formulation by predicting completions for given query prefix inputs. Most web search engines use behavioral signals to customize query completion lists for users. To be effective, such personalized QAC models rely on the access to sufficient context about each user's interest and intentions. Hence, they often suffer from data sparseness problems. For this reason, we propose the construction and application of cohorts to address context sparsity and to enhance QAC personalization. We build an individual's interest profile by learning his/her topic preferences through topic models and then aggregate users who share similar profiles. As conventional topic models are unable to automatically learn cohorts, we propose two cohort topic models that handle topic modeling and cohort discovery in the same framework. We present four cohort-based personalized QAC models that employ four different cohort discovery strategies. Our proposals use cohorts' contextual information together with query frequency to rank completions. We perform extensive experiments on the publicly available AOL query log and compare the ranking effectiveness with that of models that discard cohort contexts. Experimental results suggest that our cohort-based personalized QAC models can solve the sparseness problem and yield significant relevance improvement over competitive baselines.

Key words: Query auto-completion; Cohort-based retrieval; Topic models
https://doi.org/10.1631/FITEE.1800010 **CLC number:** TP311.5

1 Introduction

Query auto-completion (QAC) is a widely known and deployed mechanism to facilitate the task of formulating queries in search engines. As illustrated in Fig. 1, by updating a ranked list of query completions that start with the current prefix, QAC systems help users submit queries in less time and with less effort.

Typically, query prefixes tend to be short and ambiguous, making it difficult to predict users' intent and to accurately suggest relevant completions. In the worst case, a user has to manually type the entire query. The most common and intuitive approach is to rank completions by their past or future popularity (Bar-Yossef and Kraus, 2011; Cai and de Rijke, 2016b), which aims at maximizing the QAC effectiveness for all users.

However, it is far from optimal because the one-size-fits-all approach fails to take users' context information, such as submitted queries and click-through data, into consideration while such information often influences users' intended queries. In light of this, personalized QAC ranking models that use contextual information (Shokouhi, 2013; Cai et al., 2016a; Li et al., 2017b) have been proposed to suggest relevant queries. However, such personalization is effective only when there is a large amount of data

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. 61702526), the Defense Industrial Technology Development Program of China (No. JCKY2017204B064), and the National Advanced Research Project of China (No. 6141B0801010b)
ORCID: Dan-yang JIANG, http://orcid.org/0000-0001-6012-5313

available about individuals. Unfortunately, users' context is often very sparse and insufficient to identify their interest and intentions. Fig. 2 demonstrates the percentage of users submitting more than 10 queries in the AOL query log (Pass et al., 2006). Fig. 2 shows that 22% of all users' issues are less than five queries in a three-month period, and 62% users' issues are no more than 10 queries. Further compounding this issue, existing works often ignore the data sparsity problem in the personalized QAC and users with little search history are always left out. These deficiencies led us in a different research direction of personalized QAC, in which we use cohort context to deal with data sparsity and achieve a robust personalization performance.

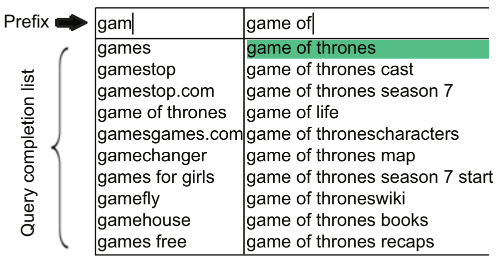

Fig. 1 Illustration of the query auto-completion (QAC) in a commercial search engine for the query "game of thrones"

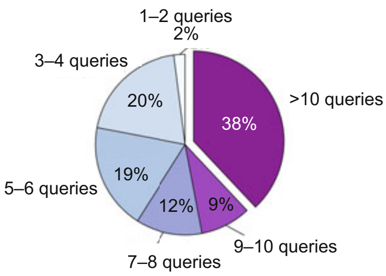

Fig. 2 The number of queries submitted by users in the publicly available AOL query log

A cohort is a group of users who share common characteristics (Hassan and White, 2013). It can address data sparseness when users' data is insufficient. Therefore, the cohort has been effectively used in applications such as recommendation systems and web search (Teevan et al., 2009; Yan et al., 2014). Given a user, cohort-based models enhance personalization by providing context (if sufficient information is unavailable) or an additional context to build personalization models richer if the personalization models already exist (Yan et al., 2014). Nevertheless, there has been no study of applying cohort-based models to improve QAC personalization. We solve this problem by proposing four cohort-based personalized QAC models that learn cohorts of interest through four distinct topic models. Apart from conventional topic models, we propose two cohort topic models (CTMs) that introduce the cohort as a latent variable to uncover hidden cohort information from individuals' topic interest. A delightful bonus is the CTMs' ability to act as a soft clustering technique that assigns each user to multiple cohorts instead of a single one. Extensive experiments are constructed on a real-world query log. The experimental results show that cohort-based personalized QAC models greatly improve ranking effectiveness with the mean reciprocal rank (MRR) score increased by 1.5% when compared with the personalized QAC without cohorts. In addition, our CTMs are more efficient in identifying cohorts than conventional topic models, which further enhances the QAC performance.

We consider our contributions to be four-fold: (1) We tackle the challenge of personalized QAC in a novel way by exploiting the contextual information of user cohorts; (2) We learn individuals' latent topic interest via topic models rather than predefined topical categories; (3) We propose two CTMs that loosely cluster users into multiple cohorts based on their interest; (4) We analyze the effectiveness of the proposed cohort-based personalized QAC models, which consider both query popularity and cohort context, and find that our models significantly outperform the competitive baselines.

2 Related work

There are a lot of works relevant to the research described in this study, including QAC ranking, collaborative web search techniques, and topic modeling.

2.1 Query auto-completion

QAC is among the first services with which users interact as they form and search their queries. Most of the existing QAC works focus on relevance ranking. For this purpose, early QAC models, such as most popular completion (MPC), employ previous query popularity as the only ranking signal (Bar-Yossef and Kraus, 2011). In essence, MPC assumes that the current query popularity is the same as the past, which is obviously insufficient to yield satisfactory completions. Numerous QAC rankers have been proposed to strengthen MPC from various aspects. Using time-series analysis techniques to obtain temporal patterns of query frequency, time-based QAC ranks completions by their predicted future popularity (Shokouhi and Radinsky, 2012; Cai et al., 2016b). Shokouhi and Radinsky (2012) proposed a long-term time-series approach to model queries' periodic patterns and to forecast the future popularity for QAC ranking. Learning-based QAC uses a supervised algorithm to train personalized QAC rankers with features generated from user-specific information (Shokouhi, 2013; Cai and de Rijke, 2016a; Jiang and Cheng, 2016). Shokouhi (2013) applied Lambda-MART (Burges et al., 2011) as the learning algorithm and developed features based on users' age, gender, location, and short- and long-term history. The neural network has been employed to train a model that reranks (Mitra, 2015) or generates (Park and Chiba, 2017) completions for QAC. More researchers have begun to explore behavior- and interaction-based QACs. Li et al. (2017b) collected a high-resolution QAC query log that records every keystroke and associated system response leading to the final clicked query, and then they used click models to model the QAC process with emphasis on users' behaviors. Zhang et al. (2015) studied implicit feedback during user-QAC interaction, and proposed a model that adapts QAC to users' implicit negative feedback to unselected completions. Mitra et al. (2014) examined individuals' interaction patterns with QAC and suggested that users are more likely to engage with auto-completion at word boundaries.

The aforementioned personalized QAC models, such as learning- and interaction-based models, require sufficient and detailed user information to ensure effectiveness; however, none of the existing works aim at investigating the context sparseness problem in QAC. We take the initiative and show that grouping users into cohorts can mitigate the sparsity issue. The QAC model proposed by Li et al. (2017a) is most similar to our work, capturing the correlation between users' behavior patterns through a probabilistic model based on latent Dirichlet allocation (LDA). However, our model is different from that in Li et al. (2017a) in the following ways: (1) Our goal is to improve QAC performance when there is little context available; however, Li et al. (2017a) did not consider this problem because they had abundant data from both QAC and click logs. (2) The CTMs we propose are designed to cluster users by user-cohort distribution, whereas the contextual-LDA model in Li et al. (2017a) has been used to analyze the behavior of individuals by behavior-pattern distribution. (3) We use context information from both users and their cohorts to suggest query completions, whereas Li et al. (2017a) have ranked completions based on behavior information from a single individual.

2.2 Collaborative web search

When there is insufficient data about the current user, the search behavior of other related users may be beneficial in modeling his/her interest and intentions. Based on this motivation, collaborative search techniques have long been introduced to improve the accuracy of web search engines, and specially designed systems have been presented to help searchers successfully collaborate on realistic web search tasks (Smyth et al., 2003; Morris and Horvitz, 2007). Recent works are concerned with a more comprehensive and cooperative way to process individuals' search behavioral data. By computing query similarity and result similarity, White et al. (2013) found users who historically performed tasks similar to the current user, and used their on-task behaviors to improve personalization performance. Aiming at overcoming data sparseness in personalization, Yan et al. (2014) described a characterization and evaluation of the use of cohort modeling. Yan et al. (2014) experimented with three pre-defined cohorts, i.e., topic, location, and top-level domain preference, independently and in combination, and showed that exploiting cohort behavior can yield significant relevance gains. Hassan and White (2013) developed machine-learning satisfaction models tailored to an individual searcher and his/her cohorts. Hassan and White (2013) found that tailoring models of dissatisfaction to similar users outperformed the baseline that applied the same model across all users, presenting a promising direction toward the development of more tailored satisfaction prediction.

Although collaborative techniques have been used in personalized web search, currently there is no work of applying them to QAC task. To the best of our knowledge, this study is the first piece of work that uses cohort-based models for QAC ranking.

2.3 Topic model

Since the introduction of LDA (Blei et al., 2003), various topic models have been proposed and employed to discover topic structures of large-scale corpora (Steyvers et al., 2004; Chen et al., 2012). Among them, topic models for community discovery most resemble our work. By placing topic variables on authors' links in documents, Zheng et al. (2011) proposed a general topic model of community discovery for multi-author link data. Zheng et al. (2011)'s proposal made it possible to mine detailed information from each author's participation and provided a reasonable interpretation of the discovered communities. Unlike Zheng et al. (2011), who considered links in the topic model of text mining, Yin et al. (2012) focused on text-associated graphs, in which they treated communities as pseudo-documents and explored the relationships between terms and communities. Yin et al. (2012) confirmed their hypothesis that topics could help understand community structure, and that community structure could help model topics. Because previous studies did not investigate community evolution over time, Li et al. (2012) presented a topic model that extends a community topic model with time variables, so that they could capture the dynamic changes in communities.

The above-mentioned community discovery topic models have been generally used to identify the underlying semantic structure of a document collection instead of addressing the data sparsity problem in personalization. We take advantage of topic models in uncovering latent cohorts of interest, and offer a potential direction to overcome context sparseness in personalized QAC.

3 Cohort discovery modeling

In this section we will investigate cohort discovery by means of topic models. We first introduce cohort modeling using two conventional topic models, and then describe the construction of cohort topic models which incorporate cohorts as latent variables.

3.1 Conventional topic models

Topic interest serves as an important indicator of users' search intentions. Many works use topical categories, such as human-generated web ontology provided by the Open Directory Project (dmoz.org), to model users' interest and to improve various aspects of web search (e.g., document ranking, query suggestion, and query classification) (Hassan and White, 2013). Nonetheless, these pre-defined topics limit the flexibility and accuracy of interest modeling. Therefore, our work uses topic models to build user profiles so that topics and cohorts are discovered in the topic modeling process.

As illustrated in Fig. 3, LDA is a general Bayesian probabilistic topic model that represents documents as finite mixtures over latent topics $\boldsymbol{z}$. Table 1 lists the major notations used in this study. The key problem to solve using LDA is computing the posterior distribution of latent topics given words $\boldsymbol{w}$ of a document:

$$p(\boldsymbol{z}|\boldsymbol{w}) = \frac{p(\boldsymbol{z},\boldsymbol{w})}{p(\boldsymbol{w})} = \frac{\prod_{i=1}^{W} p(z_i, w_i)}{\prod_{i=1}^{W}\sum_{k=1}^{K} p(z_i = k, w_i)}. \quad (1)$$

Because the denominator of Eq. (1) is a summation over K^W words, the exact inference of $p(\boldsymbol{z}|\boldsymbol{w})$ is generally intractable. However, we can use the full conditional probability $p(z_i|\boldsymbol{z}_{\neg i},\boldsymbol{w})$ to simulate $p(\boldsymbol{z}|\boldsymbol{w})$, which leads to

$$p(z_i{=}k|\boldsymbol{z}_{\neg i},\boldsymbol{w}) \propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}\left(n_{k,\neg i}^{(t)} + \beta_t\right)}\left(n_{m,\neg i}^{(k)}{+}\alpha_k\right). \quad (2)$$

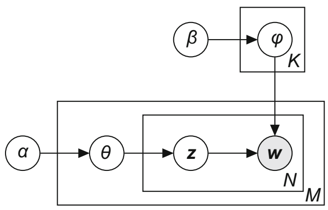

Fig. 3 Graphical representation for latent Dirichlet allocation

Applying the expectation of the Dirichlet distribution, $<\mathrm{Dir}(\boldsymbol{n})>= n_i/\sum_i n_i$, to Eq. (2) yields

$$\begin{cases} \varphi_{k,t} = \dfrac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}\left(n_{k,\neg i}^{(t)} + \beta_t\right)}, \\ \theta_{m,k} = \dfrac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}\left(n_{m,\neg i}^{(k)} + \alpha_k\right)}, \end{cases} \quad (3)$$

where $\varphi_{k,t} = p(w_i = t|z_i = k)$ represents the probability of using word t in topic k, and $\theta_{m,k} = p(z_i =$

Table 1 Notations used in the topic models

Notation	Description
M, K, W, V, A, S	The numbers of documents, latent topics, words in corpus, words in vocabulary, users, and latent cohorts, respectively
N	Document length
α, β, γ, δ	Hyperparameters of Dirichlet distributions
θ	Document-topic distribution
φ	Topic-word distribution
ϕ	User-topic distribution
μ	User-cohort distribution
χ	Cohort-topic distribution
ψ	Cohort-user distribution
η	Document-cohort distribution
n_m	The number of times that document d_m is clicked by a user
u_d	A set of users who click document d_m
$n_{m,\neg i}^{(k)}$	The number of times that topic k is observed in document d_m with token i being excluded
$n_{k,\neg i}^{(t)}$	The number of times that word t is assigned to topic k with token i being excluded
$n_{a,\neg i}^{(k)}$	The number of times that topic k is assigned to user a with token i being excluded
$n_{a,\neg i}^{(s)}$	The number of times that cohort s is assigned to user a with token i being excluded
$n_{s,\neg i}^{(k)}$	The number of times that topic k is assigned to cohort s with token i being excluded
$n_{m,\neg i}^{(s)}$	The number of times that cohort s is observed in document d_m with token i being excluded
$n_{s,\neg i}^{(a)}$	The number of times user a is assigned to cohort s with token i being excluded

$k|d_m)$ denotes the probability of topic k over document d_m. Using Eqs. (2) and (3), the Gibbs sampling procedure can be run. Hence, we formulate the user-topic distribution as

$$p(\boldsymbol{z}|u_i = a) = \sum_{m=1}^{M} \frac{n_m}{\sum_{j=1}^{M} n_j} \cdot p(\boldsymbol{z}|d_m), \tag{4}$$

where $n_m/\sum_{j=1}^{M} n_j$ is the probability that user a clicks document d_m in the entire corpus. For each user in the search log, we can obtain a $1 \times K$ topic interest vector. By applying general clustering methods to these vectors, we can group users into cohorts with the similar topic interest.

Despite being informative about the content of documents, LDA does not provide direct information about the interest of the authors of those documents. Therefore, the author topic model (ATM) (Steyvers et al., 2004) extends LDA by including authorship information in every document of the corpus, i.e., u_d, as an observed variable. The plate notation for ATM is demonstrated in Fig. 4. We adapt ATM to the QAC scenario based on the assumption that the users who click document d_m are the co-authors of d_m. Given a document, by applying Gibbs sampling which is similar to LDA, the topic and author assignments are sampled from

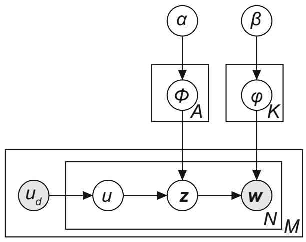

Fig. 4 Graphical representation for the author topic model ($\boldsymbol{u}$ represents an author randomly chosen from $\boldsymbol{u_d}$)

$$p(z_i = k, u_i = a|\boldsymbol{z}_{\neg i}, \boldsymbol{u}_{\neg i}, \boldsymbol{w}) \propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}\left(n_{k,\neg i}^{(t)} + \beta_t\right)} \frac{n_{a,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}\left(n_{a,\neg i}^{(k)} + \alpha_k\right)}, \tag{5}$$

and the parameters are estimated by

$$\begin{cases} \varphi_{k,t} = \dfrac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}\left(n_{k,\neg i}^{(t)} + \beta_t\right)}, \\ \phi_{a,k} = \dfrac{n_{a,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}\left(n_{a,\neg i}^{(k)} + \alpha_k\right)}, \end{cases} \tag{6}$$

where $\varphi_{k,t}$ is the same as that of LDA and $\phi_{a,k} = p(z_i = k|u_i = a)$ is the probability of using topic k

by user a. As stated earlier, u_d can be seen as a set of users who click document d_m. Thus, the user-topic distribution $p(\phi = a)$ is a $1 \times K$ vector with element k equaling $\phi_{a,k}$. Cohorts can be grouped by general clustering methods accordingly.

3.2 Cohort topic models

Because the above-mentioned two conventional topic models cannot automatically group similar users, general clustering methods, e.g., the K-means algorithm and K-nearest neighbor classification (Wu et al., 2008), should be employed to group users after computing $p(\boldsymbol{z}|\boldsymbol{u})$. However, these methods are hard clustering techniques, where each user is exactly allocated to one cohort. For users with diverse preferences, it is natural to allow multiple cohort memberships. Therefore, a soft clustering approach may produce higher performance gain because it can better capture within-user variance in interest. This motivates us to propose CTMs, which incorporate cohorts as latent variables. We separate the concepts of cohort and topic, so that one cohort can correspond to multiple topics and multiple cohorts can share one topic. Two distinct strategies are proposed to integrate cohort discovery in topic modeling. Compared with hard clustering methods, CTMs act as soft clustering techniques that assign each user to multiple cohorts with a probability associated with each cohort.

Fig. 5 depicts the plate notation for our first CTM, which is referred to as the cohort topic model 1 (CTM1) hereinafter. In this model, each user is associated with a multinomial distribution over cohorts, represented by μ; each cohort is associated with a multinomial distribution over topics, represented by χ; each topic is associated with a multinomial distribution over words, represented by φ. The multinomial distributions of μ, χ, and φ are generated from the symmetric Dirichlet priors with hyperparameters γ, α, and β, respectively. The generation process of words in CTM1 can be described as follows:

1. Generate multinomial distributions: (1) For each user u in the corpus, choose $\mu \sim \text{Dirichlet}(\gamma)$; (2) For each cohort c, choose $\chi \sim \text{Dirichlet}(\alpha)$; (3) For each topic z, choose $\varphi \sim \text{Dirichlet}(\beta)$.

2. Generate each word w in each document d_m of the corpus: (1) Set the vector of users u_d; (2) Conditioned on u_d, sample a user $u_i = a \sim \text{Uniform}(u_d)$;

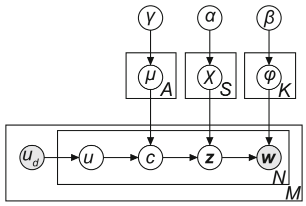

Fig. 5 Graphical representation for cohort topic model 1 (CTM1)

(3) Conditioned on a, sample a cohort $c_i = s \sim \text{Discrete}(\mu_a)$; (4) Conditioned on s, sample a topic $z_i = k \sim \text{Discrete}(\chi_s)$; (5) Conditioned on k, sample a word $w_i = t \sim \text{Discrete}(\varphi_k)$.

The second step is repeated N times to form document d_m. The update equation from which the Gibbs sampler draws the hidden variables is

$$\begin{aligned} &p(z_i = k, c_i = s, u_i = a|\boldsymbol{z}_{\neg i}, \boldsymbol{c}_{\neg i}, \boldsymbol{u}_{\neg i}, \boldsymbol{w}) \\ &\quad\propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}\left(n_{k,\neg i}^{(t)} + \beta_t\right)} \frac{n_{s,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}\left(n_{s,\neg i}^{(k)} + \alpha_k\right)} \\ &\quad\cdot \frac{n_{a,\neg i}^{(s)} + \gamma_s}{\sum_{s=1}^{S}\left(n_{a,\neg i}^{(s)} + \gamma_s\right)}. \end{aligned} \tag{7}$$

The approximated parameters are expressed as

$$\begin{cases} \chi_{s,k} = \dfrac{n_{s,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}\left(n_{s,\neg i}^{(k)} + \alpha_k\right)}, \\ \mu_{a,s} = \dfrac{n_{a,\neg i}^{(s)} + \gamma_s}{\sum_{s=1}^{S}\left(n_{a,\neg i}^{(s)} + \gamma_s\right)}, \end{cases} \tag{8}$$

where $\chi_{s,k} = p(z_i = k|c_i = s)$ represents the probability of using topic k in cohort s, and $\mu_{a,s} = p(c_i = s|u_i = a)$ denotes the probability of cohort s belonging to user a. Because $\varphi_{k,t}$ is the same as those of LDA and ATM, we omit its estimation for simplicity.

Different from CTM1, our second cohort discovery approach, the cohort topic model 2 (CTM2), builds on the idea that each user is randomly chosen from a cohort instead of co-authors of a document. As illustrated in Fig. 6, the multinomial distribution over cohorts for each document is parameterized by η; the multinomial distribution over users for each cohort is parameterized by ψ; the multinomial distribution over topics for each user is parameterized by ϕ; the multinomial distribution over words for each topic is parameterized by φ. Symmetric Dirichlet priors with hyperparameters γ, δ, α, and β are placed over the four distributions, respectively. The generation process of each word in CTM2 is listed below:

1. Generate multinomial distributions: (1) For each document d_m, choose $\eta \sim$ Dirichlet(γ); (2) For each cohort c, choose $\psi \sim$ Dirichlet(δ); (3) For each user u in the corpus, choose $\phi \sim$ Dirichlet(α); (4) For each topic z, choose $\varphi \sim$ Dirichlet(β).

2. Generate each word w in each document d_m of the corpus: (1) Conditioned on d_m, sample a cohort $c_i = s \sim$ Discrete(η_m); (2) Conditioned on s, sample a user $u_i = a \sim$ Discrete(ψ_s); (3) Conditioned on a, sample a topic $z_i = k \sim$ Discrete(ϕ_a); (4) Conditioned on k, sample a word $w_i = t \sim$ Discrete(φ_k).

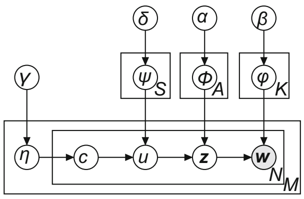

Fig. 6 Graphical representation of cohort topic model 2 (CTM2)

Similar to CTM1, we calculate the posterior conditional probability for the Gibbs sampling procedure by

$$\begin{aligned} &p(z_i = k, u_i = a, c_i = s | \boldsymbol{z}_{\neg i}, \boldsymbol{u}_{\neg i}, \boldsymbol{c}_{\neg i}, \boldsymbol{w}) \\ &\propto \frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}\left(n_{k,\neg i}^{(t)} + \beta_t\right)} \frac{n_{a,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}\left(n_{a,\neg i}^{(k)} + \alpha_k\right)} \\ &\cdot \frac{n_{s,\neg i}^{(a)} + \delta_a}{\sum_{a=1}^{A}\left(n_{s,\neg i}^{(a)} + \delta_a\right)} \frac{n_{m,\neg i}^{(s)} + \gamma_s}{\sum_{s=1}^{S}\left(n_{m,\neg i}^{(s)} + \gamma_s\right)}. \end{aligned} \tag{9}$$

Parameters of CTM2 are estimated by

$$\begin{cases} \psi_{s,a} = \dfrac{n_{s,\neg i}^{(a)} + \delta_a}{\sum_{a=1}^{A}\left(n_{s,\neg i}^{(a)} + \delta_a\right)}, \\ \eta_{m,s} = \dfrac{n_{m,\neg i}^{(s)} + \gamma_s}{\sum_{s=1}^{S}\left(n_{m,\neg i}^{(s)} + \gamma_s\right)}, \end{cases} \tag{10}$$

where $\psi_{s,a} = p(u_i = a | c_i = s)$ represents the probability of user a's interest in cohort s, and $\eta_{m,s} = p(c_i = s | d_m)$ is the probability of assigning cohort s to document d_m. Because $\phi_{a,k}$ is identical to that of ATM, we leave out its approximation for simplicity.

Overall, our CTMs assign each user to each cohort with a membership weight ($\mu_{a,s}$ in CTM1 and $\psi_{s,a}$ in CTM2). Consequently, users with similar interest can be softly clustered into multiple cohorts by their values of $\mu_{a,s}$ or $\psi_{s,a}$.

The time complexity of the four topic models mentioned above is listed in Table 2. For all these topic models, the complexity of each iteration of the Gibbs sampling process is linear to the total number of hidden variables, i.e., K and S. We observe that CTM1 and CTM2 have the same time complexity. Generally, for a given corpus, the value of KW far exceeds that of AS, i.e., $KW \gg AS$. Therefore, the complexity of CTMs is approximately of the same order as that of LDA and ATM. As Gibbs sampling can be parallelized by MapReduce (Neiswanger et al., 2014), the four topic models are scalable to large-scale datasets.

Table 2 Time complexity of the four topic models

Topic model	Time complexity
LDA	$O(KW)$
ATM	$O(AKW)$
CTM1	$O(ASKW)$
CTM2	$O(SAKW)$

4 Cohort-based personalized query auto-completion

We formally define the QAC task as follows: given a query log Q and a set of query completions $C(p)$ that matches the typed prefix p, a QAC system can rank query completions in $C(p)$ by the available ranking signals.

As mentioned in Section 3, four distinct topic models are employed, i.e., LDA, ATM, CTM1, and CTM2, to identify users with similar interest. Therefore, given user a, for each latent cohort s, we iteratively select the user's cohort members by

$$\begin{cases} b = \arg\max_{y \in U} \mathrm{sim}(a, y), \\ \mathrm{sim}(a, y) = \begin{cases} 1/D(a, y), & \text{LDA or ATM}, \\ \mu_{a,s} \cdot \mu_{y,s}, & \text{CTM1}, \\ \psi_{s,a} \cdot \psi_{s,y}, & \text{CTM2}, \end{cases} \end{cases} \tag{11}$$

where U denotes the set of all users, $\mathrm{sim}(a, y)$ the resemblance of users a and y, and $D(a, y)$ the Euclidean distance between $p(\boldsymbol{z}|a)$ and $p(\boldsymbol{z}|y)$. Note that user a is a cohort member of himself/herself with $\mathrm{sim}(a, a) = 1$. Additionally, considering computational efficiency and noise reduction, for each user a, only the top L users in terms of $\mathrm{sim}(a, y)$ are regarded as the cohort members of a.

Once the cohorts of a user are identified, we can use the contextual information of the user himself/herself and his/her cohorts to rank query completions. The scoring function of our cohort-based personalized QAC is a convex combination of two ranking signals expressed as

$$\mathrm{Score}(q) = \lambda \mathrm{FreqScore}(q) + (1-\lambda)\mathrm{CoScore}(q), \quad (12)$$

where $\lambda \in [0, 1]$ is a tunable parameter determining the weights of two signals, $\mathrm{FreqScore}(q)$ is based on the frequency of completion q, and $\mathrm{CoScore}(q)$ calculates the similarity between completion q and historical queries submitted by cohorts. Specifically, we output the two scores as

$$\begin{cases} \mathrm{FreqScore}(q) = \dfrac{f(q)}{\max_{q_c \in C(p)} f(q_c)}, \\ \mathrm{CoScore}(q) = \mathrm{norm}(\omega_j) \cdot \mathrm{sim}(q, q_j), \end{cases} \quad (13)$$

where $\mathrm{norm}(\omega_j)$ is the contribution from each cohort member to ensure $\sum_j \omega_j = 1$, satisfying

$$\mathrm{norm}(\omega_j) = \frac{\mathrm{sim}(a, b_j)}{\sum_{j=1}^{L} \mathrm{sim}(a, b_j)},$$

$f(q)$ denotes the number of times completion q occurs in $\log Q$, and $\mathrm{sim}(q, q_j)$ the N-gram similarity between completion q and query q_j (submitted by cohort member b_j). Because $\mathrm{FreqScore}(q)$ and $\mathrm{CoScore}(q)$ use different units and scales, we standardize them according to Cai et al. (2016b) before combination.

5 Experimental setup

In this section, we first list the questions to guide our experiments, and then give details about the experimental setup.

5.1 Research questions

We address the following research questions in this study: (1) How can we compare our cohort-based personalized QAC models against models that ignore cohort context? (2) Do CTMs outperform conventional topic models in terms of QAC ranking effectiveness? (3) Which part contributes more to a better QAC ranking, query popularity or query similarity? (4) How does the number of cohorts affect the performance of QAC models based on CTMs? (5) Do cohort-based models address the data sparseness problem in personalized QAC?

5.2 Dataset

We used the publicly available AOL dataset (Pass et al., 2006) to conduct our experiments. This dataset comprises sampled queries submitted by anonymized users to the AOL search engine from March 1 2006 to May 31 2006, which is sufficiently large to ensure statistical significance. We removed the navigational queries containing URL substrings and discarded queries starting with special characters. Considering the fact that the proposed QAC models need users' contextual information to implement personalization, users who issued four or more queries were kept. Descriptions of the clicked documents were extracted from the Open Directory Project (ODP). After cleaning and filtering, the dataset consisted of 122 029 unique clicked documents associated with 274 135 distinct queries submitted by 144 646 unique anonymous users. The dataset was then split into a training set and a testing set with ratios of 75% and 25%, respectively. Traditional k-fold cross-validation is not applicable to streaming settings because it would disorder the temporal data sequence (Gama et al., 2014). Therefore, queries in the training set were the first 75% of queries submitted by each user, and the latter 25% of queries formed the testing set.

5.3 Evaluation metric and baselines

As a ranking task, we evaluated QAC performance in terms of MRR. This is the average reciprocal rank in the list of query completions, expressed as

$$\mathrm{MRR} = \frac{1}{|T|} \sum_{q^* \in T} \frac{1}{\mathrm{rank}_{q^*}}, \quad (14)$$

where T denotes the length of the testing set and rank_{q^*} the position of the final submitted query q^* in the query completion list. If no matched query is found in the list, $1/\mathrm{rank}_{q^*}$ is set to 0.

To verify the effectiveness of our proposals, the following competitive methods were adopted as baselines: (1) the MPC model that ranks query completions by their aggregated occurrences in the whole training period (Bar-Yossef and Kraus, 2011); (2) the personalized QAC model that uses only the contextual information of the user himself/herself to compute CoScore(q) in Eq. (13), denoted as P-QAC (Bar-Yossef and Kraus, 2011); (3) the learning-based model that handles the short context QAC problem by classifying users' search intensions, referred to as C-QAC (Jiang and Cheng, 2016); (4) the neural-based model which addresses the QAC problem by integrating user, query, and time information in a recurrent neural network (RNN), indicated as N-QAC (Fiorini and Lu, 2018). Four variations of the cohort-based personalized QAC model were considered, ranking query completions based on Eq. (12) but differing in cohort discovery modeling: (1) LDA-QAC used LDA and the general clustering method to uncover users' topic interests and to group cohorts; (2) ATM-QAC employed ATM and general clustering technique to identify users' interest in various topics and cluster cohorts; (3) CTM1-QAC uesd CTM1 to assign users to multiple cohorts; (4) CTM2-QAC exploited CTM2 to dynamically learn cohorts.

5.4 Settings and parameters

In all of our experiments, we fixed the number of topics K at 80. Topic modeling results were reported after 50 empirical iterations. Hyperparameters in our cohort discovery modeling were set as $\alpha = 50/K$, $\beta = 0.01$, $\gamma = 50/S$, and $\delta = 0.1$. We set $L = 6$ when computing CoScore(q) in Eq. (13) across all experiments. In addition, in LDA-QAC and ATM-QAC, the K-means algorithm was employed to cluster cohorts. Following Shokouhi (2013), we set the size of N-grams at three to compute query similarity. For C-QAC and N-QAC, we adopted the same settings as in Jiang and Cheng (2016) and Fiorini and Lu (2018), respectively. In our QAC ranking experiments, for each query in the testing set, we gave a ranked list of the top 20 query completions corresponding to prefix length p ranging from 1 to 5.

6 Results and discussion

6.1 Overall performance

We first investigated whether our cohort-based personalized QAC models had advantages over competitive baselines on relevance ranking (question 1). Table 3 summarizes the evaluation results of the QAC rankings produced by different QAC models. Apart from the absolute MRR scores at various prefix lengths (p), the average MRR score of each QAC ranker under all prefix lengths was reported. The pairwise differences of four cohort-based personalized QAC models against P-QAC were detected and marked in the upper right-hand corner of the corresponding scores.

It can be seen from Table 3 that MPC performs the worst among all compared QAC models, with the MRR scores of the seven personalized QAC rankers far exceeding that of MPC. For the four baselines, P-QAC, C-QAC, and N-QAC significantly outperformed MPC in every case, with average MRR improvement produced by P-QAC, C-QAC, and N-QAC soaring to around 20%. The great enhancement of QAC performance suggests that users' context provides a valuable signal to generate good QAC ranking and that the ability to tailor query completions to a particular individual, rather than offering a unified completion list, provides a wealth of

Table 3 Mean reciprocal rank (MRR) results of different QAC models for prefix p consisting of 1–5 characters

p	MRR							
	MPC	P-QAC	C-QAC	N-QAC	LDA-QAC	ATM-QAC	CTM1-QAC	CTM2-QAC
1	0.0981	<u>0.3535</u>	0.3287	0.3365	0.3600	0.3534	$^{\triangle}$0.3670$^{\blacktriangle}$	$^{\triangle}$**0.3684**$^{\blacktriangle}$
2	0.1851	<u>0.4434</u>	0.4365	0.4413	0.4497$^{\triangle}$	0.4448	$^{\triangle}$0.4524$^{\blacktriangle}$	$^{\triangle}$**0.4553**$^{\blacktriangle}$
3	0.3165	<u>0.5246</u>	0.5183	0.5207	0.5280$^{\triangle}$	0.5258	$^{\triangle}$0.5313$^{\triangle}$	$^{\triangle}$**0.5348**$^{\blacktriangle}$
4	0.4249	0.5925	0.5947	<u>0.5963</u>	0.5920	0.5931	$^{\triangle}$0.5969$^{\triangle}$	$^{\triangle}$**0.5997**$^{\triangle}$
5	0.4921	0.6355	0.6364	<u>0.6381</u>	0.6318$^{\triangledown}$	0.6356	$^{\triangle}$0.6385$^{\triangle}$	$^{\blacktriangle}$**0.6407**$^{\triangle}$
Overall	0.2991	<u>0.5071</u>	0.4992	0.5035	0.5096	0.5077	$^{\triangle}$0.5145$^{\triangle}$	$^{\triangle}$**0.5170**$^{\blacktriangle}$

The best results of baselines and of all models in each row are underlined and boldfaced, respectively. The statistical significance of pairwise differences is determined by the t-test $^{\blacktriangle}$ for $p < 0.01$ and $^{\triangle}/^{\triangledown}$ for $p < 0.05$

opportunity to satisfy users' information needs. As to the three personalized baselines, P-QAC consistently gained advantage over C-QAC and N-QAC for short prefixes (1–3 characters). However, P-QAC falled shorter as prefixes got longer. Overall, P-QAC performed better than C-QAC and N-QAC with 0.79% and 0.36% MRR improvements, respectively. The reason behind this unexpected weak performance of learning-based C-QAC might be that it is essentially a reranking task of queries returned by MPC. Although its ranking accuracy evidently surpasses that of MPC, C-QAC cannot improve the results any further because there remain a large number of instances, in which the intended queries do not appear among the original rankings. The underperformance of N-QAC can be explained by the fact that the neural ranker generally needs abundant context to guarantee its effectiveness. However, as we discussed in Section 1, more than half of the users in the AOL dataset submit less than 10 queries. Therefore, the lack of sufficient data hinders the ability of N-QAC.

As to our cohort-based personalized QAC models, we observed that all of them outperformed P-QAC, C-QAC, and N-QAC in terms of overall MRR scores. This shows the potential of using cohort context in QAC ranking. Specifically, ATM-QAC shows minimum improvement over P-QAC. In most cases, ATM-QAC marginally outperformed P-QAC except at $p = 1$, where the MRR of ATM-QAC decreased slightly. Though encountering small drops by 0.05% at $p = 4$ and 0.37% at $p = 5$, LDA-QAC still exhibited generally better performance than P-QAC, with the overall MRR score increased by 0.25%. CTM1-QAC and CTM2-QAC substantially boosted the ranking effectiveness over P-QAC on every prefix length. All the improvement was statistically significant, which demonstrates the robustness of these two models. In particular, for all prefix lengths, the most significant gains of QAC models based on CTMs over P-QAC occurred in the short prefix category, with up to 1.35% and 1.49% improvements against P-QAC produced by CTM1-QAC and CTM2-QAC under $p = 1$, respectively. The reason is probably that when the lengths of prefixes are short, the collected cohorts' context contains useful information to filter out irrelevant completions and to improve ranking. However, when the lengths of prefixes get longer, the QAC problem becomes less challenging due to a reduction of the matched completions. Hence, it is hard to further improve the accuracy, even though the contextual information may be more targeted.

In comparison with MPC, which pays no attention to users' context as well as P-QAC, C-QAC, and N-QAC, which use only previously submitted queries from the current user, our cohort-based QAC models overcome those limitations altogether by adopting contextual information from cohorts. The remarkable MRR improvement over four baselines confirms the effectiveness of our proposals.

6.2 Cohort-based query auto-completion performance analysis

In this subsection, we take a closer look at the four cohort-based personalized QAC models and examine whether soft clustering techniques like CTM1 and CTM2 can further improve the QAC ranking accuracy (question 2). The results of significance tests of the improvement of CTM1-QAC and CTM2-QAC against LDA-QAC using a paired t-test are marked in the upper left-hand corner of CTM1-QAC and CTM2-QAC scores in Table 3.

It is encouraging to find that models based on CTMs produced better ranking results than models using the conventional topic models and the K-means algorithm in identifying cohorts, which shows the advantage of CTMs. Specifically, the MRR gap between LDA-QAC and ATM-QAC was subtle, although LDA-QAC marginally exceeded ATM-QAC on the overall ranking (0.19%); its effectiveness was not robust and showed moderate declines at $p = 4$ (0.11%) and $p = 5$ (0.38%). This may be explained by the fact that the K-means algorithm dilutes the difference between the values of $p(\boldsymbol{z}|\boldsymbol{u})$ computed by LDA and ATM. By contrast, both CTM1-QAC and CTM2-QAC outperformed ATM-QAC and LDA-QAC significantly and consistently in all cases. For instance, CTM2-QAC achieved the highest overall MRR score with up to 0.74% improvement over LDA-QAC and nearly 1% improvement over ATM-QAC. Presumably, a hard clustering method such as the K-means algorithm failed to reflect users' different interest; hence, the clustered cohorts may not be accurate. In contrast to conventional topic models, our CTMs are precisely designed for cohort discovery, and can capture users' multiple cohort memberships. Therefore, soft clustering techniques make cohort-based QAC ranking much better than hard membership assignment.

Interestingly, CTM2-QAC outperformed CTM1-QAC under all prefix lengths. This may indicate that compared with CTM1, the user clustering process described by CTM2 is closer to that in the actual situation.

6.3 Effect of contribution weight λ

To help answer question 3, we varied the contribution weight λ in Eq. (12) from 0 to 1 with a step length of 0.1, and examined the following effect on the overall performance of P-QAC as well as four variations of the cohort-based personalized QAC model.

Fig. 7 reveals that the average performance of the five personalized QAC rankers was quite sensitive to the change of λ. The MRR score of the five models reached its peak when λ reached 0.1 for P-QAC, 0.2 for LDA-QAC, 0.1 for ATM-QAC, 0.2 for CTM1-QAC, and 0.3 for CTM2-QAC. Once reaching its top, the ranking effectiveness of all models took a sharp downturn and ended at the same point when $\lambda = 1$, in which case ranking depended solely on query frequency.

As we can see from Fig. 7, all of the five rankers performed better under small λ values (0–0.3). On these circumstances, QAC models largely relied on context-based query similarity rather than frequency to rank completions. Moreover, when $\lambda = 0$, the MRR scores of all five models were significantly higher than those when $\lambda = 1$. In other words, the performance of QAC models depending only on query similarity was much better than that of models considering frequency alone. Therefore, we can conclude that query similarity is more important and beneficial than query frequency in terms of QAC ranking. Additionally, CTM1-QAC and CTM2-QAC notably outperformed the other three rankers under all λ settings, which once again verified the effectiveness and robustness of QAC models based on CTMs. We used the optimal λ values of these five personalized rankers in our experiments.

6.4 Impact of cohort number S

To answer question 4, we let cohort number S range from 2 to 20 with a step length of 2, and then evaluated the performance of CTM1-QAC and CTM2-QAC under different S settings. Fig. 8 shows the overall MRR scores of CTM1-QAC and CTM2-QAC given different cohort numbers.

From Fig. 8, we found that the MRR score of CTM1-QAC evidently fluctuated with the change of S. After reaching the highest point at $S = 18$, it went straight down to the bottom. In contrast, CTM2-QAC demonstrated insensitivity over different S values. The overall MRR score of CTM2-QAC stayed rather stable and also reached an optimal value at $S = 18$. We examined the values of $\mu_{a,s}$ and $\psi_{s,a}$ described in Section 3.2 and found that for each cohort s in CTM2, users with the top $\psi_{s,a}$ values were those who submitted hundreds of queries, regardless of the value of S. Consequently, the context used to calculate CoScore(q) in Eq. (13) almost stayed the same, whereas in CTM1, users with the highest $\mu_{a,s}$ values were not always possessing the richest contextual information or updating in accordance to S. This may explain why CTM1-QAC and CTM2-QAC presented two distinct situations with the same change of cohort number S. We set $S = 18$ for CTM1-QAC and CTM2-QAC in our experiments.

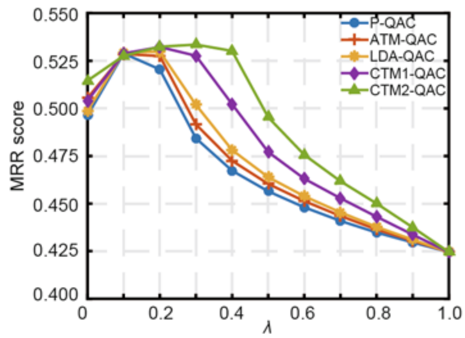

Fig. 7 Performance of the five personalized QAC rankers when varying the contribution weight λ from 0 to 1

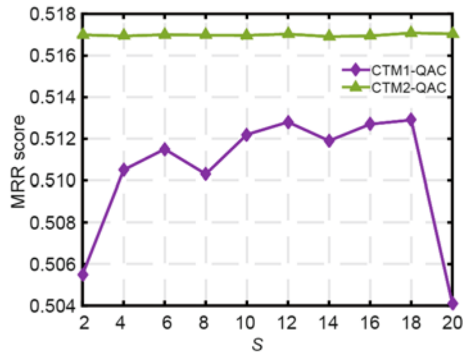

Fig. 8 Performance of CTM1-QAC and CTM2-QAC at various cohort numbers in cohort discovery

6.5 Addressing context sparseness

We considered question 5 by evaluating the MRR scores of the seven personalized QAC models discussed in this study under various context scenarios. We broke the ranked results into 10 groups according to the number of queries that each user submitted in the dataset. Fig. 9 depicts the results.

As we expected, on the whole, the greater the query volume, the better the ranking accuracy of the personalized QAC models. Evidently, N-QAC, C-QAC, and P-QAC were the weakest models. In particular, N-QAC produced the lowest MRR score in short context (4–10 queries). However, N-QAC achieved the highest score among all models for users who issued more than 12 queries. This is probably because a large amount of data is required to train an effective neural ranker. Meanwhile, C-QAC performed slightly better in short context (4–6 queries) and began to lose ground to P-QAC as context increased. This can perhaps be explained by the fact that the candidates generated by MPC limit C-QAC's benefit in addressing the short context problem. In comparison, the four cohort-based personalized models provided obviously higher MRR scores in the very limited context. Although ATM-QAC and LDA-QAC did not yield a strong performance gain over the three no-cohort baselines, CTM1-QAC and CTM2-QAC clearly dominated the other methods. Specifically, the two models based on CTMs manifested a significant and steady performance boost over P-QAC as queries accumulated. Even for users with little available context (query number ≤ 7), there was no abrupt drop of metric values. This may be due to the fact that soft clustering techniques can well capture users' topical interest and clustering cohorts, thus mitigating the sparsity issue. The above findings further underpin the superiority of cohort-based QAC models over the alternative approaches and indicate that the personalized rankers based on CTMs can appropriately tackle the context sparseness problem while remaining robust.

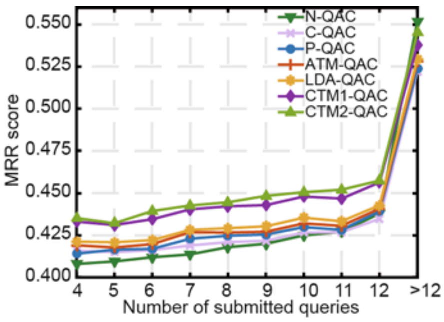

Fig. 9 Performance of seven personalized QAC rankers for users who submit different numbers of queries

7 Conclusions and discussion

QAC personalization offers the potential of significantly enhancing the users' search experience. Previous QAC models ignore individuals' differences in search intensions or face severe context sparseness challenges. We have alleviated the sparsity problem by exploiting contextual information collected from a particular user and his/her cohorts. We have clustered cohorts according to the user interest profile learned by topic models. Because conventional topic models cannot automatically group users without employing general clustering methods, we have proposed two distinct CTMs that act as soft clustering techniques, where each user has been assigned to multiple cohorts with a probability associated with each cohort. To achieve the greatest ranking improvement, query completions have been ordered by combining query frequency with query similarity based on the context obtained through cohort discovery modeling. Extensive experiments on a real-world query log demonstrated significant improvement over the competitive baselines, verifying the effectiveness of the proposed cohort-based personalized QAC models.

Our study can be developed in several directions. First, due to dataset limitations, we can access only users' preceding queries and documents as contextual information. It would be interesting to integrate extrinsic information, such as users' social networks and interactions with the system, in building users' profiles, to check if the information can further boost personalized ranking relevance. Second, because CTMs are general cohort discovery approaches, it is tempting to estimate their range of applicability to other web search areas, especially document retrieval and query suggestion. Finally, we aim to explore alternative cohort determination methods and evaluate them for QAC ranking task.

Compliance with ethics guidelines

Dan-yang JIANG and Hong-hui CHEN declare that they have no conflict of interest.

References

Bar-Yossef Z, Kraus N, 2011. Context-sensitive query auto-completion. Proc 20th Int Conf on World Wide Web, p.107-116. https://doi.org/10.1145/1963405.1963424

Blei DM, Ng AY, Jordan MI, 2003. Latent Dirichlet allocation. *J Mach Learn Res*, 3:993-1022.

Burges CJC, Svore KM, Bennett PN, et al., 2011. Learning to rank using an ensemble of lambda-gradient models. *J Mach Learn Res*, 14:25-35.

Cai F, de Rijke M, 2016a. Learning from homologous queries and semantically related terms for query auto completion. *Inform Process Manag*, 52(4):628-643. https://doi.org/10.1016/j.ipm.2015.12.008

Cai F, de Rijke M, 2016b. A survey of query auto completion in information retrieval. *Found Trends Inform Retr*, 10(4):273-363. https://doi.org/10.1561/1500000055

Cai F, Reinanda R, de Rijke M, 2016a. Diversifying query auto-completion. *ACM Trans Inform Syst*, 34(4), Artilce 25. https://doi.org/10.1145/2910579

Cai F, Liang SS, de Rijke M, 2016b. Prefix-adaptive and time-sensitive personalized query auto completion. *IEEE Trans Knowl Data Eng*, 28(9):2452-2466. https://doi.org/10.1109/TKDE.2016.2568179

Chen X, Zhou MY, Carin L, 2012. The contextual focused topic model. Proc 18th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.96-104. https://doi.org/10.1145/2339530.2339549

Fiorini N, Lu ZY, 2018. Personalized neural language models for real-world query auto completion. 16th Annual Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.208-215. https://doi.org/10.18653/v1/N18-3026

Gama J, Žliobaitė I, Bifet A, et al., 2014. A survey on concept drift adaptation. *ACM Comput Surv*, 46(4), Article 44. https://doi.org/10.1145/2523813

Hassan A, White RW, 2013. Personalized models of search satisfaction. Proc 22nd ACM Int Conf on Information and Knowledge Management, p.2009-2018. https://doi.org/10.1145/2505515.2505681

Jiang JY, Cheng PJ, 2016. Classifying user search intents for query auto-completion. Proc ACM Int Conf on the Theory of Information Retrieval, p.49-58. https://doi.org/10.1145/2970398.2970400

Li DF, Ding Y, Shuai X, et al., 2012. Adding community and dynamic to topic models. *J Inform*, 6(2):237-253. https://doi.org/10.1016/j.joi.2011.11.004

Li LD, Deng HB, Dong AL, et al., 2017a. Exploring query auto-completion and click logs for contextual-aware web search and query suggestion. Proc 26th Int Conf on World Wide Web, p.539-548. https://doi.org/10.1145/3038912.3052593

Li LD, Deng HB, Chen JH, et al., 2017b. Learning parametric models for context-aware query auto-completion via Hawkes processes. Proc 10th ACM Int Conf on Web Search and Data Mining, p.131-139. https://doi.org/10.1145/3018661.3018698

Mitra B, 2015. Exploring session context using distributed representations of queries and reformulations. Proc 38th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.3-12. https://doi.org/10.1145/2766462.2767702

Mitra B, Shokouhi M, Radlinski F, et al., 2014. On user interactions with query auto-completion. Proc 37th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.1055-1058. https://doi.org/10.1145/2600428.2609508

Morris MR, Horvitz E, 2007. Searchtogether: an interface for collaborative web search. Proc 20th Annual ACM Symp on User Interface Software and Technology, p.3-12. https://doi.org/10.1145/1294211.1294215

Neiswanger W, Wang C, Xing E, 2014. Asymptotically exact, embarrassingly parallel MCMC. Proc 13th Conf on Uncertainty in Artificial Intelligence, p.623-632. http://arxiv.org/abs/1311.4780v2

Park DH, Chiba R, 2017. A neural language model for query auto-completion. Proc 40th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.1189-1192. https://doi.org/10.1145/3077136.3080758

Pass G, Chowdhury A, Torgeson C, 2006. A picture of search. Proc 1st Int Conf on Scalable Information Systems, p.1. https://doi.org/10.1145/1146847.1146848

Shokouhi M, 2013. Learning to personalize query auto-completion. Proc 36th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.103-112. https://doi.org/10.1145/2484028.2484076

Shokouhi M, Radinsky K, 2012. Time-sensitive query auto-completion. Proc 35th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.601-610. https://doi.org/10.1145/2348283.2348364

Smyth B, Balfe E, Briggs P, et al., 2003. Collaborative web search. Proc 18th Int Joint Conf on Artificial Intelligence, p.1417-1419.

Steyvers M, Smyth P, Rosen-Zvi M, et al., 2004. Probabilistic author-topic models for information discovery. Proc 10th Int Conf on Knowledge Discovery and Data Mining, p.306-315. https://doi.org/10.1145/1014052.1014087

Teevan J, Morris MR, Bush S, 2009. Discovering and using groups to improve personalized search. Proc 2nd ACM Int Conf on Web Search and Data Mining, p.15-24. https://doi.org/10.1145/1498759.1498786

White RW, Chu W, Hassan A, et al., 2013. Enhancing personalized search by mining and modeling task behavior. Proc 22nd Int Conf on World Wide Web, p.1411-1420. https://doi.org/10.1145/2488388.2488511

Wu XD, Kumar V, Quinlan JR, et al., 2008. Top 10 algorithms in data mining. *Knowl Inform Syst*, 14(1):1-37. https://doi.org/10.1007/s10115-007-0114-2

Yan JY, Chu W, White RW, 2014. Cohort modeling for enhanced personalized search. Proc 37th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.505-514. https://doi.org/10.1145/2600428.2609617

Yin ZJ, Cao LL, Gu QQ, et al., 2012. Latent community topic analysis: integration of community discovery with topic modeling. *ACM Trans Intell Syst Technol*, 3(4), Article 63. https://doi.org/10.1145/2337542.2337548

Zhang A, Goyal A, Kong WZ, et al., 2015. adaQAC: adaptive query auto-completion via implicit negative feedback. Proc 38th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.143-152. https://doi.org/10.1145/2766462.2767697

Zheng GQ, Guo JW, Yang LC, et al., 2011. Mining topics on participations for community discovery. Proc 34th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.445-454. https://doi.org/10.1145/2009916.2009977