Table 1: Search Behavior Features

Due to space limitation, we avoid showing accumulated features that are based on similar search actions with instant features.

Feature q	Description
Click Number	The total number of clicks on the returned results of a submitted query.
Dwell Time	The average time between a user's final action on the last query and the submission of the current query.
Click Position	The average position of clicks on the results of a query.
Time Duration	The time duration of a query.
Click Speed	The number of clicks divided by the time duration of a query.
Scanned Pages	The number of result pages user scanned for an issued query.
Time Interval	The time interval between the submission of current query and that of the next query.
Query Number	The total number of queries within the search task that current query belongs to.
Click Number Per Query(CNPQ)	The average number of clicks per query within the current search task.

words in a query, N is total number of queries, K is the number of *search factor*s, M is the dimension of search behavior features, T is the number of topics, and V is the vocabulary size. Notice that in computing γ we take the advantage of the sparsity of W which contains only $N * \bar{C}$ nonzero elements. The computational cost of the estimation of hyper-parameters is $O(K * T^2 + T * V)$. The computational cost of the estimation of the parameters of *search factors* is $O(M * K * N)$. Thus the total computational cost of our algorithm is $O(N * (K * T^2 + K * M + T * \bar{C} * V))$, where $M < T^2$ can by ensured by controlling the number of search factor features we use. Also notice that $\bar{C}$ is a small constant, and we have $V \ll N$ when the number of queries N is large enough, thus the total computational cost can be simplified to $O(N * K * T^2)$, which is linear in the number of queries with a fixed number of *search factor* patterns K, a fixed number of states/topics T.

4. SEARCH BEHAVIOR FEATURES

As the search behavior features are very valuable and useful, we experimented with many search behavior features besides those popular query content features utilized in existing approaches. The objective is to capture some key search behaviors which may influence a user's choice of the topic of his/her next query given the topic of the current issued query. We summarize these search behavior features in Table 1. Our features generally originate from statistical counting of search engine users' basic actions, such as issuing query, clicking URL, turning page. Those actions are measured by counting the number or time for each issued queries, resulting in the following features. For each query, we simply measure how many URLs are clicked (denoted as "Click Number"), and how many seconds it lasts (denoted as "Time Duration"). Moreover, for each query, we further calculate some complex statistical features, including the average time interval between each click and its following click (denoted as "Dwell Time"), the average position of clicks (denoted as "Click Position"), the number of clicks divided by the time duration (denoted as "Click Speed"), and the average number of result pages scanned by user (denoted as "Scanned Pages").

Besides features that describe users' behaviors on each single query, we also make use of users' accumulated behaviors on the search task that the current query belongs to. Those behaviors also originate from search engine users' basic actions, but are measured under the scale of search tasks instead of queries. For instance, "Click Number" measures the total number of clicks completed in the current search task, "Click Position" measures the average position of clicks in the current search task. In addition, some accumulated-behavior features collect novel information compared with instant-behavior features. For example, "Query Number" measures how many queries are already issued in the current search task, and "CNPQ" calculates the number of clicks divided by the number of the issued queries in the current search task. Notice the update of topic membership of each query can change the value of accumulated features from iteration to iteration. Based on those collected features, we are able to form a feature vector $\mathbf{d}_n$ for each query n, which is used for latent variable inference and parameter estimation in GHSMM.

5. EXPERIMENTS

In this section, we first describe alternative Markov models and search task identification approaches as our baseline models. Then we evaluate our proposed GHSMM model and compare it with the baseline models on real world data. Experimental results demonstrate the effectiveness of our proposed model.

5.1 Baseline Models

To evaluate the effectiveness of the proposed GHSMM model, we compare it with the following alternative Markov models:

HMM[25]: This approach is based on hidden Markov model, which assumes the topic of the next query depends only on the topic of the present query;

HSMM[6]: This approach is based on hidden semi-Markov model, which assumes the topic of the next query depends on both the topic of the present query and the time a user spends on the present query.

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

Bestlink-SVM [28]: This method identified search tasks 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 [23]: 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[18]: 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.

LDA-Hawkes[20]: This method casted search task identification into the problem of identify semantic influence in observed query sequences, and proposed a probabilistic model by combining LDA model with Hawkes processes to address the problem using both temporal and textual information.

In addition, since the performance of search task identification depends heavily on the quality of learned topics, we also compare the proposed model with two state-of-the-art query clustering approaches:

Session-Similarity[35]: This method evaluated query similarity based on both query sessions and query content, and used those similarity scores for query clustering.