Exploring Query Auto-Completion and Click Logs for Contextual-Aware Web Search and Query Suggestion

Liangda Li†, Hongbo Deng‡∗, Anlei Dong#∗, Yi Chang◊∗, Ricardo Baeza-Yates§∗, and Hongyuan Zha¶⋆
†Yahoo Research, Sunnyvale, CA 94089
‡Google Inc., Mountain View, CA 94043
#Particle Media Inc., Santa Clara, CA 95054
◊Huawei Research America, Santa Clara, CA 95050
§NTENT, Carlsbad, CA 92008
⋆Software Engineering Institute, East China Normal University, Shanghai, China 200062
¶Georgia Institute of Technology, Atlanta, GA 30032
ldli@yahoo-inc.com, hbdeng@google.com, anlei@particle-inc.com,
yichang@acm.org, rbaeza@acm.org, zha@cc.gatech.edu

ABSTRACT

Contextual data plays an important role in modeling search engine users' behaviors on both query auto-completion (QAC) log and normal query (click) log. User's recent search history on each log has been widely studied individually as the context to benefit the modeling of users' behaviors on that log. However, there is no existing work that explores or incorporates both logs together for contextual data. As QAC and click logs actually record users' sequential behaviors while interacting with a search engine, the available context of a user's current behavior based on the same type of log can be strengthened from the user's recent search history shown on the other type of log. Our paper proposes to model users' behaviors on both QAC and click logs simultaneously by utilizing both logs as the contextual data of each other. The key idea is to capture the correlation between users' behavior patterns on both logs. We model such correlation through a novel probabilistic model based on the Latent Dirichlet allocation (LDA) model. The learned users' behavior patterns on both logs are utilized to address not only the application of query auto-completion on QAC logs, but also the click prediction and relevance ranking of web documents on click logs. Experiments on real-world logs demonstrate the effectiveness of the proposed model on both applications.

Keywords

Context, Latent Dirichlet allocation, Query auto-completion

1. INTRODUCTION

Exploring and modeling user's search behavior is very important to improve user's search experience for commercial search engines, and generally search behaviors can be collected in query auto-completion (QAC) and click logs. A QAC log records the detailed procedure while users issue queries into a search engine, with typical signals such as how fast a user types a query, which suggestion a user clicks, etc. On the other hand, a click log records how users behave on the search engine result page (SERP) of their issued queries. Typical click behaviors include which web document a user clicks, how long he/she stays on the document, and how many result pages a user scans. Essentially, modeling such user behaviors will be very useful for contextual-aware search tasks, which means previous search behaviors on both types of logs will somehow influence user's following search behaviors.

Recently several studies [24, 6, 5, 27, 1] explored contextual data to enhance Web search and query suggestion from different aspects. However, existing context-aware approaches on either query auto-completion or query suggestion only utilized a single type of log alone, while a critical fact is that QAC and click logs are different but closely correlated with each other. As shown in Figure 1, for each issued query, a QAC session first records how a user issues a query, which consists of the behaviors from the first keystroke a user typed in the search box towards the final submitted query; while a click session records how a user interacts with the SERP page, which consists of the behaviors after submitting that query. According to their temporal information, QAC and click logs can be naturally combined and ordered sequentially. Each query session refers to a combined session that starts with a QAC session and ends with a click session. Exploring contextual data from such query session will not only cover the same type of behaviors but also include the other type of behaviors that a user conducts in recent search history. For instance, different users may have different behaviors, such as how fast a user typed the previous query, and how many web pages a user clicked. Those search behaviors can be extracted and grouped as different patterns, so as to influence how the user submits the next query. Previously, most QAC logs only contain the query suggestion list along with the final typed query (last keystroke), which offers little extra information than click logs. Nowadays, high definition QAC logs are collected by commercial search engines [20], which contain the suggested query list for each keystroke and the associated users' interactions with a QAC engine. Thus combining QAC and click logs will offer much more complementary information for query prediction.

∗Work done while at Yahoo.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052593

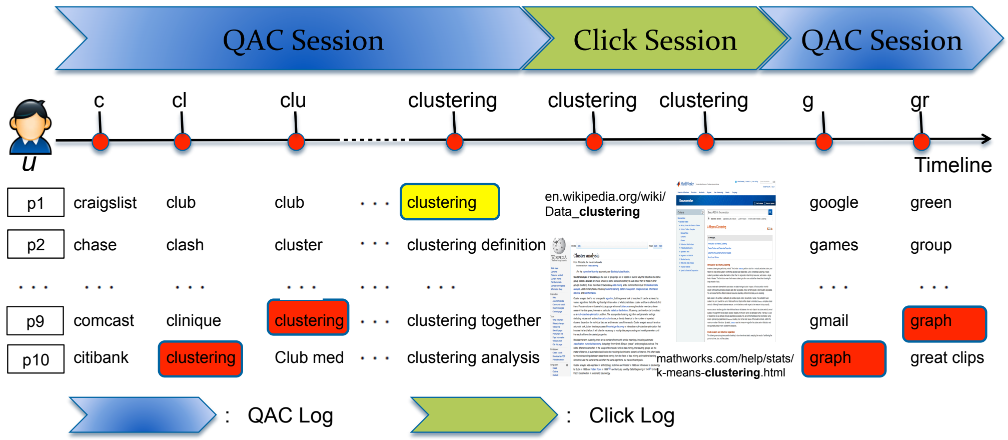


Figure 1: A Toy Example of how QAC and Click Logs Align in the Timeline. **Yellow tag highlights the query a user finally clicks, red tag highlights the user's intended query he/she doesn't click.**

Our goal is to effectively utilize the contextual data to model users' behaviors from both logs. The key idea is to cluster users' behaviors on QAC and click logs into several patterns, separately, and investigate the correlation between users' behavior patterns on QAC and click logs. We believe such correlation does exist, as users' search behaviors are usually consistent, which originate from users' personal search habits, preferences, interests, or instant circumstance. It is reasonable that a group of users may share similar behavior patterns. In addition, a user's QAC (or click) behavior pattern will be most likely correlated with a certain click (or QAC) behavior pattern. For instance, if a QAC log shows that a user types a query very fast, it is very likely that the user is familiar with the query. Then, in the following click log, the longer time the user landed on the SERP page may indicate the more relevant results presented to the user. One possible reason is that user likely will click the relevant results and check the detailed information which usually takes longer time. However, if there were no relevant results presented in the SERP page, the user might reformulate/re-issue a new query shortly which will start a new QAC session similar to previous query. Based on the learned correlation, given an inferred behavior pattern of a user on one type of log, we can leverage such information to accurately infer the user's following behavior pattern on the other type of log.

To capture such correlation, we propose a novel probabilistic model based on Latent Dirichlet allocation (LDA). Based on the likelihood of the co-occurrences of adjacent QAC behavior patterns and click behavior patterns, the model explores the conditional distribution of consequential behavior patterns given a certain behavior pattern of the other type. A mean-field variational inference algorithm is developed to estimate the membership of behavior patterns for two types of logs in each session. We evaluate the proposed model on real-world logs collected from a commercial search engine. We design experiments to evaluate the effectiveness of the learned behavior patterns on with the application of query auto-completion on QAC logs, and the prediction of web document clicking on click logs as well as the relevance ranking of web documents. Experimental results show that the proposed model achieves remarkable improvement on both applications over state-of-the-art approaches.

In a nutshell, our major contributions include: (1) This is the first study to explore two types of logs, QAC and click logs, simultaneously to model search behaviors. We utilize users' recent history on one type of log as the context for the other type of log. This new source of context data is demonstrated to mutually enhance behavior modeling on both types of logs. (2) We proposed a novel probabilistic model to capture the correlation between users' behavior patterns on QAC and click logs. The model is designed to study the conditional distribution of one type of behavior patterns given a certain preceding behavior pattern of the other type.

2. PROBLEM DEFINITION

In this section, we first introduce the concept of high-resolution QAC log, and analyze the relationship between QAC and click logs of a search engine. Then, we come up with methods for modeling users' behaviors on both logs simultaneously as the contextual data for each other.

2.1 A High-Resolution QAC Log

Traditionally, the search query log only includes the submitted query and its associated search results, while it does not contain the sequential keystrokes (prefixes) user typed in the search box, as well as their corresponding QAC suggestions. In order to better analyze and understand real users' behaviors, a high-resolution QAC log is introduced and analyzed in [20], which records users' interactions with a QAC engine at each keystroke and associated system respond in an entire QAC process. For each submitted query, there is only one record in a traditional search query log. However, in the high-resolution QAC log, each submitted query is associated with a **QAC session**, which is defined to begin with the first keystroke a user typed in the search box towards the final submitted query. The information recorded for each QAC session includes every keystroke a user entered, the timestamp and top-10 suggested queries corresponding to each keystroke, the anonymous user ID, and the final clicked query.

Let us take a toy example to briefly introduce how a user interacts with a QAC engine and makes the final click in an entire QAC session. As shown in the left part of Figure 1, the **QAC session** for the query "clustering" contains 10 keystrokes and each keystroke has a suggested query list of length 10[1]. A QAC session ends at the last keystroke when the user clicks a suggestion or hits enter/search to submit a fully typed query. Notice that although a user's actual click happens on a slot in the column of the last keystroke, the user intended query may appear in many slots in any columns. In this work, we leverage such a QAC log data to get better understanding of user's sequential behavior, which can provide useful information for predicting the user's following behavior.

[1]We experiment with real-world QAC logs where $D = 10$.

Table 1: User Behaviors on QAC and Click Logs

Log Type	Behavior/Feature	Description
QAC Log	Typing Speed	Average typing speed at keystrokes in a QAC session.
	Type Speed Standard Deviation	The standard deviation of typing speed at keystrokes in a QAC session.
	Intent Position	The average position of the appearance of queries satisfying users' search intent in a QAC session.
	Typing Completion Ratio	The percentage of entered keystrokes of the submitted query.
	Typing Completion	Whether a user finish typing the entire query or clicks some suggestions.
	Time Duration	The time duration of the current QAC session.
	Highest Non-Click Position	The highest position of the appearance of queries satisfying users' search intent but the user does not click in a QAC session. Here those queries with the same content as the final clicked query are viewed as satisfying users' intent.
Click Log	Click Number	The total number of clicks on the returned web documents of the current query.
	Dwell Time	The average time between the current click and the next click in the current click session.
	Click Position	The average position of clicks in the current click session.
	Time Duration	The time duration of the current click session.
	Click Speed	The number of clicks divided by the time duration of the current click session.
	Scanned Pages	The number of result pages user scanned for the current query.
	Time Interval	The time interval between the current click session and the next QAC session.
	Search Time	The time interval between a user's query submission and his/her first click of web documents in the current click session.

2.2 Relationship between QAC and Click logs

For each query, two types of behaviors are recorded by search engine logs. One is the above high-resolution QAC log, which includes the typed keystrokes and their suggested queries before submitting a query; the other is the click log, which includes the web document clicks after submitting a query. Figure 1 shows a toy example of QAC and click logs that align in the timeline. We can observe that the QAC session of a query is followed by the click session of that query, and that click session is followed by another QAC session of the next query. Such sequential behaviors indicate the promising opportunity of exploring appropriate relationship between QAC and click logs. Although the user's behaviors on QAC and click logs are of different types, they imply the same underlying relationship between the user and his/her issued query, such as whether the issued query satisfies the user's intent, and how familiar the user is with the issued query or the domain that query belongs to. For instance, if a user is familiar with the issued query, in QAC log, he/she may type the query very fast. Then in click log, if the SERP page provides many relevant results, the user may take long time to click and check some relevant results in more details; however, if the SERP page does not provide relevant results, the user may reformulate a new query shortly which will start a new QAC session similar to previous query.

Moreover, user's search behaviors on one type of log can be used as the contextual data for the other type of log across different query sessions, since users generally behave consistently in adjacent time slots. For instance, according to the click log, if a user's behaviors indicate he is very familiar with the current query, then similar behavior likely will be observed in the QAC session of the next query; if the issued query is under the same topic, the user will probably type the following query fast as well.

In order to quantitatively capture user behaviors on QAC and click logs, we propose a set of features as shown in Table 1. Among features of QAC behaviors, we expect "Type Speed Standard Deviation" to reflect the stability of a user's typing speed. A user who examines his/her intended queries from the suggestion list from time to time may hardly maintain a stable typing speed, even if the user has good typing skills. On the contrary, a user who plans to type the entire query without clicking a suggestion may illustrate a stable typing speed. "Typing Completion" is designed to show whether a user prefers typing than clicking suggestions. Among the features of click behaviors, "Search Time" is defined to be how fast a user can find his/her intended web documents after submitting a query. Notice that users' behaviors on different types of logs are not independent. On the QAC log, an experienced user usually spend less time to complete a QAC session than an unexperienced user, i.e., has a small "Time Duration". While on the click log, he/she is very likely to make his/her first click after only a short while, i.e., a small feature value for "Search Time". A user who tends to trust the results of search engines may miss the QAC behavior feature "Typing Completion", and owns a higher value of the click behavior feature "Click Number". Thus our designed QAC and click behavior features are somehow related, and we will design appropriate models to capture such relationship in the following sections.

2.3 Contextual Topic Distribution

To detect user behavior patterns from logs, we choose a widely used graphical model, latent Dirichlet allocation (LDA) [4], which has been proven to be effective in topic discovery by clustering words that co-occur in the same document into topics. First we consider how to use LDA to cluster user behaviors based on one single type of log only (either QAC log or click log). One straightforward idea is to treat each user's query sequence as a document, and cluster user behaviors that co-occur frequently in the same query sequence into topics, since each user maintains certain behavior patterns in query submission, and different groups of users prefer different behavior patterns. Our LDA model assumes K behavior patterns lie in the given query sequences, and each user m is associated with a randomly drawn vector π_m, where $\pi_{m,k}$ denotes the probability that the user behavior in a query session of user m belongs to behavior pattern k. For the n-th query in the query sequence of user m, a K-dimensional binary vector $Y_{m,n} = [y_{m,n,1}, \ldots, y_{m,n,K}]^T$ is used to denote the pattern membership of the user behavior in that query session. One challenge we encounter in the inference of

Table 2: Major Notations

Symbol	Description
K	The number of QAC behavior patterns.
K'	The number of click behavior patterns.
d	Users' click behaviors.
ω	Users' QAC behaviors.
Y	Users' click behavior pattern membership.
Z	Users' QAC behavior pattern membership.

pattern membership Y is that, user's choice of behavior patterns in each query session is not only decided by users' own preferences of behavior patterns, but also influenced by the context of the current query session.

To model the influence of the context on user's choice of the behavior pattern in the current query session, we assume user's preference of behavior patterns depend on the context, rather than the user alone. That is to say, a "document" in the LDA model does not contain the user behaviors in all query sessions of a user, but only the behaviors in those query sessions that the user conducts under the same status, for instance, in the same mood, or sharing the same topic. In this paper, we focus on studying how using one type of log as context can benefit the user behavior modeling on the other type of log. Thus, in the detection of user behavior patterns in one type of log, we define each status mentioned above to be the user behavior pattern in the other type of log. That is to say, instead of building a behavior pattern distribution π_m for each user m, and accordingly draw the user's behavior in each query session of that user, we construct a QAC (or click) behavior pattern distribution π_k for each click (or QAC) pattern. Then after we inferred the pattern membership of a user's behavior on click (or QAC) log, we obtain the corresponding QAC (or click) behavior pattern distribution, and in the next QAC (or Click) session, draw the QAC (or click) pattern accordingly.

2.4 Contextual-LDA Model

Let us consider a typical scenario where M users issue M corresponding query sequences. For each query n, we have the QAC log records a user's behavior $\omega_{m,n}$ in the QAC engine before submitting the query and the click log records a user's behavior $d_{m,n}$ on returned web documents after the query is issued. We assume that K QAC behavior patterns exist in the QAC log, and K' click behavior patterns exist in the click log.

Finally, we present our generative model as follows:

- For each click behavior pattern k', draw a K dimensional membership vector $\pi_{k'} \sim \text{Dirichlet}(\alpha)$.
- For each QAC behavior pattern k, draw a K' dimensional membership vector $\pi'_k \sim \text{Dirichlet}(\alpha')$.
- For each click behavior pattern k', draw a T' dimensional distribution vector $\theta'_{k'}$.
- For each QAC behavior pattern k, draw a T dimensional distribution vector θ_k.
- For the n-th click session issued by user m,
  - Draw the user's click session behavior $d_{m,n} \sim \text{Gaussian}(\theta'_{Y_{m,n}})$;
  - Draw the user's next QAC behavior pattern membership $Z_{m,n+1} \sim \text{Multinomial}(\pi_{Y_{m,n}})$;

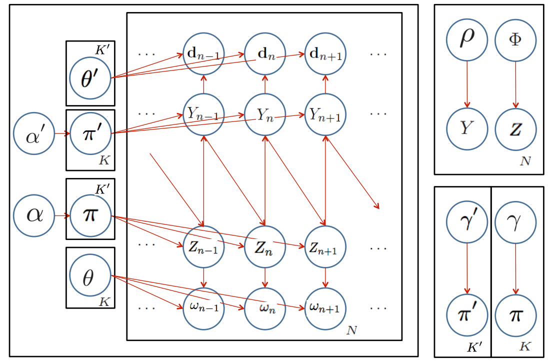


Figure 2: Graphical model representation of Contextual-LDA and the variational distribution that approximates the likelihood.

- For the n+1-th QAC session issued by user m,
  - Draw the user's QAC session behavior $\omega_{m,n+1} \sim \text{Gaussian}(\theta_{Z_{m,n+1}})$;
  - Draw the user's next click behavior pattern membership $Y_{m,n+1} \sim \text{Multinomial}(\pi'_{Z_{m,n+1}})$;

Here T is the number of features of QAC behaviors, and T' is the number of features of click behaviors. Also notice that in the generative process of the proposed model, user's QAC behavior pattern membership Z and click behavior pattern membership Y mutually infer each other in an interleaved manner. In specific, for a user m, the QAC behavior pattern membership $Z_{m,n}$ in the n-th QAC session decides his/her click behavior pattern membership $Y_{m,n}$ in the click session under the same query, and in the following this click behavior pattern membership $Y_{m,n}$ decides his/her QAC behavior pattern membership $Z_{m,n+1}$ in the QAC session of the next query, i.e., the n+1-th QAC session. In the Gaussian distribution, θ defines the mean and the covariance matrix is identity. We name the proposed model Contextual-LDA.

Under our Contextual-LDA model, the joint probability of data $D = \{D_m\} = \{\{d_{m,n}\}_{n=1}^{N_m}\}$, $\omega = \{\{\omega_{m,n}\}_{n=1}^{N_m}\}$, and latent variables $\{Y, Z\}$ can be written as follows:

$$
\begin{aligned}
&p(D, \omega, \pi, \pi', Y, Z|\alpha, \alpha', \theta, \theta') \\
=&\prod_m \prod_n P(d_{m,n}|Y_{m,n}, \theta') P(\omega_{m,n}|Z_{m,n}, \theta) \\
&\times \prod_m \prod_n P(Z_{m,n}|Y_{m,n-1}, \pi) P(Y_{m,n}|Z_{m,n}, \pi') \\
&\times \prod_k P(\pi_k|\alpha) \prod_{k'} P(\pi'_{k'}|\alpha').
\end{aligned}
$$

3. INFERENCE

In this section, we derive a mean-field variational Bayesian inference algorithm for our proposed Contextual-LDA model.

3.1 Variational Inference

Under Contextual-LDA model, given observations of both the high definition QAC log $D = \{D_m\} = \{\{d_{m,n}\}_{n=1}^{N_m}\}$ and the click log $\omega = \{\{\omega_{m,n}\}_{n=1}^{N_m}\}$, the log-likelihood for the complete data is given by $\log P(D, \omega|\alpha, \alpha', \theta, \theta')$. Since this true posterior is hard to infer directly, we turn to variational methods [3], whose main idea is to posit a distribution over the latent variables with variational parameters, and find the settings of the parameters so as to make the distribution close to the true posterior in Kullback-Leibler (KL) divergence. In Figure 2, the right-hand part shows the variational distribution that approximates the data likelihood. Our paper chooses to introduce a distribution of latent variables q specified as the mean-field fully factorized family as follows:

$$q(Y, Z, \pi, \pi' | \rho, \phi, \gamma, \gamma') = \prod_m \prod_n q_1(Y_{m,n}|\rho_{m,n}) q_1(Z_{m,n}|\phi_{m,n}) \prod_{k'} q_2(\pi_{k'}|\gamma_{k'}) \prod_k q_2(\pi'_k|\gamma'_k),$$

where q_1 is a multinomial, q_2 is a Dirichlet, and $\{\Phi, \rho, \gamma, \gamma'\}$ are the set of variational parameters. We optimize those free parameters to tight the following lower bound $\mathcal{L}'$ for our likelihood:

$$\log p(D, \omega|\alpha, \alpha', \theta, \theta') \geq E_q[\log p(D, \omega, \pi, \pi', Y, Z|\alpha, \alpha', \theta, \theta')] - E_q[\log q(Y, Z, \pi, \pi'|\rho, \phi, \gamma, \gamma')]. \quad (1)$$

Under a coordinate descent framework, we optimize the lower bound as in Eqn (1) against each variational latent variable and the model hyper-parameter. For variational latent variables, we have the following process

- update rules for γ's as:
  $\gamma_{k',k} = \alpha_k + \sum_m \sum_n \phi_{m,n+1,k}\rho_{m,n,k'}$;
- update rules for γ''s as:
  $\gamma'_{k,k'} = \alpha'_{k'} + \sum_m \sum_n \phi_{m,n,k}\rho_{m,n,k'}$;
- update rules for ρ's as:
  $\rho_{m,n,k'} \propto \exp\Big(-\frac{1}{2\sigma^2}\sum_{m,n}(d_{m,n} - \theta'_{m,n,k'})^2 + \sum_k \phi_{m,n+1,k}[\Psi(\gamma_{k,k'}) - \Psi(\sum_{k'} \gamma_{k,k'})] + \sum_k \phi_{m,n,k}[\Psi(\gamma'_{k',k}) - \Psi(\sum_k \gamma'_{k',k})]\Big)$,
- update rules for ϕ's as:
  $\phi_{m,n,k} \propto \exp\Big(-\frac{1}{2\sigma^2}\sum_{m,n}(\omega_{m,n} - \theta_{m,n,k})^2 + \sum_{k'} \rho_{m,n-1,k'}[\Psi(\gamma_{k,k'}) - \Psi(\sum_{k'} \gamma_{k,k'})] + \sum_{k'} \rho_{m,n,k'}[\Psi(\gamma'_{k',k}) - \Psi(\sum_k \gamma'_{k',k})]\Big)$,

3.2 Learning

We use a variational expectation-maximization (EM) algorithm [9] to compute the empirical Bayes estimates of the LDA hyper-parameters α and α' in our Contextual-LDA model. This variational EM algorithm optimizes the lower bound as in Eqn (1) instead of the real likelihood, it iteratively approximates the posterior by fitting the variational distribution q and optimizes the corresponding bound against the parameters.

In updating α, we use a Newton-Raphson method, since the approximate maximum likelihood estimate of α doesn't have a closed form solution. The Newton-Raphson method is conducted with a gradient and Hessian as follows:

$\frac{\partial \mathcal{L}'}{\partial \alpha_k} = K(\Psi(\sum_k \alpha_k) - \Psi(\alpha_k)) + \sum_k(\Psi(\gamma_{k,k}) - \Psi(\sum_k \gamma_{k,k}))$,

$\frac{\partial \mathcal{L}'}{\partial \alpha_{k_1}\alpha_{k_2}} = N(\mathbb{I}_{(k_1=k_2)}\Psi'(\alpha_{k_1}) - \Psi'(\sum_k \alpha_k))$.

Similar update rules can be derived for α'. Here, Ψ is the digamma function.

On the other hand, to obtain the approximate maximum likelihood estimation of parameters describing QAC and click behavior patterns θ and θ', we optimize the lower bound as in Eqn (1) against each parameter, and update θ and θ' independently with closed-form solutions as follows:

$$\theta'_{k'} = \frac{\sum_{m,n}\rho_{m,n,k'}d_{m,n}}{\sum_{m.n}\rho_{m,n,k'}}, \qquad \theta_k = \frac{\sum_{m,n}\phi_{m,n,k}\omega_{m,n}}{\sum_{m.n}\phi_{m,n,k}};$$

Table 3: Log Predictive Likelihood on Real-world Data
The platform "All" means the combination of data from both PC and mobile platforms. Smaller value means better performance.

Platform	Contextual-LDA	HMM	LDA
PC	-253.69	-286.55	-295.21
Mobile	-223.82	-252.30	-257.91
All	-241.74	-272.85	-280.29

In our mean-field variation inference algorithm, the computational cost of inferring variational variables is $O((\sum_m N_m)KK')$, where N_m is the number of sessions of user m, K is the number of QAC behavior patterns, and K' is the number of click behavior patterns. The computational cost of the estimation of LDA hyper-parameters is $O(KK')$. The computational cost of the estimation of behavior patterns is $O(\sum_m N_m(K + K'))$, Thus the total computational cost of our algorithm is $O((\sum_m N_m)KK')$. Since we can control the value of KK' by limiting the number of QAC and click behavior patterns, this total computational cost can be viewed as linear to the number of queries in the entire log.

4. EXPERIMENTS

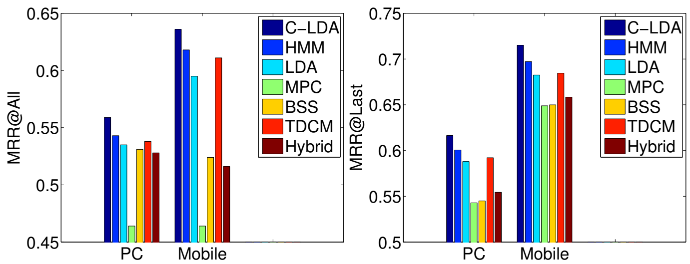
In this section, we evaluated our Contextual-LDA model on real-world data sets, and compared the performance with various alternative probabilistic models, as well as the state-of-the-art QAC methods and click models. First, we conducted a series of experiments to measure the fitness of the proposed model on real-world query logs together with alternative probabilistic models. Then, we 1) present how the proposed model addresses the real-world applications including query auto-completion (QAC) and learning to rank, respectively; 2) evaluate the performance on real-world data; and 3) compare it with state-of-the-art QAC methods and click models, separately. For all the experiments, besides measuring the performance of the above real-world applications by corresponding metrics that are widely used, we also perform significance tests using paired t-test with 0.05 as the p-value threshold.

4.1 Real-world Data

We conducted extensive experiments on a real-world QAC log and the corresponding click log collected from Yahoo. This data set contains QAC and click logs collected from May 2014 to July 2014, which include a sample of 7.4 million query sessions from about 40,000 users over a 3-month period. As defined in previous sections, each query session refers to a combined session that starts with a QAC session and ends with a click session. We randomly selected a subset of active users who submitted over 500 query sessions during this period, and collected their corresponding search activities, including the anonymized user ID, query string, timestamp, and the clicked URL. As a result, we collected 3,954 users with 2.6 million queries, and their activities span from 22 days to 3 months. According to the platform each query session belongs to, we separate the entire data set into two subsets. One is PC, which contains 1.6 million query sessions, while the other is mobile phone, which contains 1.0 million query sessions. On query auto-completion experiments, we evaluate the performance on those separate subsets, since users' behaviors on QAC engines are significantly influenced by the platform they use.

4.2 Model Fitness

To evaluate the fitness of the proposed model on real-world QAC and click logs, we use this series of experiments to measure the performance of Contextual-LDA by the log predictive likelihood



(a) MRR@All (b) MRR@Last

Figure 3: Performance of Query Auto-Completion. In the figure we use C-LDA to denote Contextual-LDA

on the observed query logs, and compare it with two alternative LDA-based probabilistic models that are also capable of learning QAC and click behavior patterns.

LDA: This model uses a normal LDA to learn users' behavior patterns on QAC and click logs, separately. No contextual data is utilized in the process of pattern learning.

HMM: This is a hidden Markov model that builds a hidden state for each QAC and click session, and takes the users' behaviors in each QAC and click session as observation. The transition matrix between hidden states learned by the HMM model is expected to capture the effect of using one type of log as the contextual data of modeling the other type of log.

Table 3 shows the log predictive likelihood on query sessions falling in the final 10% of the total number of the observed query sequences. The same split of training and test data will be used in all following series of experiments. To avoid overfitting issues, we adopt a k-fold cross validation strategy, and select the optimal number of QAC behavior patterns K and the optimal number of click behavior patterns K' that maximize the log likelihood on the validation data. According to Table 3, Contextual-LDA outperforms the two alternative probabilistic models on all data sets. Contextual-LDA and HMM fits real-world data better than LDA. This illustrates the effectiveness of using one type of log as the contextual data for modeling the behaviors on the other type of log. Contextual-LDA performs better than HMM, which shows that the proposed model can better utilize the contextual data to help the behavior modeling through appropriate usages of relationship between QAC and click logs. Moreover, the advantage of the proposed model on mobile data over those two alternative probabilistic models, especially LDA, is more significant than that on PC data. Such phenomenon implies that search engine users' QAC and click behaviors on mobile phones are more closely correlated than on PCs, and Contextual-LDA can better capture such correlation.

4.3 Query Auto-Completion

In this series of experiments, we show how to utilize the pattern membership inferred by the proposed model to enhance the performance of query auto-completion. We design a new QAC method based on a two-dimensional click model (TDCM) [20], which is known to be the first model proposed for solving the QAC task using high definition QAC logs. Instead of learning a TDCM model on the entire QAC log, our new method separates the log according to the behavior pattern membership in each QAC session, and learns separate TDCM models on each subset, under the same experimental setting (same features, etc) as in [20]. To justify how effective appropriate search patterns benefit solving the QAC task, we compare the performance of the above method with those methods adopting a similar strategy using the QAC behavior patterns learned by LDA and HMM. We compare the performance with several state-of-the-art QAC algorithms, where two of them are context-aware QAC algorithms:

MPC [1, 25]: This method, named MostPopularCompletion, is a widely used baseline in Query Auto-Completion, and employed as one main feature in many QAC engines.

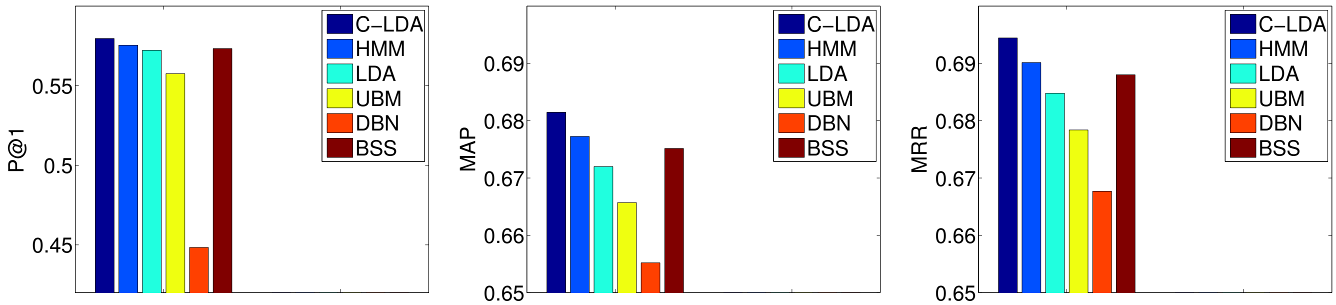
BSS [28]: This Bayesian Sequential State model uses a probabilistic graphical model to characterize the document content and dependencies among the sequential click events within a query with a set of descriptive features. This is a content-aware model, which is able to predict unobserved prefix-query pairs.

TDCM [20]: This is a two-dimensional click model which emphasizes two kinds of user behaviors. It consists of a horizontal model which explains the skipping behavior, and a vertical model that depicts the vertical examination behavior. It is the first work that utilizes high definition QAC logs.

Hybrid [11]: This is a context-sensitive query auto completion algorithm, which outputs the completions of the user's input that are most similar to the context queries. The similarity is measured by representing queries and contexts as high-dimensional term-weighted vectors and resorting to cosine similarity.

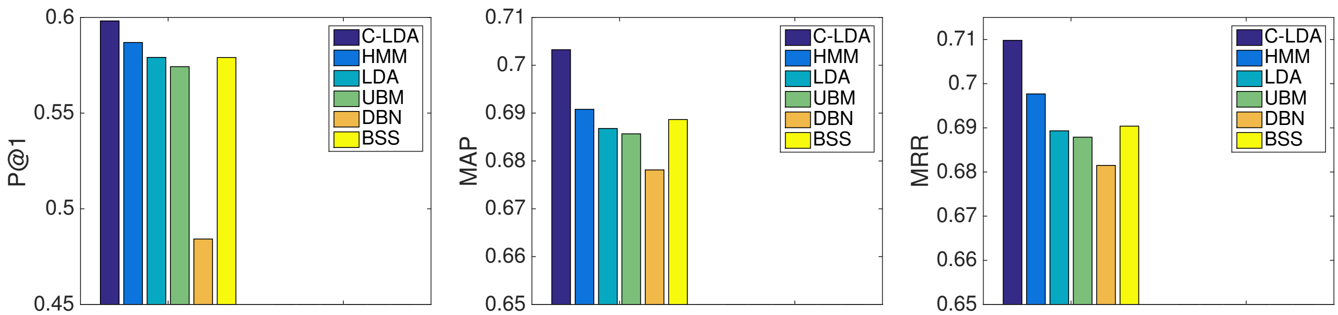
We employ the Mean Reciprocal Rank (MRR) as the relevance measurement, which is a widely used evaluation metric in measuring QAC performance [1, 25, 20], $MRR = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{\mathrm{rank}_q}$, where Q is the set of queries a user finally submitted, and rank_q denotes the rank of the query q in the suggested query list. Since our experiments are conducted on high-resolution QAC data, we report both the average MRR score of all keystrokes (denoted as MRR@ALL), and the average MRR of the last keystroke only (denoted as MRR@Last), since this is the keystroke where the user's click occurs. Notice that existing works which didn't make use of high-resolution QAC logs usually used the MRR of the last keystroke to measure their performance.

Figure 3 compares Contextual-LDA with alternative probabilistic models, and state-of-the-art QAC algorithms on real-world data sets. We can find that Contextual-LDA outperforms all compared approaches. It improves over the second best method by up to 5%. And the differences between the proposed model and those baselines are statistically significant. TDCM performs the best among all state-of-the-art QAC methods, which demonstrates that high definition QAC logs provide rich additional information for the modeling of users' interactions with QAC engines than normal QAC logs. Contextual-LDA performs better than HMM and LDA, which shows the importance of appropriate modeling of user behaviors, and appropriate behavior patterns play a very positive effect in solving QAC tasks. HMM performs better than TDCM and LDA, since HMM utilizes the contextual relationship between QAC and click logs, while LDA models user behaviors on each log separately, and similarly TDCM only focuses on users' behaviors on high definition QAC logs alone in solving the query auto-completion tasks. BSS and Hybrid generally perform better than MPC, which demonstrates the effectiveness of using contextual data for user behavior modeling and the prediction of suggestions in query auto-completion. Meanwhile, we find that when measured by MRR@All the advantage of the proposed model over those baselines are very obvious and statistically significant. Comparing with the performance using all the keystrokes and last keystroke only, the advantages of the proposed model when measured by MRR@All are ever more significant than that measured by MRR@Last. It indicates that the proposed model can recommend



(a) P@1 (b) MAP (c) MRR

Figure 4: Performance Comparison of Prediction of Clicks in Web Documents. In the figure we use C-LDA to denote Contextual-LDA.



(a) P@1 (b) MAP (c) MRR

Figure 5: Performance Comparison of Learning to Rank based on RankSVM with Different Training Signals. In the figure we use C-LDA to denote Contextual-LDA.

user intended queries higher when only a few keystrokes have been typed, which may due to rich information provided by the context from the other type of log in the preceding click session.

4.4 Click Prediction and Learning to Rank

In this part of experiments, we show how to utilize the pattern membership inferred by the proposed model to enhance the prediction of click/relevance on SERP page. Meanwhile, we provide an appropriate ranking of those candidate documents that satisfy users' intent. We design a new click model based on BSS, which is known to be an efficient click model. Rather than learning a BSS model on the entire click log, this proposed method separates the log according to the behavior pattern membership in each click session, and learns separate BSS models on each subset, under the same experimental setting (same features, etc) as in [28]. To justify how effectively appropriate search patterns improve the click prediction, we rank the candidate documents based on the estimated relevance given by a click model, and compare the ranking result against the recorded user clicks to see whether we can rank those candidate documents that satisfy users' intent higher than other candidate documents. We compare the performance of the above method with existing approaches using a similar strategy, i.e., partitioning click logs based on the click behavior patterns learned by LDA and HMM. We evaluate the performance with several state-of-the-art click models as follows, including a context-aware click model:

UBM [11]: This User Browsing Model proposes a number of assumptions on user browsing behavior that allows the estimation of the probability of observing a document. It depends on statistical counting of query-document pairs, thus unable to predict unseen query-document pairs.

DBN [7]: This Dynamic Bayesian Network model provides unbiased estimation of the relevance from the click logs. This model also relies on the counting of query-document pairs.

BSS [28]: This Bayesian Sequential State model uses a probabilistic graphical model to characterize the document content and dependencies among the sequential click events within a query with a set of descriptive features. This is a content-aware model which is able to predict unobserved query-document pairs.

Figure 4 compares the proposed model with alternative probabilistic models and state-of-the-art click models. We find that the proposed model performs the best among all compared approaches, and the differences are all statistically significant. The advantage of the proposed model over BSS provides a clear evidence that the proposed model can appropriately model the click behaviors of search engine users, which contributes to an more accurate click prediction. Contextual-LDA's advantage over HMM and LDA illustrates the importance of appropriate click behavior patterns in predicting the clicks of web documents. Both Contextual-LDA and HMM outperforms BSS, which may due to the fact that BSS only utilizes the contextual information from the click log, while the first

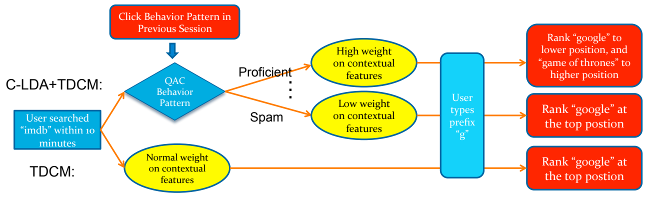


Figure 7: Case Study: How a Contextual-LDA based TDCM model performs better in a QAC session than a normal TDCM model.

two methods also utilize the QAC log as the context of click behavioral modeling, which not only contains rich information, but also occurs in more recent timestamps. Meanwhile, BSS performs better than UBM, DBN, and LDA, which shows the effectiveness of using contextual data for click modeling.

In addition, we conduct another series of experiments to justify how the proposed model benefit a learning to rank task. Here we use the estimated relevance given by a click model as labels to extract ranking-preference pairs of candidate documents for training a learning-to-rank algorithm, which we select the pairwise RankSVM [16]. In particular, we rank the documents returned when given a certain query based on their predicted relevance from a click model, and categorize the top ranked document as positive and others as negative. We extract the preference pairs according to those relevance labels under each query and train them by a RankSVM model. Figure 5 compares the performance of RankSVM models trained by different relevance signals provided by the proposed model and above baselines. From the figure, we can find that the training signals extracted from the Contextual-LDA based click model result in better ranking performance of RankSVM than those extracted from the rest click models. Moreover, we notice that the advantage of the proposed model over those baselines measured by MAP and MRR is much greater than that measured by P@1. The main reason can be that the proposed model is more flexible in learning the relative order of candidate documents, since it takes contextual information into consideration.

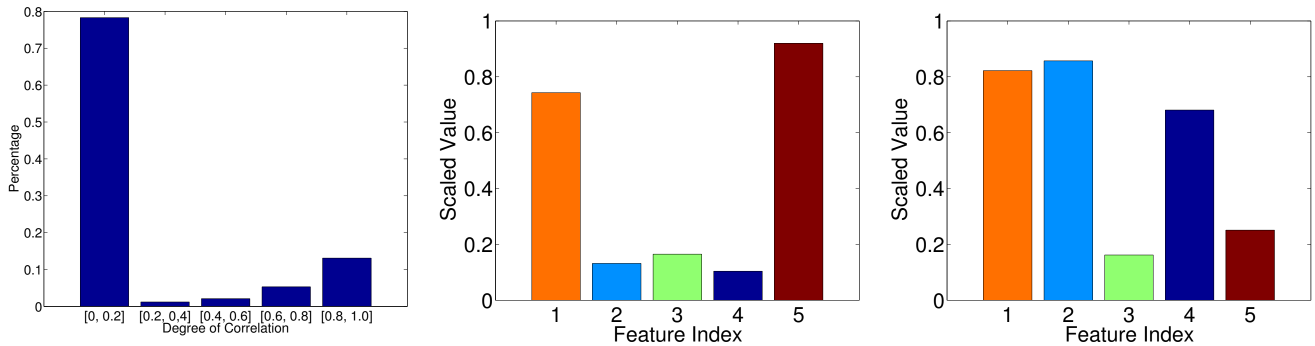
Correlation between Behavior Patterns in QAC and Click Logs. In this part, we analyze the correlation between behavior patterns in QAC and click logs based on the inferred conditional pattern membership distribution, and then try to show a few examples of highly correlated QAC behavior patterns and click behavior patterns. First, we estimate the degree of correlation between QAC and click behavior patterns based on the inferred conditional pattern membership distributions $\{\pi_{k'}\}$ and $\{\pi'_k\}$. Here we take $\pi_{k',k}$ and $\pi'_{k,k'}$ as the directed partial correlations of the k-th QAC behavior pattern and the k'-th click behavior pattern. Then, we statistically count the number of pattern pairs whose degree of correlation falls in the range of [0, 0.2], [0.2, 0,4], [0.4, 0.6], [0.6, 0.8], [0.8, 1.0], separately. Finally, we show the percentage of pattern pairs in each bin in Figure 6(a). From the figure, we can find that between most pairs of behavior patterns, the degree of correlation is very small. And among the rest pattern pairs with significant correlations, between most of them the corresponding degrees of correlation are larger than 0.8, i.e., there is a one to one mapping between those pattern pairs. (Notice that since the inference of π and π' ensures that the degrees of correlations of one pattern with others sum to 1, if the degree of correlation between a QAC pattern and a certain click pattern is larger than 0.8, the QAC pattern's correlation with all other click patterns will be very small (<0.2).) Such phenomenon shows that a lot of users retain the same behavior mode for quite a while.

Figure 6(b) and (c) show an example pair of QAC and click behavior patterns between which the degree of correlation is larger than 0.8. From the QAC behavior pattern shown in (b), we find that 1) the user's typing speed is very fast; 2) the time cost of completing a QAC session is very small; 3) the user does not like to click suggested queries even if they satisfy his/her search intent and ranked at top positions; 4) the user types keystrokes in a consistent speed; and 5) most of the time, the user types his/her intended query completely instead of stopping to click the suggestions returned by QAC engines. Based on the above behaviors, we can conclude that this is probably a user who is proficient in searching or his/her intended topic. From the click behavior pattern shown in (c), we find that 1) the user clicks a lot of web documents returned by the search engine; 2) the user spends a lot of time in viewing the clicked web documents, 3) the user scans several pages of results, and 4) it does not take a lot of time for the user to find his/her intended web documents after submitting the query. Those behaviors also illustrate that this is a proficient user. Thus the correlation captured by the proposed model from real-world QAC and click logs is appropriate and meaningful. Based on such correlation, the proposed model can more accurately infer the behavior patterns on one type of log according to those on the other type of log in preceding sessions.

Figure 7 shows an example that illustrates how a Contextual-LDA based TDCM model recommends a user queries that better satisfy user intent in a QAC session than a normal TDCM model. Before this QAC session, which the user finally clicks "game of thrones", he just searched "imdb" within 10 minutes. Based on the behavior pattern in the previous click session, the proposed Contextual-LDA model infers that the QAC behavior pattern is "Proficient" as shown in Figure 6(b). When the user type a prefix "g", since the TDCM model trained under "Proficient" pattern gives contextual features large weights, it consequently ranks "google" to a lower position, and ranks "game of thrones" a higher position. While a normal TDCM model trained on the entire log just gives contextual features normal weights, and still ranks "google" at the top position. Our Contextual-LDA model enables the usage of appropriate TDCM models under various contextual scenarios, which consequently better satisfy user intent than a generalized TDCM model in QAC tasks.

5. RELATED WORK

Contextual Search. Contextual search is heavily researched in literature and is explored from different angles. A large portion of a recent comprehensive survey on contextual search is devoted to the study of personal interest from interaction, content, social, and geographical variables [22]. Traditional personalization approaches usually build a profile of interests for each user from her/his search or browsing history. Contextual information is useful in identifying users' search needs. Shen et al. [24] presented context-aware language models by assuming that documents are not only similar to the current query but also similar to the previous queries and the summaries of the documents clicked on. Sun and Lou [27] focus on right-click query that is submitted to a search engine by making a text string in a Web page, and extract the contextual information from the source document to improve search results. Cao et al. [6, 5] extracted context information in Web search sessions by modeling search sessions as sequences of user queries and clicks. They learned sequential prediction models such Hidden Markov Model [29, 30] from search log data. Different from our study here, their models [6, 12] either fail to leveraging both QAC and clickthrough logs, or do not fully explore the relationship inbetween.



(a) Correlation (b) QAC Pattern (c) Click Pattern

Figure 6: Correlation between QAC Behavior Patterns and Click Behavior Patterns. **The left figure shows the percentage of the number of pattern pairs with the degrees of correlation in different ranges. The middle and right figures together show a pair of highly correlated QAC and click behavior patterns. The middle figure shows the scaled value of some selected features in the QAC behavior pattern. Indices of selected features of QAC behaviors are: 1-'Typing Speed', 2-'Time Duration', 3-'Highest Non-Click Position', 4-'Type Speed Standard Deviation', 5-'Typing Completion', The right figure shows the scaled value of some selected features in the click behavior pattern. Indices of selected features of click behavior patterns are: 1-'Click Number', 2-'Dwell Time', 3-'Click Speed', 4-'Scanned Pages', 5-'Search Time'.**

Query Auto-Completion (QAC). The main objective of QAC [19, 18] is to predict users' intended queries and assist them formulate a query while typing. The most popular QAC algorithm is to suggest completions according to their past popularity. Generally, a popularity score is assigned to each query based on the frequency of the query in the query log from which the query database was built. This simple QAC algorithm is called MostPopularCompletion (MPC), which can be regarded as an approximate maximum likelihood estimator [1].

Several QAC methods [1, 26, 25, 31] were proposed to extend MPC from various aspects. Bar-Yossef and Kraus [1] introduced the context-sensitive QAC method by treating users' recent queries as context and taking into account the similarity of QAC candidates with this context for ranking. But there is no consensus of how to optimally train the relevance model. Shokouhi [25] employed learning-based strategy to incorporate several global and personal features into the QAC model. However, these methods only exploit the final submitted query or simulate the prefixes of the clicked query, which do not investigate the users' interactions with the QAC engine. In addition the above models, there are several studies addressing different aspects of QAC. For example, [26, 31] focused on the time-sensitive aspect of QAC. Other methods studied the space efficiency of index for QAC [2, 14]. Duan and Hsu [10] addressed the problem of suggesting query completions when the prefix is mis-spelled. Kharitonov et al. [17] proposed two new metrics for offline QAC evaluation, and [15] investigated user reformation behavior for QAC.

The QAC is a complex process where a user goes through a series of interactions with the QAC engine before clicking on a suggestion. As seen from the related work, little attention has been paid to understand the interactions with the QAC engine. Until recently, Li et al. [20] created a two-dimensional click model to combine users' behaviors with the existing learning-based QAC model. The study assumed users' behaviors at different keystrokes, even for the consecutive two keystrokes, are independent in order to simplify the model estimation, which results in information lose. In this paper, we attempt to directly model and leverage the relationship between users' behaviors, so as to improve the performance of QAC.

Click Models. This work is also related to click models. In the field of document retrieval, the main purpose for modeling users' clicks is to infer the intrinsic relevance between the query and document by explaining the positional bias. The position bias assumption was first introduced by Granka et al. [13], stating that a document on higher rank tends to attract more clicks. Richardson et al. [23] attempted to model the true relevance of documents by imposing a multiplicative factor. Later examination hypothesis is formalized in [8], with a key assumption (Cascade Assumption) that a user will click on a document if and only if that document has been examined and it is relevant to the query. In addition, several extensions were proposed, such as the User Browsing Model (UBM) [11], the Bayesian Browsing Model [21], the General Click Model [33], and the Dynamic Bayesian Network model (DBN) [7]. Despite the abundance of click models, these existing click models cannot be directly applied to QAC without considerable modification. The click model most similar to our work is [32], which models users' clicks on a series of queries in a session. However because of the main difference between QAC and document retrieval, our model is very different from [32].

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a LDA-based probabilistic model to study users' behaviors on both QAC and click logs simultaneously by using both logs as the contextual data for each other. The model is designed to capture the correlation between users' behavior patterns on a QAC log and those on a click log. The learned users' behavior patterns on both QAC and click logs are utilized to benefit the query auto-completion task and the prediction of users' clicks on web documents as well as the relevance ranking of them. We have applied the proposed Context-LDA model to study users' behaviors on both real-world QAC and click logs collected from a commercial search engine, and compare with several alternative approaches. Experimental results show that our proposed model offers a better context-aware solution to both applications of query auto-completion and learning to rank. In future work, it would be interesting to consider the usage of additional user behavior features in the proposed model. Meanwhile, we plan to investigate alternative models, such as point process models, that can effectively capture the relationship between user behaviors on QAC and click logs.

7. REFERENCES

[1] Z. Bar-Yossef and N. Kraus. Context-sensitive query auto-completion. In *Proceedings of the 20th international conference on World wide web*, pages 107–116. ACM, 2011.

[2] H. Bast and I. Weber. Type less, find more: fast autocompletion search with a succinct index. In *Proceedings of the 29th annual international ACM SIGIR conference*, pages 364–371. ACM, 2006.

[3] D. Blei and M. Jordan. Variational inference for dirichlet process mixtures. In *Bayesian Analysis*, volume 1, pages 121–144, 2005.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[5] H. Cao, D. Jiang, J. Pei, E. Chen, and H. Li. Towards context-aware search by learning a very large variable length hidden markov model from search logs. In *Proceedings of the 18th International Conference on World Wide Web*, pages 191–200, 2009.

[6] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li. Context-aware query suggestion by mining click-through and session data. In *KDD*, pages 875–883, 2008.

[7] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1–10, New York, NY, USA, 2009. ACM.

[8] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 87–94. ACM, 2008.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[10] H. Duan and B.-J. P. Hsu. Online spelling correction for query completion. In *Proceedings of the 20th international conference on World wide web*, pages 117–126. ACM, 2011.

[11] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st annual international ACM SIGIR conference*, pages 331–338. ACM, 2008.

[12] J. Gao, G. Xu, and J. Xu. Query expansion using path-constrained random walks. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 563–572, New York, NY, USA, 2013. ACM.

[13] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th annual international ACM SIGIR conference*, pages 478–479. ACM, 2004.

[14] B.-J. P. Hsu and G. Ottaviano. Space-efficient data structures for top-k completion. In *Proceedings of the 22nd international conference on World Wide Web*, pages 583–594, 2013.

[15] J.-Y. Jiang, Y.-Y. Ke, P.-Y. Chien, and P.-J. Cheng. Learning user reformulation behavior for query auto-completion. In *Proceedings of the 37th international ACM SIGIR conference*, pages 445–454. ACM, 2014.

[16] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142. ACM, 2002.

[17] E. Kharitonov, C. Macdonald, P. Serdyukov, and I. Ounis. User model-based metrics for offline query suggestion evaluation. In *Proceedings of the 36th international ACM SIGIR conference*, pages 633–642. ACM, 2013.

[18] L. Li, H. Deng, J. Chen, and Y. Chang. Learning parametric models for context-aware query auto-completion via hawkes processes. WSDM '17, pages 131–139, New York, NY, USA, 2017. ACM.

[19] L. Li, H. Deng, A. Dong, Y. Chang, H. Zha, and R. Baeza-Yates. Analyzing user's sequential behavior in query auto-completion via markov processes. SIGIR '15, pages 123–132, New York, NY, USA, 2015. ACM.

[20] Y. Li, A. Dong, H. Wang, H. Deng, Y. Chang, and C. Zhai. A two-dimensional click model for query auto-completion. In *Proceedings of the 37th International ACM SIGIR Conference*, pages 455–464. ACM, 2014.

[21] C. Liu, F. Guo, and C. Faloutsos. Bayesian browsing model: Exact inference of document relevance from petabyte-scale data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(4):19, 2010.

[22] M. Melucci. Contextual search: A computational framework. *Foundations and Trends in Information Retrieval*, 6(4-5):257–405, 2012.

[23] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.

[24] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th international ACM SIGIR conference*, pages 43–50, 2005.

[25] M. Shokouhi. Learning to personalize query auto-completion. In *Proceedings of the 36th international ACM SIGIR conference*, pages 103–112. ACM, 2013.

[26] M. Shokouhi and K. Radinsky. Time-sensitive query auto-completion. In *Proceedings of the 35th international ACM SIGIR conference*, pages 601–610. ACM, 2012.

[27] A. Sun and C. Lou. Towards context-aware search with right click. In *The 37th International ACM SIGIR Conference*, pages 847–850, 2014.

[28] H. Wang, C. Zhai, A. Dong, and Y. Chang. Content-aware click modeling. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1365–1376, 2013.

[29] S. Wang, Y. Wang, J. Tang, C. Aggarwal, S. Ranganath, and H. Liu. Exploiting hierarchical structures for unsupervised feature selection. SDM, 2017.

[30] Y. Wang, S. Wang, J. Tang, G. Qi, H. Liu, and B. Li. Clare:a joint approach to label classification and tag recommendation. AAAI, 2017.

[31] S. Whiting and J. M. Jose. Recent and robust query auto-completion. In *Proceedings of the 23rd international conference on World wide web*, pages 971–982, 2014.

[32] Y. Zhang, W. Chen, D. Wang, and Q. Yang. User-click modeling for understanding and predicting search-behavior. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1388–1396. ACM, 2011.

[33] Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen. A novel click model and its applications to online advertising. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 321–330. ACM, 2010.