

# Identifying and Labeling Search Tasks via Query-based Hawkes Processes

Liangda Li<sup>†</sup>, Hongbo Deng<sup>‡</sup>, Anlei Dong<sup>‡</sup>, Yi Chang<sup>‡</sup>, and Hongyuan Zha<sup>†</sup>

<sup>†</sup>College of Computing  
Georgia Institute of Technology  
Atlanta, GA 30032  
{ldli, zha}@cc.gatech.edu

<sup>‡</sup>Yahoo Labs  
701 First Avenue  
Sunnyvale, CA 94089  
{hbdeng, anlei, yichang}@yahoo-inc.com

## ABSTRACT

We consider a search task as a set of queries that serves the same user information need. Analyzing search tasks from user query streams plays an important role in building a set of modern tools to improve search engine performance. In the paper, we propose a probabilistic method for identifying and labeling search tasks that rely on the following intuitive observations: queries that are issued temporally close by users in many sequences of queries are likely to belong to the same search task, meanwhile, different users may having the same information needs tend to submit topically coherent search queries. To capture the above intuitions, we directly model query temporal patterns using a special class of point process called Hawkes processes, and combine topic model with Hawkes processes for simultaneously identifying and labeling search tasks. Essentially, Hawkes processes utilize their self-exciting properties to identify search tasks if influence exists among a sequence of queries for individual users, while the topic model exploits query co-occurrence across different users to discover the latent information needed for labeling search tasks. More importantly, there is mutual reinforcement between Hawkes processes and the topic model in the unified model that enhances the performance of both. We evaluate our method based on both synthetic data and real-world query log data. In addition, we also consider application to query clustering and search task identification. By comparing with state-of-the-art methods, the results demonstrate that the improvement in our proposed approach is consistent and promising.

### Categories and Subject Descriptors:

I.2.6 [Artificial Intelligence]: Learning; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; J.4 [Computer Applications]: Social and Behavioral Sciences

**General Terms:** Algorithm, Experimentation, Performance

**Keywords:** Hawkes process, latent Dirichlet allocation, variational Inference, search task

## 1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

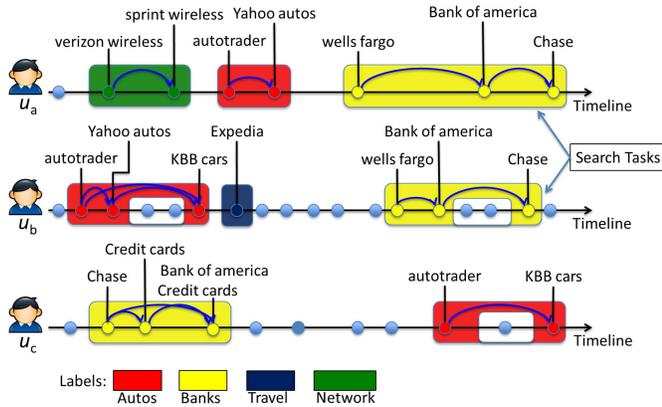
WOODSTOCK '97 El Paso, Texas USA

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

Nowadays, search engines have become the most important and indispensable Web portal, whereby people pursue a wide range of searches in order to satisfy a variety of *information needs*. To better understand users' information needs and search behaviors, one important research direction is to detect and split users' temporal sequences of queries into disjoint *query sessions*, which are often defined as a sequence of queries issued within a fixed period of time, ranging from 5 to 120 minutes. However, a user's single session may contain queries with multiple intents, or consist of seeking information on single or multiple topics [23]. Beyond that, a *search task* [20, 15, 25], which is defined as a set of queries serving for the same *information need*, has been recognized as a more suitable atomic unit than single query or session, not only for better modeling user search intent but also for improving other downstream search engines' applications, such as query suggestion [8, 15] and personalized search [28]. Analyzing the formation of search tasks also deepens our understanding of the simultaneous temporal diffusion of multiple memes, i.e. intents or ideas, in social networks. Therefore, how to effectively identify and label search tasks becomes an interesting and challenging problem.

Recently, there have been attempts to extract in-session tasks [23, 20, 15], and cross-session tasks [16, 17, 1, 25] from sequences of queries. They build clustering or classification methods to identify tasks based on time splitting, lexicon similarity, and query reformulation patterns. Even though the temporal submission patterns in query sequences carry valuable information for mining search tasks, those existing methods only use them for either simply splitting sequences of queries into temporally-demarcated sessions [16, 17, 20], or transforming them as pairwise features among queries [15, 25], and *none of them explicitly exploits the temporal information directly in their models*. We believe that by directly modeling temporal information as part of extracting search tasks we can substantially improve search task mining. Another key drawback of those existing methods is that most of them focus only on the query sequences of individual users instead of considering the whole query log. But it is obvious that different users may have the same information need, and share the same search task, thus modeling query sequences across different users will be very valuable for capturing semantically similar search tasks in a global context.

Generally, two consecutive queries issued by a user are more likely to belong to the same search task than two non-contiguous queries, but that is not necessary always the case. For example, as shown in Figure 1, the consecutive queries "autotrader" and "Yahoo autos" issued by user  $u_a$  belong to the same search task, while the consecutive queries "Yahoo autos" and "wells fargo" belong to two separate search tasks. Another complicated case is that two non-contiguous queries may belong to the same or different tasks



**Figure 1: An Illustration of Relationship between Consecutive Queries and Search Tasks.** Every circle represents a query issued by a user at time  $t_n$ . The blue arrow line indicates an influence exists between queries. A set of queries linked by blue lines denotes a search task, and some topically coherent search tasks across three users are labeled by different colors.

as well, e.g., “verizon wireless” and “autotrader” issued by user  $u_a$  belong to two different search tasks, but “Yahoo autos” and “KBB cars” issued by users  $u_b$  belong to the same search task. These examples show that in reality we cannot simply rely on the time splitting or an individual user behavior for identifying search tasks. It makes more sense to take into account the explicit temporal information of query sequences exhibited by many different users in the whole query logs. *The basic intuition is that if two consecutive or temporally-close queries are issued many times by the same user or many others users, it is more likely these two queries are semantically related to each other, i.e., belong to the same search task.* As we can see, the consecutive queries “wells fargo” and “Bank of america” are issued by both  $u_a$  and  $u_b$  (and possible many other users), while the consecutive queries “Yahoo autos” and “wells fargo” are only issued by user  $u_a$ . Therefore, according to the above intuition, “wells fargo” and “bank of america” are more likely to belong to the same search task compared with “Yahoo auto” and “wells fargo”. Similarly, for non-contiguous queries, “autotrader” and “KBB cars” are issued temporally very close by both  $u_b$  and  $u_c$ , which indicates they have higher chance of belonging to the same search task. Moreover, different users may engage in different search patterns, e.g., user  $u_b$  search more frequently than user  $u_a$ , which indicates how likely the search tasks may change within a certain time period for different users, and then they should be treated differently based on their search activities. All in all, we choose to identify search tasks by leveraging the temporally weighted query co-occurrence — this not only guarantees sound performance by making full use of both textual and temporal information of the entire query sequences, but it also enables the labeling of the identified search tasks since semantically related queries are clustered together through query links determined by co-occurrence.

To model temporally close query co-occurrences, we turn to Latent Dirichlet Allocation (LDA) [6], one powerful graphical model that exploits co-occurrence patterns of queries in query sequences. Existing temporal LDA models [14, 27] learned distinguished topic distributions from temporal fragments of data, while ignored query co-occurrence across different fragments, thus failed to make full use of the temporal information. Recently, some spatial-LDA models [26] encouraged queries that are very close in space to share similar topic distributions, i.e., weighing the reliability of query

co-occurrence based on spatial closeness. However, there exists no uniform standards for measuring such closeness across different instances, especially in temporal data. Our paper, on the other hand, considers to make full use of temporal information by weighing the reliability of each co-occurrence of a pair of queries based on how likely an *influence* exists between this pair of queries. Here we define query *influence* as:

- The occurrence of one query raises the probability that the other query will be issued in the near future.

*Influence*, rather than closeness, enables us to distinguish temporally close query co-occurrence from temporally regular query co-occurrence for each user based on one’s own frequency of query submission. For instance, in Figure 1, the absolute temporal distance between “KBB cars” and “Expedia” is smaller than that of “verizon wireless” and “sprint wireless”, however, *influence* exists between the latter pair of queries rather than the former one, since user  $u_b$ ’s query submission frequency is much larger than that of user  $u_a$ . To model such personal frequency and *influence*, in this paper, we utilize Hawkes processes [12], a special class of point processes, whose intensity function implies how likely an event will happen at each timestamp. The intensity function of Hawkes includes a base intensity, along with a positive influence of the past events on the current one. Such a positive influence is originated from the *self-exciting* property that the occurrence of one event in the past increases the probability of events happening in the future. We find that Hawkes’s *self-exciting* property coincides with the concept of *influence* in our situation, and its base intensity captures the personal frequency. Thus we employ Hawkes processes to fully utilize temporal information in query sequences for identifying *influence* existence. From the perspective of Hawkes, *influence* generally exists between temporally-close queries. However, for an observed query sequence, not all temporally-close query-pairs have the actual *influence* in between, since in some cases the occurrence of the later queries may result from the base intensity rather than *self-exciting* property. Furthermore, existing Hawkes models [18, 29] find it intractable to obtain an optimal solution of *influence* existence based on temporal information only. Last but not the most important, it is unable to directly identify search tasks by either generating topics based on query co-occurrence using LDA, or estimating all *influence* candidates by Hawkes. To address the above issues, we concentrate on the *influence* existence between semantically related queries, whose estimation can be simplified by the joint efforts of LDA and Hawkes, and enables a direct identification of search tasks.

According to the above intuition, a search task can be viewed as a sequence of semantically related queries linked by *influence*. A query that does not satisfy user’s *information need* will *self-excite* the submission of another semantically related query in the near future. Thus our paper solves search task identification directly by identifying those semantic *influence*. To limit the solution space of such *influence*, we cast both *influence* existence and query-topic membership into latent variables, and equalize the existence probability of pairwise *influence* with the similarity of the memberships of associated two queries. This equalization works as a bridge between the LDA model and Hawkes processes, as LDA assigns high *influence*-qualified co-occurred queries to the same topic, while query co-occurrence frequency narrows the solution space of semantic *influence*. In this way, LDA and Hawkes mutually benefit each other in identifying search tasks using both temporal and textual information. We propose a probabilistic model that incorporates this equalization to combine the LDA model with Hawkes processes, and develop a mean-field variational inference algorithm to estimate the semantic *influence* by optimizing the data likeli-

hood. We evaluate our method on synthetic data, and also apply it to mine search tasks in both AOL and Yahoo query log data. Experimental results show that the proposed method can achieve better performance than existing state-of-the-art methods.

In a nutshell, our major contributions of this paper include: (1) We cast search task identification into the problem of identify semantic *influence* in observed query sequences, and propose a probabilistic model by combining LDA model with Hawkes processes to address the problem. Most importantly, there is mutual reinforcement between Hawkes processes and the topic model in the unified model that enhances the performance of both. (2) We employ Hawkes processes to directly model temporal information as part of search tasks identification, which has never been explicitly exploited in the existing works.

The rest of the paper is organized as follows. We first introduce Hawkes processes, and the proposed model by combining LDA with Hawkes processes in Section 2. In Sections 3, we develop a fast mean-field variational inference algorithm to optimize the solution. We then describe and report the experimental evaluations in Section 4. Finally, we introduce the related work in Section 5, and present our conclusions and future work in Section 6.

## 2. PROBLEM DEFINITION

Let us consider a typical scenario that  $M$  users issue  $M$  corresponding query sequences, and we mark the query sequence issued by user  $m$  as  $T_m = \{t_{m,n}, n = 1, \dots, N_m\}$ . We denote the word set of the  $n$ -th query by user  $m$  as  $W_{m,n} = \{w_{m,n,1}, \dots, w_{m,n,c_{m,n}}\}$ . Existing works generally identify search task by sequentially solve two subproblems: 1) using query’s textual information to cluster queries in observed query sequences, and 2) using obtained clusters together with temporal information to partition query sequences into search tasks. In this section, we show how these two subproblems can be simultaneously addressed by combining Hawkes processes with the LDA model, and how temporal and textual information can be collaboratively combined to address the above two subproblems. We also show how our model can be used to automatically label search tasks along with search task identification.

### 2.1 Query Co-occurrence and LDA

Our paper chooses to address the query clustering problem using graphical models like LDA [6], which has been proven to be effective in topic discovery by clustering words that co-occur in the same document into topics. Let us first introduce how to use LDA to cluster queries based on their textual information only. One straightforward idea is to treat each user’s query sequence as a document, and cluster queries that co-occur in the same query sequence into topics, since queries issued by the same user are generally more likely to share the same *information need* than queries issued by different users. Since we focus on query co-occurrence instead of word co-occurrence, we enforce that words in one query belong to the same topic. Our LDA model assumes  $K$  topics lie in the given query sequences, and each user  $m$  is associated with a randomly drawn vector  $\pi_m$ , where  $\pi_{m,k}$  denotes the probability that a query issued by user  $m$  belongs to topic  $k$ . For the  $n$ -th query in the query sequence of user  $m$ , a  $K$ -dimensional binary vector  $Y_{m,n} = [y_{m,n,1}, \dots, y_{m,n,K}]^T$  is used to denote the query’s topic membership. One challenge we encounter in the inference of topic membership  $Y$  is that, without temporal information of queries, it is difficult to judge whether two non-contiguous co-occurred queries should belong to the same topic or not. A pair of queries that co-occurs a lot may be completely unrelated if the temporal gap between them is always large.

Since the co-occurrence of queries with large temporal gap is

useless or harmful, we make use of temporal information to decide which query co-occurrence should be taken into account by LDA, i.e., how a document in LDA model is defined/constituted. One simple way of utilizing temporal information is to define a document as consecutive queries in a fixed time window (or time session), thus enable us to focus on temporally close query co-occurrence. Temporally close queries that issued many times by the same user or many other users are more likely to be semantically related to each other, i.e., belong to the same search task. However, a time window based LDA model may suffer from the following drawbacks: 1) Usually no optimal solution exists for cutting the entire query sequence into different time-sessions. If we allow different time-sessions to overlap, redundant query co-occurrence will be taken into account; otherwise, pairs of queries with very small temporal gap can be partitioned into different tasks, which may cause information loss. 2) Using time windows will ignore or misunderstand users’ own temporal patterns in searching.

To address the above drawbacks, we can weigh each query co-occurrence based on how likely an *influence* exists between this pair of queries, i.e., the occurrence of one query raises the probability that the other query will be issued in the near future. That is to say, one document is a subsequence of queries linked through *influence*. This *influence*, rather than time window, enables us to distinguish temporally close query co-occurrence from temporally regular query co-occurrence for each user based on his/her own frequency of query submission. To model such personal frequency and *influence*, in this paper, we utilize Hawkes processes, to capture the temporal information in different query sequences.

### 2.2 Hawkes Process

One powerful tool in statistics for modeling event (query) sequence data is the point process, which is widely used to describe data that are localized at a finite set of time points  $\{t_1, \dots, t_N\}$ . Typically, in a point process,  $N(t)$  counts the number of points (i.e., occurrences of events) in  $(-\infty, t]$ , and the conditional intensity function  $\lambda(t|\mathcal{H}_t)$  denotes the expected instantaneous rate of future events at timestamp  $t$  depending on  $\mathcal{H}_t$ , the history of events preceding  $t$ . For clarity, hereafter we use  $*$  to imply the dependence on  $\mathcal{H}_t$ , i.e.,  $\lambda(t|\mathcal{H}_t)$  will be denoted  $\lambda^*(t)$ .

The Hawkes process is a class of self- or mutually-exciting point process models [12]. A univariate Hawkes process  $\{N(t)\}$  is defined by its intensity function

$$\lambda^*(t) = \mu(t) + \int_{-\infty}^t \kappa(t-s)dN(s),$$

where  $\mu : \mathbb{R} \rightarrow \mathbb{R}_+$  is a deterministic base intensity,  $\kappa : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is a kernel function expressing the positive influence of past events on the current value of the intensity process. The process is well known for its *self-exciting* property, which refers to the phenomenon that the occurrence of one event in the past increases the probability of events happening in the future. Such *self-exciting* property can either exist between every pair of events as assumed in a normal univariate Hawkes process, or only exists between limited pair of events. For instance, any query but the last in a search task can imply an increased probability of future queries issued in the same search task, since the user’s *information need* in this search task hasn’t been satisfied. Meanwhile, queries from different search tasks may rarely affect each other.

Since our definition of *influence* coincides with the *self-exciting* property of Hawkes process, we propose to identify the *influence* among queries by building one separate Hawkes process on each user’s query sequence. In the query sequence of user  $m$ , we use  $R_{m,n,n'}$  to denote whether *influence* exists between the  $n$ -th and

$n'$ -th query. If *influence* exists, we believe that the occurrence of  $n$ -th query has a time-decay effect on increasing the intensity at the timestamp of the occurrence of the  $n'$ -th query. Thus based on *influence*  $R_m$ , we model the query sequence issued by user  $m$  with a univariate Hawkes process, whose intensity can be written as:

$$\lambda_m(t) = \mu_m + \sum_{t_{m,l} < t} R_{m,l,n} \beta_m \kappa(t - t_{m,l}) \quad (1)$$

The baseline intensity  $\mu_m$  captures how often user  $m$  issues a query spontaneously<sup>1</sup> (i.e., not triggered by any other queries), while  $\beta_m$  models the degree of *influence* between sequential queries issued by user  $m$ , and  $\kappa(t - t_{m,l})$ <sup>2</sup> captures the time-decay effect only.

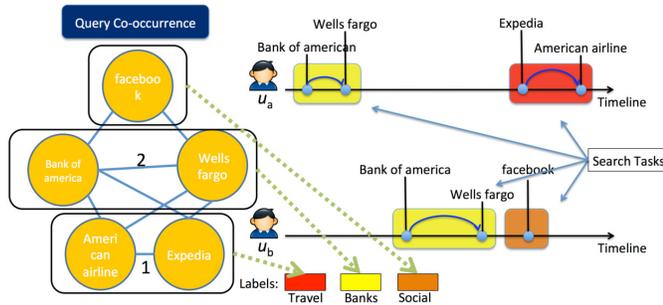
*Influence*  $R$  can be estimated together with  $\mu$  and  $\beta$  by maximizing the likelihood of the proposed Hawkes model on observed query sequence  $\{T_m = \{t_{m,n}\}\}$ . The estimation of  $R$  is actually to identify query-pairs that the occurrence of the later query most likely violates the normal query-submission frequency, and gets triggered by the earlier one. In other words, if *influence* exists between two queries, the corresponding temporal gap can be significantly less than the average temporal gap of pairs of queries in the same query sequence (issued by the same user). Since the definition of *influence* suggests that queries linked by significant *influence* naturally form search tasks, a thresholding of  $R_{m,l,n} \beta_m \kappa(t - t_{m,l})$  with a small constant automatically results in search task partition. The estimation of  $R$  consequently partitions observed query sequences into search tasks.

### 2.3 LDA-Hawkes

Estimated by Hawkes processes, *influence*  $R$  captures the unique temporal pattern of each user's query sequence. We use  $R$  to weight the query co-occurrence, which bridges the LDA model and Hawkes process through:

$$R_{m,n,n'} = Y_{m,n}^T * Y_{m,n'}, \quad (2)$$

that is to say, *influence* exists between these two queries if and only if the two queries share the same topic. Since queries in the same search task are linked by *influence*, all queries in the same search task share the same topic, which labels this search task as well.



**Figure 2: A Toy Example of our LDA-Hawkes model. Blue line denotes the influence among queries. Green dash line shows the label each query belongs to.**

<sup>1</sup>For simplicity, we assume this cascade-birth process is a homogeneous Poisson process with  $\mu_m(t) = \mu_m$ .

<sup>2</sup>Our paper uses the exponential kernel in experiments, i.e.,  $\kappa(\Delta t) = \omega e^{-\omega \Delta t}$  if  $\Delta t \geq 0$  or 0 otherwise. However, the model development and inference is independent of kernel choice and extensions to other kernels such as power-law, Rayleigh, non-parametric kernels are straightforward.

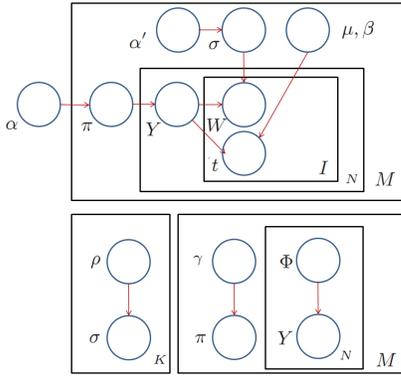
Though our defined bridge between *influence*  $R$  and query-topic membership  $Y$ , the Hawkes process and the LDA model mutually benefit each other in identifying and labeling search tasks. On one hand, provided *influence* among queries, we obtain 0-1 weighted query co-occurrence of each candidate query-pair in observed query sequences, and generate topics accordingly. For instance, in Figure 2, although 8 pairs of queries (9 possible combinations with 8 unique query-pairs) co-occur in query sequences, only the co-occurrences of query-pairs “bank of america”–“wells fargo” and “Expedia”–“american airline” have positive weights. These weighted query co-occurrences embed personal temporal information, thus are expected to lead to improved topics compared with existing LDA-based methods [14, 27, 26] that used no weight scheme or only uniform standard weight scheme.

On the other hand, the estimation of *influence*  $R$  based on temporal data  $\{T_m\}$  only can be intractable, since the exploration of the whole space of  $R$  is known to be very costly ( $2^{\sum_m N_m}$  possible solutions). LDA-Hawkes further makes use of textual data to limit the output space of  $R$  to the most probable subspace, since topics learned by the LDA part in turn justify the *influence* existence between each pair of queries. Two queries rarely co-occur can be clustered into different topics by the LDA part, based on such query-topic membership no *influence* exists between these two queries. For example, in the query sequence of user  $u_b$  shown in Figure 2, the temporal gap between query-pair “bank of america”–“wells fargo” is larger than the temporal gap between query-pair “wells fargo”–“facebook”. However, the pair of queries “bank of america”–“wells fargo” also co-occurs in the query sequence of user  $u_a$ , while “wells fargo”–“facebook” does not, which in turn emphasizes that *influence* should exist between “bank of america” and “wells fargo” rather than between “wells fargo” and “facebook”. To sum up, combined through *influence*, Hawkes process and LDA reciprocally contribute to the search task identification and labeling.

Finally, we present our generative model that combines Hawkes process and LDA as follows:

- For each topic  $k$ , draw a  $V$  dimensional membership vector  $\sigma_k \sim \text{Dirichlet}(\alpha')$ .
- For each user  $m$ , draw a  $K$  dimensional membership vector  $\pi_m \sim \text{Dirichlet}(\alpha)$ .
- For the content of the  $n$ -th query issued by user  $m$ ,
  - $Y_{m,n} \sim \text{Multinomial}(\pi_m)$ ;
  - For the  $i$ -th word in the  $n$ -th query issued by user  $m$ ,
    - \*  $w_{m,n,i} \sim \text{Multinomial}(Y_{m,n}, \sigma)$ ;
- For the timestamp of the sequence of queries issued by user  $m$ ,
  - draw personal base intensity  $\mu_m$  and degree of *influence*  $\beta_m$ ;
  - derive  $R_m$  from  $\{Y_{m,n}\}$  through Eqn (2);
  - $N_m(\cdot) \sim \text{HawkesProcess}(\lambda_m(\cdot))$ , where the intensity  $\lambda_m$  is defined as in Eqn (1).

Here  $V$  is the size of vocabulary. Note that in our LDA-Hawkes model, queries issued by one user share the same topic distribution, while words in one query belong to the same topic. The topic membership of the  $n$ -th query of user  $m$ ,  $Y_{m,n}$ , determines not only the words the query owns, but also the timestamp of its occurrence through Hawkes process  $\lambda_m(\cdot)$ .



**Figure 3: Graphical model representation of LDA-Hawkes and the variational distribution that approximates the likelihood. The upper figure shows the graphical model representation of LDA-Hawkes, while the lower figure shows the variational distribution that approximates the likelihood.**

Under our LDA-Hawkes model, the joint probability of data  $T = \{N_m(\cdot)\} = \{\{t_{m,n}\}_{n=1}^{N_m}\}$ ,  $W = \{\{W_{m,n}\}_{n=1}^{N_m}\}$  and latent variables  $\{\pi_{1:M}, Y\}$  can be written as follows:

$$p(T, W, \pi_{1:M}, Y, \sigma | \alpha, \alpha', \mu, \beta) \\ = \prod_m P(\{t_{m,n}\}_{n=1}^{N_m} | Y_{m,1:N_m}, \mu_m, \beta_m) \prod_m \prod_n \prod_i P(w_{m,n,i} | Y_{m,n}, \sigma) \\ \prod_m \prod_n P(Y_{m,n} | \pi_m) \prod_m P(\pi_m | \alpha) \prod_k P(\sigma_k | \alpha')$$

### 3. INFERENCE

Statistical inference of non-Markovian point process has attracted increasingly interest recently. On the other hand, despite that a tremendous amount of works on inference of topic models have been published, very few of them are proposed to solving topic model combined with point processes. In this section, we derive a mean-field variational Bayesian inference algorithm for our proposed LDA-Hawkes model.

#### 3.1 Variational Inference

Under LDA-Hawkes model, given observations of both temporal information  $T = \{N_m(\cdot)\} = \{\{t_{m,n}\}_{n=1}^{N_m}\}$  and textual information  $W = \{\{W_{m,n}\}_{n=1}^{N_m}\}$  of query sequences, the log-likelihood for the complete data is given by  $\log P(T, W | \mu, \beta, \alpha, \alpha')$ . Since this true posterior is hard to infer directly, we turn to variational methods [5], whose main idea is to posit a distribution of the latent variables with free parameters, and then fit those parameters such that the distribution is close to the true posterior in Kullback-Leibler (KL) divergence. The variational distribution is supposed to be simpler than the true posterior, thus enable us to approximately solve the original optimization problem. In Figure 3, the lower part shows the variational distribution that approximates the data likelihood. Our paper chooses to introduce a distribution of latent variables  $q$  that depend on a set of free parameters, and specify  $q$  as the mean-field fully factorized family as follows:

$$q(\pi_{1:M}, Y, \sigma_{1:K} | \gamma_{1:M}, \Phi, \rho_{1:K}) \\ = \prod_m q_1(\pi_m | \gamma_m) \prod_m \prod_n q_2(y_{m,n} | \phi_{m,n}) \prod_k q_1(\sigma_k | \rho_k)$$

where  $q_1$  is a Dirichlet,  $q_2$  is a multinomial, and  $\{\gamma_{1:M}, \Phi, \rho_{1:K}\}$  are the set of free variational parameters that are optimized to tight

the following lower bound  $\mathcal{L}'$  for our likelihood:

$$\log p(T, W | \mu, \beta, \alpha, \alpha') \geq E_q[\log p(T, W, \pi_{1:M}, Y, \sigma | \alpha, \alpha', \mu, \beta)] \\ - E_q[\log q(\pi_{1:M}, Y, \sigma_{1:K})]. \quad (3)$$

Isolating terms containing  $\lambda$  in Eqn (3), we have

$$\mathcal{L}_h = \sum_{m=1}^M \sum_n E_q(\log \lambda(t_{m,n})) - \sum_{m=1}^M \int_0^T E_q(\lambda(s)) ds, \quad (4)$$

as the partial likelihood on temporal data assuming query-topic distribution is known, where the second term reduces to  $\sum_{m=1}^M b_m + T \sum_{m=1}^M \mu_m$ . Here

$$b_m = \sum_{n=1}^{N_m} \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n})(K(t_{m,n} - t_{m,l}) - K(t_{m,n-1} - t_{m,l})).$$

where  $K(t) = \int_0^t \kappa(s) ds$ , and we define function  $r(\phi_{m,l}, \phi_{m,n}) = \sum_k \phi_{m,l,k} \phi_{m,n,k}$ , which can be viewed as the latent variable that approximates *influence*  $R$ . On the other hand, to break down the expectation of the log-intensity  $E_q(\log \lambda(t_{m,n}))$  involved in the first term in Eqn (4), we apply Jensen's inequality

$$\mathbb{E}_q[\log(\lambda(t_{m,n}))] \geq \eta_{m,nn} \log(\mu_m) + \sum_{l=1}^{n-1} \eta_{m,ln} \log(\beta_m \kappa(t_{m,n} - t_{m,l})) \\ - \eta_{m,nn} \log(\eta_{m,nn}) - \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \eta_{m,ln} \log(\eta_{m,ln}),$$

where we introduce a set of branching variables  $\{\eta_m\}_{m=1}^M$ . Note that each  $\eta_m$  is a  $n \times n$  lower-triangular matrix with  $n$ -th row  $\eta_{m,n} = [\eta_{m,1n}, \dots, \eta_{m,nn}]^T$ . The branching  $\eta$  also satisfies the following conditions:

$$\eta_{m,ln} \geq 0, \eta_{m,nn} + \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \eta_{m,ln} = 1.$$

Under a coordinate descent framework, we optimizing the lower bound as in Eqn (3) against each variational latent variables<sup>3</sup> and the model hyper-parameters, including both LDA hyper-parameters and Hawkes hyper-parameters. For variational latent variables, we have

- update rules for  $\phi^*$ 's as:

$$\phi_{m,n,k} \propto \exp(\sum_m (\Psi(\gamma_{m,k}) - \Psi(\sum_k \gamma_{m,k}))) : \text{Topic distribution} \\ \times \sum_i \sum_v w_{m,n,i,v} [\Psi(\rho_{k,v}) - \Psi(\sum_v \rho_{k,v})] : \text{Word content} \\ \times \sum_{l=1}^{n-1} f_{l,n} : \text{influences from past} \\ \times \sum_{l'=n+1}^{N_m} f_{n,l'} : \text{influences to future} \quad (5)$$

where we define  $f_{l,n} = \phi_{m,l,k} (\eta_{m,ln} \log(\frac{\beta_m \kappa(t_{m,n} - t_{m,l})}{\eta_{m,ln}}) - \beta_m (K(t_{m,n} - t_{m,l}) - K(t_{m,n-1} - t_{m,l})))$ ;

- update rules for  $\gamma^*$ 's as:

$$\gamma_{m,k} = \alpha_k + \sum_n \phi_{m,n,k};$$

<sup>3</sup>Here we categorize branching variables  $\eta$  as variational latent variables.

- update rules for  $\rho$ 's as:

$$\rho_{k,v} \propto \alpha'_v + \sum_m \sum_n \sum_i \phi_{m,n,k} w_{n,i,v};$$

- and update rules for  $\eta$  as:

$$\eta_{m,nn} = \frac{\mu_m}{\mu_m + \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \beta_m \kappa(t_{m,n} - t_{m,l})},$$

$$\eta_{m,ln} = \frac{\beta_m \kappa(t_{m,n} - t_{m,l})}{\mu_m + \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \beta_m \kappa(t_{m,n} - t_{m,l})}.$$

### 3.2 Learning

We use a variational expectation-maximization (EM) algorithm [10] to compute the empirical Bayes estimates of LDA hyper-parameters  $\alpha$  and  $\alpha'$  in our LDA-Hawkes model. This variational EM algorithm optimize the lower bound as in Eqn (3) instead of the real likelihood, it iteratively fits the variational distribution  $q$  to approximate the posterior and maximizes the corresponding bound with respect to the parameters. The latter M-step is equivalent to finding the MLE using expected sufficient statistics under the variational distribution. We consider the maximization step for each parameter in turn.

Notice that a closed form solution for the approximate maximum likelihood estimate of  $\alpha$  does not exist, we use a linear-time Newton-Raphson method, where the gradient and Hessian are

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

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

Similar update rules can be derived for  $\alpha'$ .

In the following, we derive the maximum likelihood estimation of Hawkes hyper-parameters of our LDA-Hawkes model. Hawkes hyper-parameters include the base intensity  $\mu \in \mathbb{R}_+^M$  and the degree of *influence*  $\beta \in \mathbb{R}_+^M$ , where  $\mathbb{R}_+$  denotes the nonnegative real domain. Similar as the case of the LDA hyper-parameters, the MLE for the Hawkes hyper-parameters are obtained by optimizing the lower bound as in Eqn (3) upon the convergence of the variational inference, which leads to the following update formulas:

$$\beta_m = \frac{1}{b_m} \sum_{n=1}^{N_m} \sum_{l=1}^{n-1} r(\phi_{m,l}, \phi_{m,n}) \eta_{m,ln}, \quad \mu_m = \frac{1}{T} \sum_{n=1}^{N_m} \eta_{m,nn}.$$

Our mean-field variation inference algorithm, named LDA-Hawkes, is intuitively interpretable. This algorithm has two loops. The inner loop 1) infers the label/topic distribution  $\gamma$  in each user's query sequence, based on both the topic prior and the topic assignment of each query; 2) infers the word distribution  $\rho$  in each topic, based on both the word prior and the topic assignment of each word; 3) and clusters queries accordingly. The query clustering process as in Eqn (5) assigns queries to different topics based on not only the textual information of queries, but also the temporal pattern underlying the corresponding query sequence. As marked in the formula, the query-topic assignment of each query  $n$  is inferred by integrating four types of evidences: (a) Users' label/topic distribution; (b) Semantic clustering of queries; (c) Past influence: how labels of queries in the past affect the label of the current query; (d) Future influence: how the label of the current query affect labels of queries in the future. The outer loop estimates our model's hyper-parameters, both LDA parameters interpreted as topic/word priors and Hawkes parameters that capture the temporal pattern of the query submission of each user.

In our mean-field variation inference algorithm, the computational cost of inferring variational latent variables is  $O((\sum_m N_m) * K * C)$ , where  $C$  is the average number of words in a query. The computational cost of the estimation of LDA hyper-parameters is  $O(K + V)$ . The computational cost of the estimation of Hawkes hyper-parameters is  $O(\sum_m N_m^2)$ . However, the expected computational cost of this estimation is  $O(\sum_m N_m)$  only. The reason is that, a pair of queries have two many other queries issued between them, usually no *influence* exists between this pair of queries. Thus each query can only be associated to limited number of *influence* links.

## 4. EXPERIMENTS

We evaluated our LDA-Hawkes model on both synthetic and real-world data sets, and compared the performance with the following:

- two alternative LDA-based probabilistic models:

**Time-Window(TW):** This model assumes queries belong to the same search task only if they lie in a fixed or flexible time window, and uses LDA to cluster queries into topics based on the query co-occurrences within the same time window. We tested time windows of various sizes.

**Word-Related:** This model assumes queries belongs to the same search task only if they share at least one word, and uses LDA to cluster queries into topics based on the co-occurrences of queries that sharing at least one word.

- two state-of-the-art query clustering approaches:

**Session-Similarity[30]:** This method evaluated query similarity based on both query sessions and query content, and used those similarity scores for query clustering.

**GATE[2]:** This is a Greedy Agglomerative Topic Extraction algorithm. It extracted topics based on a pre-defined topic similarity function, which considered both semantic similarity and mission similarity. Here mission similarity refers to the likelihood that two queries appear in the same mission, while missions are sequences of queries extracted from users' query logs through a mission detector.

- and three state-of-the-art search task identification approaches:

**Bestlink-SVM [25]:** This method identified search task using a semi-supervised clustering model based on the latent structural SVM framework. A set of effective automatic annotation rules were proposed as weak supervision to release the burden of manual annotation.

**QC-HTC/QC-WCC [20]:** This series of methods viewed search task identification as the problem of best approximating the manually annotated tasks, and proposed both clustering and heuristic algorithms to solve the problem. QC-WCC conducted clustering by dropping query-pairs with low weights, while QC-HTC considered the similarity between the first and last queries of two clusters in agglomerative clustering.

**Reg-Classifier[16]:** This method designed a diverse set of syntactic, temporal, query log and web search features, and used them in a logistic regression model to detect search tasks.

### 4.1 Synthetic data

**Data Generation.** Given parameters  $(M, N, K, \alpha, \alpha', \mu, \beta)$ , the synthetic data is sampled according to the proposed generative model. We record the sampled values of  $Y$ , and calculate the ground-truth

*influence*  $R$  for evaluating the accuracy of our prediction of *influence* among queries. Notice  $\mu$  and  $\beta$  are both vectors of size  $M$ , where each element  $\mu_m$  and  $\beta_m$  is randomly generated in  $[0.5\hat{\mu}, 1.5\hat{\mu}]$  and  $[0.5\hat{\beta}, 1.5\hat{\beta}]$  respectively before simulation. Vectors  $\alpha$  and  $\alpha'$  are of size  $K$  and  $V$  respectively, where each element  $\alpha_k$  and  $\alpha'_v$  is generated in  $[0.5\hat{\alpha}, 1.5\hat{\alpha}]$  and  $[0.5\hat{\alpha}', 1.5\hat{\alpha}']$  respectively before simulation.

Our synthetic data are simulated with two different settings:

- **Small:**  $M = 100$ ,  $N = 120$ ,  $K = 10$ ,  $\hat{\mu} = 0.01$ ,  $\hat{\beta} = 0.5$ ,  $\hat{\alpha} = 0.1$ ,  $\hat{\alpha}' = 0.1$ . Simulations were run 1,000 times using the pre-generated parameters  $\mu, \beta$ ;
- **Large:**  $M = 10,000$ ,  $N = 10,000$ ,  $K = 50$ ,  $\hat{\mu} = 0.01$ ,  $\hat{\beta} = 0.5$ ,  $\hat{\alpha} = 0.1$ ,  $\hat{\alpha}' = 0.1$ . Simulations were run 10 times.

To test the robustness of our method, we add two types of noise to the original synthetic data:

**Event Noisy:** We generate additional 10% of total number of queries randomly in the time window of each already sampled query sequence, and add them to the sequence;

**Intensity Noisy:** Instead of using  $\lambda(t)$  to simulate the query occurrence at time  $t$ , we use a noisy value  $\lambda'(t)$ , which is obtained by adding Gaussian noise on  $\lambda(t)$ :

$$\lambda'(t) = \max(0.1 * e + 1, 0) * \lambda(t), \quad e \sim \mathcal{N}(0, \sigma). \quad (6)$$

The default value of  $\sigma$  is set to be 1.

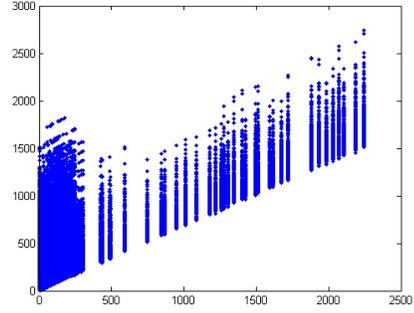
**Inference and Estimation.** Table 1 evaluates both training likelihood, and the accuracy of our proposed variational inference algorithm in parameter estimation and latent variable inference on the synthetic data. We can find that, on the small synthetic data, LDA-Hawkes can recover the Hawkes parameters  $\mu$  and  $\beta$  very well, which represent users' personal temporal patterns of query submission. Meanwhile, based on the inferred query-topic membership  $\hat{Y}$ , we predict the *influence*  $\hat{R}$  among queries, and compare with the ground-truth *influence*  $R$  to evaluate the accuracy of our *influence* prediction through:

$$\text{Proc}_R = \sum_m \frac{1}{N_m(N_m - 1)/2} \sum_{n=1}^{N_m} \sum_{n'=n+1}^{N_m} I(R_{n,n'} = \hat{R}_{n,n'}).$$

Results in Table 1 show that LDA-Hawkes can accurately predicts *influence*. We also find an interesting phenomenon that the accuracy of our estimated Hawkes parameters and the accuracy of our predicted *influence* are highly correlated, since given different predicted *influence*  $\hat{R}$ , the optimal parameters  $\mu$  and  $\beta$  that maximize the likelihood of Hawkes processes on a query sequence can be very different. On the large synthetic data, LDA-Hawkes's performance on parameter estimation becomes worse, while the accuracy of *influence* prediction also decreases. Due to the shapely increased data size, the combination of textual and temporal information becomes more complicated, which makes *influence* prediction more difficult, and further affects the learning of users' personal temporal patterns. On both noisy data set, LDA-Hawkes's performances in both inference and estimation become worse.

## 4.2 Real-world Data

We also conducted extensive experiments on two real-world data sets. The first data set is adapted from the query log of AOL search engine [4]. The entire collection consists of 19.4 million search queries from about 650,000 users over a 3-month period. We cleaned the data by removing the duplicated queries which were submitted consecutively within 1 minute. We randomly selected a subset of users who submitted over 1,000 queries during this period,



**Figure 4: Q-Q plot of the predictive query sequence simulated with inferred Hawkes parameters versus the real query sequence.**

and collected their corresponding search activities, including the anonymized user ID, query string, timestamp, the clicked URL. As a result, we collected 1,786 users with 2.2 million queries, and their activities span from 18 days to 3 months. The second data set is collected from Yahoo search engine, from Jan 2013 to September 2013. Similarly, we cleaned the data and randomly selected a subset of users who submitted over 3,000 queries during this period. As a result, we collected 1,475 users with 1.9 million queries, and their activities span from 54 days to 9 months.

**Model Fitness.** Table 2 shows the log predictive likelihood on events falling in the final 10% of the total time of query data. According to Table 2, LDA-Hawkes fits both synthetic and real-world data better than TW and Word-Related. This illustrates that a Hawkes process better utilizes the temporal information in benefiting LDA's learning of textual data than simply considering the co-occurrence of queries within a time session or queries sharing at least one same word. The larger time-window TW use, the worse its performance will be. Time-window based LDA models generally perform better than Word-Related. Word-Related performs the worst, which illustrates using lexicon-similarity only is far from enough for grouping semantically related queries. On both noisy data sets, the performance of all models become worse. However, the decrease of the performance of LDA-Hawkes is smaller than that of TW and Word-Related, which demonstrates the robustness of our proposed model.

In addition, another experiment is conducted to study how well the proposed model can fit the temporal data of query logs. Figure 4 shows the Q-Q plot of the predictive query sequences based on Hawkes parameters inferred from AOL versus the real query sequence in AOL. If the distribution of the timestamps of the predictive query sequences and that of the real query sequence are similar, the points in the Q-Q plot will approximately lie around the diagonal. If these two distributions are linearly related, the points in the Q-Q plot will approximately lie on a line, but not necessarily on the diagonal. From Figure 4, we can find that LDA-Hawkes fits the temporal data of real-world query logs very well.

**Query clustering.** Along with search task identification, the proposed model simultaneously clusters queries into topics, and automatically labels identified search tasks, thus the performance of identifying and labeling search tasks mainly depends on how we cluster query words into different topics. In this series of experiments, we evaluate the quality of obtained query clusters/topics, which depends on their purity, or semantic coherence. Since no ground truth about the correct composition of a topic is available, we assess purity by the average similarity of each pair of queries

**Table 1: Inference and Estimation of LDA-Hawkes on Synthetic data**

Data set	$\frac{1}{M} \sum_m \frac{ \mu_m - \mu_m }{\mu_m}$	$\frac{1}{M} \sum_m \frac{ \beta_m - \beta_m }{\beta_m}$	Prec <sub>R</sub>	log likelihood
Small Synthetic	0.058	0.204	0.9175	-92.38
Small Event Noisy	0.083	0.317	0.8847	-95.02
Small Intensity Noisy	0.101	0.362	0.8675	-96.80
Large Synthetic	0.174	0.381	0.8573	-115.29
Large Event Noisy	0.202	0.413	0.8291	-119.38
Large Intensity Noisy	0.219	0.436	0.8107	-122.25

**Table 2: Log Predictive Likelihood on Both Synthetic and Real-world Data**

Model/Data set	LDA-Hawkes	TW(5 min)	TW(1 hour)	TW(1 day)	TW(1 week)	Word-Related
Small Synthetic	-110.32	-121.87	-124.08	-137.21	-168.40	-504.83
Small Event Noisy	-122.83	-135.23	-139.37	-152.15	-184.50	-536.21
Small Intensity Noisy	-127.36	-139.21	-146.59	-159.42	-192.23	-543.19
Large Synthetic	-163.84	-177.48	-182.43	-198.20	-239.04	-846.14
Large Event Noisy	-179.34	-193.05	-200.13	-221.49	-263.91	-880.04
Large Intensity Noisy	-184.27	-198.30	-207.23	-228.91	-270.92	-889.36
AOL	-153.12	-165.03	-169.83	-184.27	-221.32	-815.42
Yahoo	-192.36	-217.32	-222.95	-236.03	-275.74	-896.17

within the same topic as:

$$\text{Purity} = \frac{1}{K} \sum_k \frac{\sum_{q_i, q_j \in t_k} \text{Sim}(q_i, q_j)}{N_k(N_k - 1)/2} * 100\%,$$

where  $N_k$  is the number of queries in topic  $k$ .

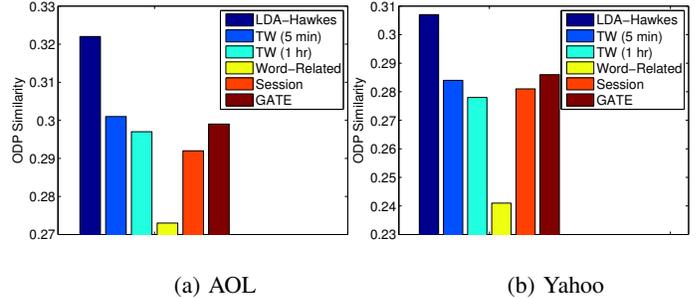
We evaluate the query similarity based on their categorical labels from the Open Directory Project (ODP)<sup>4</sup>. The ODP is a human-edited directory of more than 4 million URLs. These URLs belong to over 590,000 categories organized in a tree-structured taxonomy where more general topics are located at higher levels. For instance, the URL {tech.groups.yahoo.com/group/amrc-l/} belongs to Top/Arts/Animation/Anime/Clubs\_and\_Organizations, while the URL {http://valleyofazure.tripod.com/} belongs to another directory Top/Arts/Animation/Anime/Characters. Hence, to measure how related two queries are, we can use a notion of similarity between the corresponding categories provided by ODP. In particular, we measure the similarity between category  $C_i$  of query  $q_i$  and category  $C_j$  of query  $q_j$  as the length of their longest common prefix  $P(C_i, C_j)$  divided by the length of the longest path between  $C_i$  and  $C_j$ . More precisely, we define this similarity as:

*ODP Similarity*

$$\text{Sim}(q_i, q_j) = |P(C_i, C_j)| / \max(|C_i|, |C_j|),$$

where  $|C|$  denotes the length of a path. For instance, the similarity between the two queries above is 3/5 since they share the path “Top/Arts/Animation” and the longest one is made of five directories. We evaluate the similarity between two queries by measuring the similarity between the most similar categories of the two queries, among the top 5 answers provided by ODP.

Figure 5 compares the purity of topics detected by LDA-Hawkes, alternative probabilistic models, and state-of-the-art query clustering approaches on AOL and Yahoo data sets. We can find that LDA-Hawkes outperforms all compared approaches. It improves over the second best method by up to 10%. Gate and TW(5 min) take the second place, both of them are slightly better than Session-Similarity and TW(1 hr), which again demonstrates that a small time window better benefits the LDA model in de-



**Figure 5: Query Clustering measured by Topic Purity. This metric relies on ODP Similarity to evaluate the pairwise similarity between queries.**

tecting semantically related queries. Word-Related performs significantly worse than other methods, which shows that considering only the co-occurrence of queries sharing words is very limited. Meanwhile, we find that compared with TW, LDA-Hawkes, Session-Similarity, and Gate perform relatively better on Yahoo data set, which implies that LDA-Hawkes works for various real-world query logs. Notice that the absolute value of topic purity is not very high, since the ODP categories are fine-grained, the categories of queries from the same search task are very likely to be different, but share paths, i.e. have common prefix.

**Search Task Identification.** To justify the effectiveness of the proposed model in identifying search tasks in query logs, we employ a commonly used AOL data subset<sup>5</sup> with search tasks annotated, and also recruit eight editors to annotate search tasks in a chosen subset from the Yahoo data, which contains 100 users with around 50 queries per user. We measure the performance by a widely used evaluation metric,

*F<sub>1</sub> score*

$$F_1 = \frac{2 * p_{pair} * r_{pair}}{p_{pair} + r_{pair}},$$

where  $p_{pair}$  denotes the percentage of query-pairs in our predicted search tasks that also appear in the same ground-truth task, while  $r_{pair}$  denotes the percentage of query-pairs in the ground-truth tasks that also appear in the same predicted task.

<sup>4</sup><http://www.dmoz.org/>

<sup>5</sup>[http://miles.isti.cnr.it/tolomei/?page\\_id=35](http://miles.isti.cnr.it/tolomei/?page_id=35).

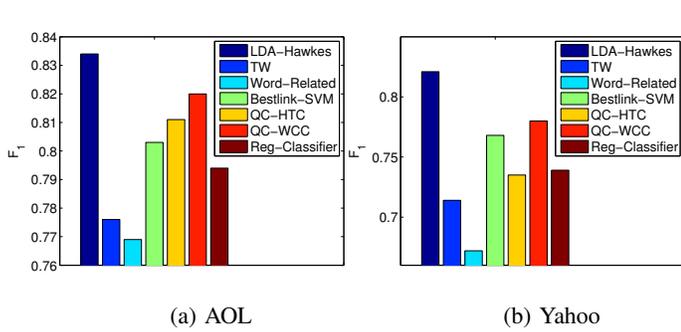


Figure 6: Performance Comparison of Search Task Identification measured by  $F_1$  Score.

Figure 6 compares the proposed model with alternative probabilistic models and state-of-the-art search task identification approaches by  $F_1$  score. Here among TW models with various time-window size, we only include the “5 min” sized Time-Window in comparison, since it performs the best in both model fitness and query clustering. From Figure 6, we find that LDA-Hawkes performs the best among all compared approaches, and outperforms the second best approach by over 5%. Furthermore, LDA-Hawkes outperforms baselines in terms of both accuracy and recall. TW and Word-Related perform the worst since their assumptions on query-relationship within the same search task are too strong. Moreover, LDA-Hawkes’s advantage over Bestlink-SVM and Reg-Classifier illustrates that employing self-exciting point processes like Hawkes to utilize the temporal information in query logs can be a better choice than incorporating temporal information in features. The advantage over QC-HTC and QC-WCC demonstrates that appropriate usage of temporal information in query logs can even better reflect the semantic relationship between queries, rather than exploiting it in some collaborative knowledge.

**Case Study of Identified Search Tasks.** In this part, we try to show a few examples that LDA-Hawkes identify and label search tasks in Yahoo query log. From Figure 7, we can find that both the word co-occurrence and temporal gap play a important role in predicting *influence* among sequential queries. Although chances are very small that queries “aircraft carrier” and “aircraft carrier” will co-occur, we predict an *influence* between them, since they are temporally close. On the other hand, query-pair “tibet” and “ryukyu islands”, and query-pair “aircraft carrier” and “battleships us” are not consecutive, however, we predict that *influence* exist between those pairs of queries, as they co-occur in quite a few number of users’ query sequences. Thus we may conclude that the existence of *influence* demands both temporal and semantic closeness. Queries linked by *influence* belong to the same search task since the user’s *information need* is not satisfied by the former query, which makes the user additionally issue the later semantically related query, whose occurrence violates that user’s regular query submission frequency. The figure also shows that LDA-Hawkes is able to assign the same label to different search tasks which are semantically related, despite that the temporal gap between them are very long.

## 5. RELATED WORK

Search query logs have been extensively studied to improve the search relevance and provide better user experience. There has been a large body of work focused on the problem of identifying search tasks or sessions from sequences of queries. Many of these methods use the idea of a “timeout” cutoff between queries, where two consecutive queries are considered as two different sessions or tasks if the time interval between them exceeds a certain threshold.

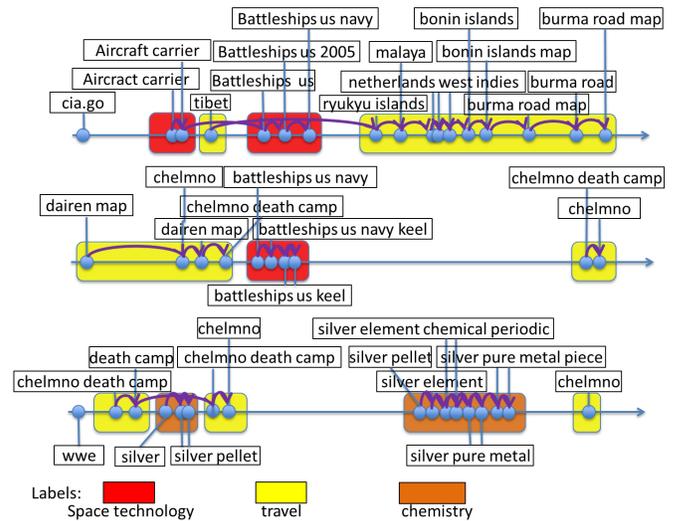


Figure 7: Case Study: Purple arrow line denotes the *influence* identified by the proposed model, rounded rectangle denotes the identified search tasks, rectangle denotes the labels our model assigns to search tasks.

Often a 30-minute timeout is used to segment sessions [8, 19, 25]. In addition, other timeout thresholds have been proposed, from 1 to 120 minutes [13, 16, 20]. However, the experimental results of these methods indicate that the timeouts, whatever their lengths, are of limited utility in predicting whether two queries belong to the same task, and unsuitable for identifying session boundaries. Beyond that, Wang et al. [25] and Hua et al. [15] treated the time intervals between queries as pairwise features in their models. But no previous work has explicitly exploited the temporal information directly in their models. In our work, we directly integrate the temporal information into our model, rather than highly relying on different timeouts, for identifying search tasks.

There have been attempts to extract in-session tasks [23, 16, 20], and cross-session tasks [16, 17, 1, 25] from query sequences based on classification and clustering methods. Jones and Klinkner [16] proposed to learn a binary classifier to detect whether two queries belong to the same task or not, which organized and segmented query sequences into hierarchical units. Moreover, Kotov et al. [17] and Agichtein et al. [1] studied the problem of cross-session task extraction via binary same-task classification, and found different types of tasks demonstrate different life spans. Another suitable mechanism for identifying sessions or tasks may rely on *unsupervised learning* approaches, i.e., query clustering algorithms, especially when no labeled training set is available. The intuition for using query clustering is based on the assumption that if two queries belong to the same cluster, then they are topically related. Cao et al. [8] proposed a clustering algorithm for summarizing queries into concepts throughout a click-through bipartite graph built from a search log. Lucchese et al. [20] and Hua et al. [15] exploited the knowledge base for detecting semantically related query pairs that are not similar from a lexical content point of view. In addition, Wang et al. [25] proposed a semi-supervised clustering method for identifying cross-session tasks. Different from these existing methods, our paper assumes that queries belonging to the same search task are linked by influence. Moreover, instead of focusing on the query sequence of single users, we take into account the query sequences issued by different users simultaneously in a unified framework, such that our model can identify and label coherent search tasks across users.

Our proposed model is closely related to point processes, which have been used to model social networks [7] and natural events [31]. People find self-exciting point processes naturally suitable to model continuous-time events where the occurrence of one event can affect the likelihood of subsequent events in the future. One important self-exciting process is Hawkes process, which is first used to analyze earthquakes [21, 31], and then widely applied to many different areas, such as market modeling [11, 3], crime modeling [24], terrorist [22], conflict [29], and viral videos on the Web [9]. To solve such models, an EM algorithm is generally adopted to estimate the maximum likelihood of Hawkes process [18].

## 6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel probabilistic model to integrate the LDA model with Hawkes process for identifying and labeling search tasks. Basically, Hawkes processes utilize its self-exciting property to identify search tasks if influence exists among a sequence of queries for individual users, while the LDA model exploits query co-occurrence across different users to discover the latent information needed for labeling search tasks. By leveraging the temporally weighted query co-occurrence, our model not only guarantees sound performance by making full use of both textual and temporal information of the entire query sequences, but also enables the labeling of the identified search tasks since semantically related queries are clustered together through query links determined by co-occurrence. We have applied the proposed LDA-Hawkes model to analyze search tasks on both AOL and Yahoo query logs, and compare with several alternative approaches. Experimental results show that the improvements of our proposed model are consistent, and our LDA-Hawkes model achieves the best performance. In future work, it would be interesting to consider other information, e.g., click-through data, into this framework, and investigate the performance of LDA-Hawkes in other domains.

## 7. REFERENCES

- [1] E. Agichtein, R. W. White, S. T. Dumais, and P. N. Bennett. Search, interrupted: understanding and predicting search task continuation. In *SIGIR*, pages 315–324, 2012.
- [2] L. M. Aiello, D. Donato, U. Ozertem, and F. Menczer. Behavior-driven clustering of queries into topics. In *CIKM*, pages 1373–1382, New York, NY, USA, 2011. ACM.
- [3] Y. Ait-Sahalia, J. Cacho-Diaz, and R. Laeven. Modeling financial contagion using mutually exciting jump processes. *Tech. rep.*, 2010.
- [4] AOL. <http://gregsadedtsky.com/aol-data/>.
- [5] D. Blei and M. Jordan. Variational inference for dirichlet process mixtures. In *Bayesian Analysis*, volume 1, pages 121–144, 2005.
- [6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, March 2003.
- [7] C. Blundell, K. A. Heller, and J. M. Beck. Modelling reciprocating relationships with hawkes processes. *NIPS*, 2012.
- [8] 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.
- [9] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, 105(41):15649–15653, 2008.
- [10] 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*, pages 1–38, 1977.
- [11] E. Errais, K. Giesecke, and L. R. Goldberg. Affine point processes and portfolio credit risk. *SIAM J. Fin. Math.*, 1(1):642–665, Sep 2010.
- [12] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58:83–90, 1971.
- [13] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *IPM*, 38(5):727–742, 2002.
- [14] M. D. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *NIPS*, 2010.
- [15] W. Hua, Y. Song, H. Wang, and X. Zhou. Identifying users’ topical tasks in web search. In *WSDM*, pages 93–102, 2013.
- [16] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *CIKM*, pages 699–708, 2008.
- [17] A. Kotov, P. N. Bennett, R. W. White, S. T. Dumais, and J. Teevan. Modeling and analysis of cross-session search tasks. In *SIGIR*, pages 5–14, 2011.
- [18] E. Lewisa and G. Mohlerb. A nonparametric em algorithm for multiscale hawkes processes. *Journal of Nonparametric Statistics*, 1, 2011.
- [19] Z. Liao, Y. Song, L.-w. He, and Y. Huang. Evaluating the effectiveness of search task trails. In *WWW*, 489–498, 2012.
- [20] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Identifying task-based sessions in search engine query logs. In *WSDM*, pages 277–286, 2011.
- [21] Y. Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association.*, 83(401):9–27, 1988.
- [22] M. D. Porter and G. White. Self-exciting hurdle models for terrorist activity. *The Annals of Applied Statistics*, 6(1):106–124, 2011.
- [23] A. Spink, S. Koshman, M. Park, C. Field, and B. J. Jansen. Multitasking web search on vivisimo.com. In *ITCC (2)*, pages 486–490, 2005.
- [24] A. Stomakhin, M. B. Short, and A. L. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems.*, 27(11), 2011.
- [25] H. Wang, Y. Song, M.-W. Chang, X. He, R. W. White, and W. Chu. Learning to extract cross-session search tasks. In *WWW*, pages 1353–1364, 2013.
- [26] X. Wang and E. Grimson. Spatial latent dirichlet allocation. In *NIPS*, 2007.
- [27] Y. Wang, E. Agichtein, and M. Benzi. Tm-lda: efficient online modeling of latent topic transitions in social media. In *KDD*, pages 123–131, New York, NY, USA, 2012. ACM.
- [28] R. W. White, W. Chu, A. Hassan, X. He, Y. Song, and H. Wang. Enhancing personalized search by mining and modeling task behavior. In *WWW*, pages 1411–1420, 2013.
- [29] A. Z.-Mangion, M. Dewarc, V. Kadirkamanathand, and G. Sanguinetti. Point process modelling of the afghan war diary. *PNAS*, 109(31):12414–12419, July 2012.
- [30] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *WWW*, pages 1039–1040, New York, NY, USA, 2006. ACM.
- [31] J. Zhuang, Y. Ogata, and D. V. Jones. Stochastic declustering of space-time earthquake occurrences. *Journal of the American Statistical Association.*, 97(458):369–380, 2002.