Towards Hierarchies of Search Tasks & Subtasks

Rishabh Mehrotra
University College London
r.mehrotra@cs.ucl.ac.uk

Emine Yilmaz
University College London
emine.yilmaz@ucl.ac.uk

ABSTRACT

Current search systems do not provide adequate support for users tackling complex tasks due to which the cognitive burden of keeping track of such tasks is placed on the searcher. As opposed to recent approaches to search task extraction, a more naturalistic viewpoint would involve viewing query logs as hierarchies of tasks with each search task being decomposed into more focussed sub-tasks. In this work, we propose an efficient Bayesian nonparametric model for extracting hierarchies of such tasks & subtasks. The proposed approach makes use of the multi-relational aspect of query associations which are important in identifying query-task associations. We describe a greedy agglomerative model selection algorithm based on the Gamma-Poisson conjugate mixture that take just one pass through the data to learn a fully probabilistic, hierarchical model of trees that is capable of learning trees with arbitrary branching structures as opposed to the more common binary structured trees. We evaluate our method based on real world query log data based on query term prediction. To the best of our knowledge, this work is the first to consider hierarchies of search tasks and subtasks.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval—*Search Process*

Keywords

Search tasks; Bayesian Nonparametrics; Hierarchies

1. INTRODUCTION

Contemporary search environments are tailored to support a small set of basic search tasks and provide searchers with few options to search and interact with information, and little to help them synthesize and integrate information across sessions. Search behavior, and information behavior more generally, is often motivated by tasks that prompt search processes that are often lengthy, iterative, and intermittent, and are characterized by distinct stages, shifting goals and multitasking.

A complex search task like planning a vacation involves sub-tasks like booking fights, finding hotels and looking up places of interests among others. Each of these subtasks themselves warrant issuing queries by users to accomplish them. As a result, accomplishing complex search tasks places intense cognitive burden on the user to explore and discover varied aspects of the task and necessitates the user to (i) explore the domain-space, (ii) identify the necessary sub-tasks involved and (iii) issue queries to accomplish these sub-tasks. Such a multiple stage process becomes prohibitively challenging for searchers who might have little to no domain knowledge of the task they're trying to accomplish.

Prior work on identifying search-tasks mainly explores task extraction from search sessions [6][2], wherein the objective is to segment a search session into disjoint sets of queries where each set represents a different task. These approaches do not aggregate tasks across users, thus cannot combine or differentiate between tasks extracted from different user sessions. Prior work on identifying cross-session tasks has targeted pairs of queries, and made predictions about whether they share the same goal or represent the same task[3]. Unfortunately, pairwise predictions alone cannot generate the partition of tasks, and post-processing is needed to obtain the final task partitions[5]. Finally, authors in [4] 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.

While existing search engines are adept at handling simple information seeking needs spanning single or a session full of queries, users get little or no help when their information need transcends this boundary. This major limitation in existing task-extraction methods stems from their treatment of search tasks as a flat structure-less clusters which inherently lack insights about the presence or demarcation of subtasks associated with individual search tasks. A more naturalistic viewpoint would involve considering query logs as hierarchies of tasks & subtasks with each search task being decomposed into more focussed sub-tasks. Additionally, a hierarchically constructed task-subtask system would help in reducing searcher's cognitive burden by assisting in (i) identifying and later (ii) accomplishing the sub-tasks associated with complex tasks undertaken by the searcher.

To this end, we propose a Bayesian nonparametric model for extracting task/sub-task hierarchies as described below.

2. EXTRACTING TASK HIERARCHIES

We consider the challenge of extracting hierarchies of search tasks and their associated subtasks from a given search log given just the log data without the need of any manual annotation of any sort. We present an efficient Bayesian nonparametric model for discovering task hierarchies and propose a tree based bayesian hierarchical task construction algorithm

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742777.

to discover this rich hierarchical structure embedded within search logs. We extend the Bayesian Hierarchical Community Detection model[1] and formulate the task extraction problem as finding a hierarchical tree of queries.

Tree as partitions & mixtures

Our model organises the queries into a nested hierarchy T of tasks/subtasks, with all queries in one node at the root and singleton queries at the leaves. We interpret a tree (T) as a mixture of partitions over those group of queries (Q). We define the probability of a group of such queries as: $p(Q|T) = \sum_{\phi} p(\phi(t))p(Q|\phi(t))$ where $p(\phi(T))$ is the mixing proportion of partition $\phi(T)$, and $p(Q|\phi(t))$ is the probability of the group of queries Q given a partitioning by $\phi(T)$. In general the number of partitions consistent with T can be exponentially large. To make computations tractable, we define the mixture model in such a way that $p(Q|\phi(t))$ can be computed using dynamic programming over T:

$$p(Q|T) = \pi_T f(Q) + (1 - \pi_t) \prod_{T_i \in ch(T)} p(leaves(T_i)|T_i) \quad (1)$$

Gamma-Poisson Model of Query Affinities

A tree in our setting is comprised of a group of queries which potentially compose a search task. The likelihood of such a tree should encapsulate information about the different relationships which exists between queries. We define three broad categories of query-query affinities: (i) query term based affinity, (ii) query documents based affinity and (iii) user/session information based query affinity. Our goal is to capture information from all three affinities when defining the likelihood of the tree. We assume that the global affinity among a group of queries can be decomposed down into a product of independent terms, each of which represents one of the three affinities among the query-group and place a conjugate Gamma-Poisson conjugate prior to define the marginal likelihood of a tree with a group of queries.

Forming Arbitrary Tree Structure

In the beginning, each query is regarded as a tree on its own. For each step, the algorithm selects two trees T_i and T_j and merges them into a new tree T_m. Unlike binary hierarchical clustering, Blundell *et al.*[1] propose three possible merging operations: **(i) Join**: $T_m = \{T_i, T_j\}$, such that the tree T_m has two children now ; **(ii) Absorb**: $T_m = \{children(T_i) \cup T_j\}$, i.e., the children of one tree gets absorbed into the other tree forming an absorbed tree with >2 children; and **(iii) Collapse**: $T_m = \{children(T_i) \cup children(T_j)\}$, all the children of both the sub-tree get combined together at the same level. Such a setting allows each task to be composed of an arbitrary number of sub-tasks without restricting tasks to contain only binary subtasks.

Agglomerative Model Selection

The problem of learning hierarchy of tasks T is treated as one of greedy model selection: each tree T is a different model, and we wish to find the model that best explains the data. The tree is built in a bottom-up greedy agglomerative fashion, starting from a forest consisting of $|Q|$ trivial trees with one query each. Each iteration then merges two trees in the forest, wherein, each query is a leaf of exactly one tree. The algorithm finishes when just one tree remains. At each iteration a pair of trees in the forest F is chosen to be merged by considering the pair and type of merger that yields the largest **Bayes factor improvement**[1] over the current model.

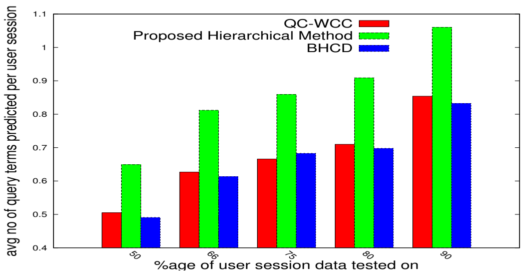

Figure 1: Performance on Query Term Prediction. X-axis represents the %-age of user session data we test on; the rest data was used for matching the task to the task extracted from Session Track data.

3. EXPERIMENTAL EVALUATION

Users engage in search sessions to accomplish search tasks; hence if a system can understand the task well, then, given initial queries, it should be able to predict future queries in the session. Based on this intuition, we base our evaluation on **Query Term Prediction** wherein given initial set of queries of user session, we leverage queries from the identified task to predict future query terms. This is in line with our goal of supporting users tackling complex search tasks since a task identification system which is capable of identifying "*good*" search tasks will indeed perform better in predicting future query terms.

To evaluate the performance of the proposed task extraction method, we work with two datasets: (i) Session Track 2014 (to construct search task hierarchy) and (ii) AOL log data (to extract user sessions used for evaluation). The session track data consists of over 1200 sessions of queries while the AOL logs consists of 20M search queries issued by over 657000 users in about 3 months. We find the intersection set of queries between the Session Track data & AOL logs to identify user sessions in AOL data trying to achieve similar task objectives. We baseline against a standard task identification system (QC-WCC with content similarity) as proposed by Lucchese *et al.*[6] and a very recent hierarchy extraction algorithm as described by Blundell *et al.*[1]. To make fair comparisons, we flatten our hierarchy in a way to obtain similar number of tasks as the baselines methods. We compare the performance on term prediction on AOL user sessions. Our initial results in Figure 1 demonstrate the advantage of considering a hierarchical task extraction system. We intend to extend the evaluation setup in future work.

Overall, we presented a bayesian algorithm to extract hierarchies of search tasks & subtasks. We believe that insights from this work could spark future research in developing richer and generalizable models of search tasks.

4. ACKNOWLEDGEMENTS

This work was supported in part by a Google Faculty Research Award.

5. REFERENCES

[1] C. Blundell and Y. W. Teh. Bayesian hierarchical community discovery. In *NIPS 2013*.
[2] Hua, Song, and Wang. Identifying users' topical tasks in web search. In *ACM WSDM 2013*.
[3] Kotov, Bennett, White, Dumais, and Teevan. Modeling and analysis of cross-session search tasks. In *ACM SIGIR 2011*.
[4] Li, Deng, Dong, Chang, Zha, and Ho. Identifying and labeling search tasks via query-based hawkes processes. In *KDD 2014*.
[5] H. Liao, Song. Evaluating the effectiveness of search task trails. In *ACM WWW 2012*.
[6] Lucchese, Orlando, Perego, Silvestri, and Tolomei. Discovering tasks from search engine query logs. *ACM Transactions on Information Systems (TOIS)*, 2013.