Intent-aware Query Obfuscation for Privacy Protection in Personalized Web Search

Wasi Uddin Ahmad
University of California, Los Angeles
Los Angeles, CA
wasiahmad@ucla.edu

Kai-Wei Chang
University of California, Los Angeles
Los Angeles, CA
kwchang.cs@ucla.edu

Hongning Wang
University of Virginia
Charlottesville, VA
hw5x@virginia.edu

ABSTRACT

Modern web search engines exploit users' search history to personalize search results, with a goal of improving their service utility on a per-user basis. But it is this very dimension that leads to the risk of privacy infringement and raises serious public concerns. In this work, we propose a client-centered intent-aware query obfuscation solution for protecting user privacy in a personalized web search scenario. In our solution, each user query is submitted with l additional cover queries and corresponding clicks, which act as decoys to mask users' genuine search intent from a search engine. The cover queries are sequentially sampled from a set of hierarchically organized language models to ensure the coherency of fake search intents in a cover *search task*. Our approach emphasizes the *plausibility* of generated cover queries, not only to the current genuine query but also to *previous queries in the same task*, to increase the complexity for a search engine to identify a user's true intent. We also develop two new metrics from an information theoretic perspective to evaluate the effectiveness of provided privacy protection. Comprehensive experiment comparisons with state-of-the-art query obfuscation techniques are performed on the public AOL search log, and the propitious results substantiate the effectiveness of our solution.

CCS CONCEPTS

• **Security and privacy → Privacy protections**; • **Information systems** → *Query intent*; *Personalization*;

KEYWORDS

Search privacy, query obfuscation, search tasks

ACM Reference Format:
Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Intent-aware Query Obfuscation for Privacy Protection in Personalized Web Search. In *SIGIR '18: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8-12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209978.3209983

1 INTRODUCTION

Personalization techniques in modern information retrieval systems are a double-edged sword. Search engines trace, analyze and exploit their users' personal information [29, 47] and behavior signals [18, 32, 45, 46] to infer what the users are looking for, when, and in what context, so as to deliver more relevant and customized content or advertising [13, 50]. Although personalization techniques largely increase the utility of provided services on a per-user basis [17], however, it is this very dimension that raises serious public concerns about privacy infringement [12, 48]. According to [36], almost three-quarters of search engine users in the U.S. are *not okay* with their personal information being tracked and used to personalize their future search.

Current solutions for privacy protection in online systems mostly focus on the identifiability aspect of privacy, i.e., who issued the query, via providing secured communication [15, 37], encrypted data storage [10, 35] and releasing [19, 24]. However, another important aspect of privacy, ***linkability***, i.e., *determining the interests of an individual from their observed behaviors*, has not received enough attention. Specifically, linkability is what enables a service provider to link multiple queries to the same user, and thereby learn detailed information about a user's interests. According to Jones et al.'s study in [28], a simple supervised classifier based on the textual query content recorded in search engine logs can link a sequence of queries to a set of candidate users with known gender, age and location; and this set is 300-600 times smaller than a random chance would allow. This leaves the users with little control to avoid "curious" systems abusing their personal information, e.g., targeted advertising [22] and digital discrimination [34].

In this work, we develop intent-aware query obfuscation solution to protect the linkability aspect of privacy in a personalized web search system. In particular, we focus on search log based personalization techniques [17, 41, 46, 47], which have been extensively studied in literature and widely used in practical systems. In such type of personalization methods, search engines profile users with their historical queries and clicks and use such profiles to customize future system output, such as the search results. At a high level, the basic idea behind our solution is to hide a user's true search intents among a set of randomly generated but loosely related cover queries and clicks. In order to compensate the degenerated search quality that is caused by the injected noise, on the client-side, we maintain a noise-free user profile to re-rank the received search results. Our proposed solution is totally client-centered, with no support required from the search engine side, which makes it general and adaptable to many personalized application scenarios, e.g., recommender systems.

The key challenge in this research is to ensure the *plausibility* of the generated cover queries, not only with respect to the user's current query but also to the sequence of his/her previous queries and clicks in the same *search task* [27, 49]. Existing studies in query obfuscation only focus on the first aspect of plausibility. For example, latent semantic analysis [33] and topic models [2] have been used to cluster queries into semantically coherent groups; every

SIGIR'18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3209983

time, cover queries are randomly generated from qualitatively similar but different query groups with respect to the true user query. However, users' true search intents usually span a wide spectrum in practice; for complex needs like health diagnosis, they often submit multiple queries and even continue the search for several days [1]. As a result, the strong dependency between a user's sequential search behaviors leave the genuine query sequence distinctive, as the cover queries are *independently* generated for each query in the same task from those conventional query obfuscation methods.

We address the challenge by generating sequences of related cover queries to form *cover tasks* with respect to a user's gradually developed search tasks. To ensure semantic relatedness within the sequence of cover queries, we sample queries from a set of hierarchically organized language models based on the topic ontology defined in the Open Directory Project (ODP) [44]. To make the sampled queries comparable to the genuine ones in plausibility, we adopt the rejection sampling method [21] to ensure the entropy of sampled queries is close to that of the enuine query. Additionally, as query term matching still plays a very important role in modern search engine [17, 18], we submit the genuine user query together with the cover queries and only return the search results for genuine queries to the user to ensure utility of the search results. As a result, there is a trade-off between privacy protection our solution provides and the utility of search results; and the number of generated cover queries controls such trade-off.

Quantitatively evaluating the effectiveness of privacy protection is also challenging. Previous studies have shown that people do not necessarily reveal their true privacy needs in laboratory experiments and/or questionnaires [14]. And it is even more challenging (if not impossible) for third-parties to judge privacy needs from the logged search history because different users hold different criteria about privacy [5, 12]. In this work, we assume *all* user queries are sensitive and need privacy protection; and we measure the change of entropy between the prior and posterior distributions over the search intents inferred for a user after a genuine search task is finished. An algorithm is considered as a good privacy protection solution if it creates *similar* changes of entropy in the cover queries to those in the genuine queries, i.e., Bayes-optimal privacy [30]. In our empirical evaluations, promising improvement in privacy protection and search quality measured by a distinct set of performance metrics confirm the effectiveness of our proposed solution.

2 RELATED WORK

The inherent tension between personalization and privacy roots in the nature of personalization techniques, which heavily depend on utilizing various types of personal information to differentiate individuals' information need from the general groups' needs [18, 32, 45–47]. Shen et al. [43] envisioned four levels of privacy protection in a personalized retrieval system, e.g., pseudo identity v.s., no identity, and analyzed various software architectures to achieve such a purpose. Their categorization of privacy preservation can be summarized as the identifiability and linkability aspects of privacy, and we discuss the related works to our proposed research with respect to these aspects.

• **Protecting identifiability.** A prodigious amount of research effort has been devoted to protecting this aspect of privacy. Anonymized proxies, e.g., the Tor Project [15], provide a way of concealing users' identities from third-party's monitoring when accessing an online service system. Collaborative schemes are proposed in [11, 16] to let each user submit queries generated by other users, such that individual users are hidden inside groups' identities. However, such solutions suffer from slow response time [39], heavy dependency on the availability of backend system, and ignorance of the semantic content of users' inquiries, which may inherently reveal users' identities. More importantly, such solutions largely disable personalization by physically hiding users' identities.

• **Protecting linkability.** The incident of AOL query log release [7] has demonstrated the risk of privacy breaches from the inquiries received on the server side. Query privacy is achieved in Boolean retrieval by matching the encrypted or hashed keywords in place of their plaintext counterparts [9, 20]. Murugesan and Clifton defined a relaxed problem called "Plausibly Deniable Search" in relevance-based retrieval and proposed a latent semantic indexing based approach to generate cover-up queries to hide a user's original queries [33]. However, it can only obfuscate queries falling into a predefined dictionary, which greatly limits its practical value. Follow-up work further considers the semantic relatedness between the generated queries and genuine query [40], injects decoy terms with similar specificity as the original query terms [35] or according to a given knowledge base [40], and generates scrambled queries [3, 4] by maintaining anonymity and generality of user queries. Zhu and Xiong anonymized simple keyword-based user profiles to protect privacy while facilitating personalized web search [53].

A large pool of prior works are done based on Obfuscation-based private web search (OB-PWS) [6] where dummy queries are generated and sent to the search engine along with users' true queries to prevent accurate inference of search profiles and provide query deniability. Browser extensions, such as TrackMeNot [23], are developed following OB-PWS notion to conceal users' general interests. However, most existing research works anonymizes user queries in an isolated manner without considering the relatedness between consecutive user queries in the same task; the dependency among users' sequential interactions has not been considered. In our solution, we formulate a sequence of related cover queries and clicks with respect to a user's gradually developed search intents. Both query-level and task-level plausibility are achieved via controlled statistical sampling.

3 METHODOLOGY

In this section, we first describe the privacy definition that we focus on in this paper, and then discuss in detail about each component of our developed intent-aware query obfuscation solution, where 1) a sequence of cover queries are sampled from a topic hierarchy regarding a user's gradually developed search intent, and 2) client-side reranking is performed on the true user profile to compensate the degenerated search quality caused by the injected noise.

3.1 Defining Privacy

We focus on Bayes-optimal privacy [30] that bounds the difference between the prior and posterior beliefs of an adversary about a user's private information. Consider in a search scenario, a (malicious) search engine keeps track of search queries and clicks from its users and performs inference of user search intents from such data. To protect a particular user's sensitive information, a privacy protection mechanism has to manipulate the queries and clicks sent to the search engine from this user. Bayes-optimal privacy is

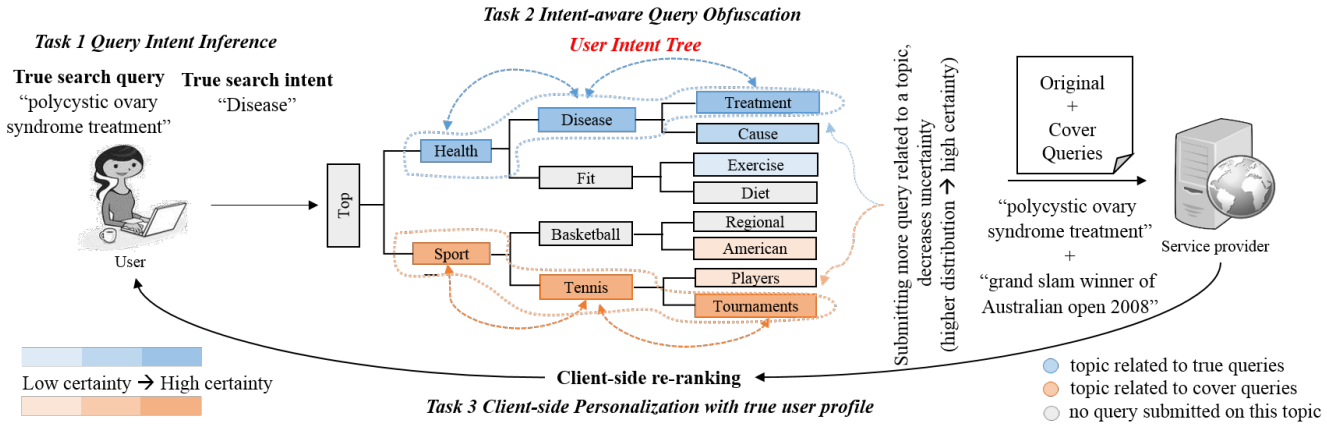


Figure 1: Intent-aware Query-obfuscation framework for Privacy-protection. The dashed arrows represent the generalization and specification operations on the topic tree to generate sequential cover queries with respect to a user's genuine search task.

obtained if the protection mechanism provides an adversary little additional information for inferring a user's sensitive information beyond its background knowledge (such as that from the user's nonsensitive information or public data or logs from other users). To be more specific, Bayes-optimal privacy requires the adversary's posterior belief about the user's sensitive information not to exceed ϵ amount of its prior belief, after observing the manipulated queries and clicks.

However, the original definition of Bayes-optimal privacy is not easy to operate in the search scenario. First, it assumes a clear separation between sensitive and nonsensitive information, as it was originally designed for privacy-preserving data publishing. However, as we discussed in the introduction, privacy is a highly emotional issue and different users might have different privacy requirements. Second, the inference function is originally defined solely as generalization mappings between attribute values. For example, partitioning ordered attributes into intervals and only publishing the intervals. In the search scenario, users submit natural language queries and can click on any of the returned documents, which can hardly be predefined. This leads to unrestricted inference a search engine may perform, and it further complicates the prior and posterior belief estimation.

To make the Bayes-optimal privacy operational in the search scenario, we make two assumptions. First, we assume the *topics* of search queries are sensitive, which indicate a user's (private) search intent. This assumption is practical and serves as the basis for many deployed personalization algorithms [8, 13, 17, 47]. Therefore, the inference a search engine needs to perform is to map queries to topics. To realize the fact that modern search engines utilize a sequence of queries (i.e., search tasks [1, 27, 49]) to infer a user's search intent, we do not limit our inference function to a single query, but to a set of queries. Second, we assume *all* user search topics are sensitive. Although arguably this assumption may not hold in practice, it leads to a stronger privacy protection. The Bayes optimal privacy can then be effectively measured in the space of search topics, i.e., the difference between the prior and posterior distributions of topics inferred from a user's search history. Accordingly, the privacy attack imposed in the resulting threat model is that a search engine performs topical inference about users' private information from their submitted queries and clicks. We do *not* assume the search engine would deliberately provide degenerated search results to force users to provide more sensitive queries and clicks, as it is against its business model.

We should note the difference between our privacy definition and the notion of **plausible deniability** defined in [33]. Specifically, plausibility is defined as the distance in a vector space among queries, and deniability requires all submitted queries to locate within a certain distance to each other so that they are equally plausible. First, plausibility only focuses on individual queries and ignores the dependency among users' sequential search behaviors in the same search task. It thus fails to measure the privacy disclosure resulted from the observable sequential interactions on the search engine side. Second, it does not account for the background knowledge an adversary might have in inference. This will lead to a less effective protection of privacy, e.g., generating cover queries with close distances to each other but highly rare with respect to a global distribution of queries.

We have to admit that defining search privacy at a topic level cannot well handle information leakage from users' "ego-surfing" behavior, e.g., searching for one's own name and SSN. Special treatment can be added to address such situations, such as generating common person names if a person's name is detected in a user query. We leave this as our future work and focus on topic-level privacy protection defined above. In the following, we will discuss our proposed solution for generating the obfuscated user queries and clicks from client-side, which controls a search engine's posterior belief about a user's sensitive information at the topic level, such that the aforementioned Bayes-optimal privacy is obtained.

3.2 Protecting Privacy

We consider the search intent behind a user query as the semantic *topic* embedded in its text content, and the topics are hierarchically organized such that they form an ontology structure named *intent tree*. To accommodate a user's sequential search behaviors, queries serving for the same intent can be grouped into tasks [27], so that a task maps to a sub-graph on the intent tree. Such concept has been extensively exploited in modern search engines to construct *user profiles* to personalize search results [32, 45–47]. The server

inferred user profile represents its posterior belief about the user's sensitive information in this topic space, regarding the definition of Bayes-optimal privacy.

To protect user privacy in this topic space, we introduce controlled noise to the user profiles maintained on the server-side by submitting cover queries and clicks from *related* topics on the intent tree: the obfuscated queries and clicks are generated with respect to the changes detected in the search topics of genuine queries. In this way, the user profile maintained on the server side gets lower perception about the user's developing true intents. As a result, it reduces the difference between the adversary's prior and posterior beliefs about a user's private information.

We name our proposed solution Intent-aware Query-obfuscation for Privacy-protection, or IQP in short. The workflow of IQP is illustrated in Figure 1, where a user interacts with a potentially non-trustworthy search engine. The goal of IQP is to assist users in fulfilling their information need with reduced effort (i.e., *personalization*) while minimizing sensitive information disclosure to the search engine (i.e., *privacy*).

In IQP, the search process works as follows. First, when a user issues a query, the client-side model infers its underlying search topic against the intent tree; then based on the inferred topic, cover queries are generated from related topics with a similar specificity. The specificity of a user's intent is estimated by the depth of inferred topics on the intent tree and the frequency of this query against a reference corpus, which ensemble the prior knowledge a search engine might have. It is easy to prove that when one proportionally samples from those equally specific search topics to generate the cover queries, the difference between the server-side prior and posterior beliefs of a user's sensitive information is minimized (as the relative entropy between these two distributions is kept). Second, as the user continues searching with related topics, i.e., in the same task, subsequent cover queries are generated with respect to the relative changes in the genuine queries' search topics (e.g., become more specific or more general). This also helps minimize the difference between server's prior and posterior beliefs of a user's sensitive information. Third, once the search results are returned for each submitted query, only those related to the user's genuine query will be provided to the user after re-ranking them using the locally maintained true user profile. In the meanwhile, cover clicks are also generated in all the returned search results to prevent privacy disclosure from click patterns.

To materialize our IQP framework, three components need to be realized, namely: 1) query intent inference, 2) intent-aware cover query and click generation; 3) client-side personalization. We will use the search process illustrated in Figure 1 as a running example to discuss our design of these components.

• **Task1: Query Intent Inference.** We choose the topic hierarchy defined in the Open Directory Project (ODP) [44] as our intent tree for query intent inference, as it has been widely used for search intent modeling [8] and personalization [13, 17, 47]. Note that our solution is not restricted to the ODP topic structure, and it can be easily adopted to any other general or specific topic ontology, such as Wikipedia categories. We build a set of language models up to four-grams for each node on the intent tree, as [25] suggested that the typical length of a web query is between 3 and 4. Those language models are estimated based on the crawled content from the associated URLs in each ODP topic node; and hierarchical Dirichlet prior smoothing [31] is performed to smooth the language models. The search intent of a given user query can then be effectively predicted by the maximum a posterior inference,

$$P(t|q) \propto P(q|t)P(t) = P(W_m W_{m-1} \cdots W_1|t)P(t) \quad (1)$$

where $P(W_m W_{m-1} \cdots W_1|t)$ is provided by the n-gram language models for topic t. The prior distribution for topic t can be estimated with regard to the topic hierarchy by,

$$P(t) \propto \#[\text{nodes in the sub-tree rooted at } t] \quad (2)$$

This estimation can be improved by the statistics from an annotated search log, when it is available. To ensure its runtime efficiency, we build an inverted index over the raw content of the intent tree nodes, and only perform inference on the matched nodes with a given query. For the nodes with no content overlapping with the query, its language models can only score this query via the prior distribution introduced by smoothing. Therefore we consider it as less significant.

We have to note that topic inference in search queries is still an open research problem [8], and no perfect solution exists so far. The inaccuracy of topic inference undeniably affects our IQP solution; and the search engine might also use different methods than ours for this purpose. It is interesting to study how the difference between server-side and client-side inference accuracy leads to different level of privacy disclosure in IQP. In this paper, we assume the server side would use a similar topic inference algorithm; and leave this more general question as our future work.

Algorithm 1 Intent-aware Cover Query Generation

Input: User profile, p; Query, q; Topic tree, T; parameter, l
Output: Set of cover queries, Q

$q' \leftarrow GetLastSubmittedQuery(p)$
$t_q \leftarrow GetTopic(q)$, $t_{q'} \leftarrow GetTopic(q')$
$\mathcal{T} \leftarrow \{\}$, $Q \leftarrow \{\}$,
if q and q' belong to the same task **then**
  $T' \leftarrow GetTopic(GetCoverQueries(q'))$
  for each $t \in T'$ **do**
    $\mathcal{T} \leftarrow \mathcal{T} + [t' \in T : distance(t_q, t_{q'}) = distance(t, t')]$
else
  $\mathcal{T} \leftarrow GetCoverTopics(T, l)$
for each $t \in \mathcal{T}$ **do**
  $k \leftarrow PoissonDistribution(length(q))$
  $Q \leftarrow Q + RejectionSampling(nGramLM(t), k)$
return Q

• **Task2: Intent-aware Cover Query and Click Generation.** We adopt the entropy l-diversity principle [30] to protect Bayes-optimal privacy. Intuitively, entropy increases as frequencies become more uniform; if there are l equally "well-represented" sensitive topics in a user profile, a search engine needs l-1 damaging pieces of background knowledge to eliminate l-1 possibilities and infer a positive disclosure. Hence, by setting the parameter l, we can determine how much protection is provided against background knowledge – even if this background knowledge is unknown to us.

In IQP, after inferring the topic of a user's genuine query, those closely related topics will be selected to generate cover queries. Instead of selecting the topics independently (as performed in existing works [2, 33]), IQP maintains the relative transition of those selected cover topics with respect to those in the recognized genuine

search task. We consider two factors when selecting the cover topics: 1) specificity of user search intents, and 2) transition between current and previous search intents in the current search task. The first aspect ensures the selected cover intents are comparable to the genuine ones. The second ensures the sequentially generated cover queries are comparable to the true query sequence, because the transition among true queries in the same task discloses additional information about a user's search intent. Various solutions have been proposed to identify search tasks [1, 27, 49]; and we can use any off-the-shelf algorithm for this purpose (even the tasks are performed in an interleaved manner), as it is orthogonal to our query generation procedure.

We list the procedure of cover query generation in Algorithm 1. For the first query in a search task, IQP selects l cover topics with respect to the prior probability of the inferred genuine search intent (i.e., Step 9 in Algorithm 1), where we use the same prior distribution of search topics defined in Eq (2). A straightforward way is to use rejection sampling [21]. However, consider the intent is organized in a tree structure, sibling nodes would always have similar prior and therefore being preferred; but selecting them as cover intents discloses their shared common ancestors (as shown in Figure 1). To avoid this type of disclosure, we add another constraint in rejection sampling: only a fraction of cover topics can be selected from the sibling nodes, and the rest should be randomly sampled from the non-sibling nodes of similar prior probability. This ratio is dynamically adjusted by the depth of the intent node of the true user query: if it is more specific (deeper in the tree), fewer cover topics will be selected from its sibling nodes, as they would share more common ancestors; and vice versa.

Once IQP detects the user continues a previous search task, the cover topics will be selected to retain similar transitions on the intent tree as in the true query sequence (i.e., Step 5–7 in Algorithm 1). We choose to follow the intent transition pattern detected in the genuine query sequence, rather than to perform random walk on the intent tree, because the transition probability also discloses information about a user's search intent. It is unlikely for a user to randomly traverse in the topic space to fulfill an information need (although every step seems plausible); and therefore if the cover queries are generated in this way, a search engine can easily recognize and filter them. In particular, IQP assumes if the previous and current query intents belong to the same topic path on the intent tree, the queries are serving for closely related intents. For example, in Figure 1, if the user's next query becomes *"Polycystic Ovary Syndrome"*, it indicates the user switches to a more general intent (i.e., from "Top/Health/Disease/Treatment" to "Top/Health/Disease"). IQP will retrieve the topics used to cover the previous query and follow the same transition pattern to select new cover topics, i.e., move those topics upwards along the intent tree. For the selected topics, if they cannot follow the detected transition on the intent tree, for example leaf nodes cannot move downwards, we will keep them intact with probability β, otherwise we use rejection sampling method described above to select a new cover topic (i.e., to initiate a new cover task).

After the cover topic is selected, we use rejection sampling to generate cover queries from the n-gram language models associated with the topic, so that similar specificity between the genuine and cover queries can be achieved (i.e., Step 10–12 in Algorithm 1). Specifically, we use the difference between entropy of genuine and generated cover queries as the condition in rejection sampling. In addition, we use a Poisson distribution to randomize the length of the cover queries. Rate parameter λ of the Poisson distribution is set to the average length of the queries submitted by the user. This further reduces information available for the search engine to discover the genuine query. Besides, as users would only click on results for the genuine queries, we also generate cover clicks for the cover queries, otherwise a search engine can easily recognize the injected queries by their click-through rates. We use a positional click model [38] trained on a large reference search log, and sample clicks from it accordingly. Specifically we estimated click model per topic nodes on the intent tree to maintain different click distributions under different search intents.

One practical concern of this solution is its *induced burden* on network traffic. Treatments exist to alleviate this burden: for example, submitting the cover query and click in an asynchronized fashion regarding the genuine query and click (e.g., randomly postpone the submission of cover traffic), so that peak time traffic is reduced (but total traffic is the same). However, the total amount of additional information cannot be reduced, as it is necessary to achieve Bayes Optimal privacy in this scenario.

• **Task3: Client-side Personalization.** As cover queries and clicks are injected from client-side for privacy protection, search quality from a search engine that builds user profile on such noisy input for personalization will be undermined. To retain the utility of personalization, we perform client-side reranking with an uncontaminated user profile. We use rank aggregation by Borda's method [51] to merge the ranked search results from search engine and client-side user profile. Specifically, Borda's method assigns a ranking score corresponding to the position in which a candidate document appears within each ranker's ranked list, and the candidates are sorted by this integrated ranking score:

$$score(d) = \alpha/R_1(d) + (1-\alpha)/R_2(d) \tag{3}$$

where $R_i(d)$ denotes the ranking order of document d in ranker i's ranked list, and α controls the weight of each ranker. We use a language model estimated on the uncontaminated user profile to compute personalization score as follows [42, 46],

$$UPScore(d) = \sum_{w \in q \cap d} \log \frac{(1-\lambda)p_{ml}(w|d) + \lambda p(w|C)}{\lambda p(w|C)} \cdot tf(w)$$

then aggregate the rankings using Eq (3) to compute the final ranking of the retrieved documents. [2] used a similar client-side reranking strategy, but their client-side ranking is simply based on document profile matching, without considering the background popularity of a matched query term outside a user profile. However, we should note that the main focus on this work is privacy protection rather than yet another personalization method; more advanced techniques, such as learning-based methods, can be applied to further boost the retrieval quality and we leave this as one of our future works.

4 METRICS OF PRIVACY PROTECTION

Evaluating the effectiveness of privacy protection is vital and as important as developing the protection solution. But as we discussed earlier, there is insufficient attention in this direction. Some prior work uses KL-divergence to measure the statistical difference between the user profiles constructed on the server-side and client-side, and Normalized Mutual Information [2] to measure the

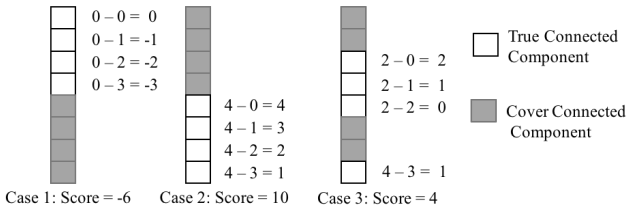


Figure 2: Illustration of the Confusion Index calculation

statistical relatedness between a set of genuine user queries and corresponding cover queries. These two metrics reflect the difference between a search engine's prior and posterior beliefs of a user's search behavior. But such metrics treat queries as independent, so that it cannot measure information disclosure at the task-level, where queries are dependent on each other.

We design two new metrics to evaluate the *task-level* privacy disclosure from the perspective of task distribution in the space of search intents and intent transitions. We name them as *Confusion Index* and *Transition Index* accordingly.

• **Confusion Index (cIndex)** We measure a search engine's belief about a user's information need conveyed in a search task by the difference between the prior and posterior distributions on the intent tree. According to [27], a task can be represented as connected components on the intent tree. Hence, the difference only needs to be measured on those components (as other parts have no change resulted from this search task). We quantify the belief updates by the change of entropy on those associated components in the prior and posterior distributions, i.e., the entropy l-diversity principle. A larger entropy reduction on a component indicates a user's intent is more likely to concentrate on it, and therefore the search engine will be more confident to assert it is the user's true intent. Therefore, in cIndex we rank the connected components affected by the genuine and cover tasks in a descending order regarding entropy change, and count how many connected components created by the cover tasks are ranked ahead of those created by genuine tasks. The larger cIndex is, the less accurately a search engine will be able to recognize the true user intent in a search task, and therefore the less sensitive information is disclosed. Intuitively, cIndex directly quantifies privacy disclosure at the level of related search intents. The prior probability of a connected component can be estimated in the same way as defined in Eq (2), and the posterior can be estimated based on the counts collected from the inferred intents in the associated task (Dirichlet prior smoothing is used in our experiments). Formally, we define cIndex on an intent tree as,

$$cIndex(m) = \sum_{i=1}^{|m|} \Big[\sum_{j=1}^{i-1} \big(1 - \delta(j)\big) - \sum_{j=1}^{i-1} \delta(j) \Big] \qquad (4)$$

where m is a list of connected components resulted from all historical queries ranked in a descending order of entropy change; and $\delta(j) = 1$ if the jth component in m is related to a true user task, and 0 otherwise. To keep cIndex value invariant with respect to the input sequence length, we normalize it by $(cIndex - min)/(max - min)$ where min represents cIndex score when all true components are ranked ahead of cover components (case 1 in Figure 2) and max represents score when all cover components are ranked ahead of true components (case 2 in Figure 2) in a given set of connected components.

The logic behind the design of cIndex is the Bayes-optimal Privacy principle. The entropy changes on the connected components on the intent tree directly relate to the update of a search engine's posterior belief from an observed query sequence. If connected components with large entropy change are resulted from cover queries, the search engine's posterior belief on the true user intent can hardly be increased. Hence, privacy is obtained. We visualize the definition of cIndex in Figure 2. Intuitively, cIndex counts how many connected components resulted from a cover task can be ranked above those resulted from a true task at every position in this ranked list. It is easy to verify that cIndex requires a balanced distribution between the cover components and true components (i.e., comparable change in entropy). On the one hand, if an algorithm concentrates all cover queries on one component, although its entropy change will be maximized, it can only cover one true component and therefore leads to a worse cIndex. On the other hand, if an algorithm puts every cover query into a disjoint component to maximize the number of cover components, their entropy change will be minimal and make the true intents ranked all above the cover ones.

• **Transition Index (tIndex)** In cIndex, we measure task plausibility regarding its queries' concentration on the intent tree. This ignores the internal relation between queries in a task – their transitional patterns. We follow a similar principle as in cIndex to define tIndex, which counts how many cover tasks will be ranked ahead of true tasks with respect to the intent transition probability.

However, if we define transition as the change of search intent in two consecutive queries, the transitional space would be quadratic to the number of nodes on an intent tree. This is unfortunately too large to obtain a confident estimation for any practical topic ontology (e.g., ODP tree has more than 7,600 nodes in its first four levels). To reduce the problem space, we define transition of intents against the intent tree structure: if the follow-up query moves the inferred intent to the parent node of the previous query's intent, we denote the change as "UP1;" if it is moved to the grandparent node, it is denoted as "UP2," and etc. We consider up to two levels along the intent tree, and also include "SA," "MB" and "Others" to indicate staying at the same node, moving to a sibling node and the other nodes. As a result, our transition matrix is defined over those relative changes on the intent tree and can be effectively estimated from reference search logs with annotated search topics. Despite such definition of transition looses precision in recognizing fine-grain transition patterns in tasks, it makes the evaluation possible.

With an estimated transition matrix, every (both genuine and cover) task can be scored by its transition likelihood, and ranked in a descending order. We also normalized the transition probability with respect to length to avoid potential bias introduced by such a factor. The same method as defined in Eq (4) and normalization process is used to compute the corresponding tIndex.

5 EXPERIMENTS

In this section, we compare our approaches with five state-of-the-art query obfuscation techniques [2, 33, 35, 40, 53] in terms of privacy protection, query plausibility, and search effectiveness.

5.1 Dataset & Setup

We used the AOL search log, which contains 16,946,938 queries submitted by 657,426 unique users from March 1 to May 31, 2006.

There are 1,632,797 unique clicked documents. We have to admit that the AOL data set is a bit out of date, but it is the largest English search log data publicly available. We built our customized search engine using *Apache Lucene*, where we chose *Okapi BM25* as the ranking algorithm. For simplicity, our search engine always returns the top 100 documents. To simulate a personalized search engine, we compute relevance score of documents using a language model estimated by the server-side constructed user profiles [42]. In particular, the server-side user profiles are built on all submitted user queries and the corresponding clicked document content (including cover queries and cover clicks). All the documents are re-ranked using the rank aggregation method described in Eq (3) before returning to the users. We found $\alpha = 0.5$ for server-side rank aggregation and $\alpha = 0.3$ for client-side re-ranking provided the best ranking results. And we fixed them in all our experiments.

In our evaluation, we used the top 1000 most active users based on the volume of their query history; and this resulted in total 318,023 testing queries. We considered the clicked documents of each query as relevant in the ranking-based evaluation. As our solution is orthogonal to the choice of search task extraction method, we choose to identify in-session tasks in our experiments for simplicity; but any other task identification method can be seamlessly inserted into our framework. We used 30-minutes inactive time threshold to segment sessions [26], and we further segmented a session into tasks if the cosine similarity of two consecutive queries is less than 0.5 [27]. Since the total number of nodes in the ODP tree is large, we only considered categories up to level four. This gives us 7,600 topic nodes. To build the language models, we crawled 82,020 web documents pointed by the URLs in those topic nodes.

We employed mean average precision over the top 100 returned documents, i.e., MAP@100, to evaluate the ranking quality. In addition to the newly proposed cIndex and tIndex metrics, we also included the previously used *Normalized Mutual Information (NMI)* between true and cover queries and *Kullback–Leibler divergence (KL)* between true and noisy user profiles to measure the effectiveness of privacy protection [2]. In particular, to compute tIndex, we randomly sampled 10 thousands users from the AOL search log except those in our testing set, and performed the topic inference defined in Eq (1) to estimate the transition probability matrix. Basically, cIndex and tIndex measure privacy disclosure at a task level, and NMI and KL divergence measure it at a query level.

We also employed another two previously used metrics based on external web resources to evaluate the statistical property of true and cover queries, i.e., statistical plausibility. This reflects the additional knowledge a search engine might have about individual users. First, we compute *Information Content ratio* [40] by,

$$\text{IC ratio}(a, b) = \frac{Max\big(hit\_count(a), hit\_count(b)\big)}{Min\big(hit\_count(a), hit\_count(b)\big)}, \qquad (5)$$

which measures the ratio of search result hits for query a and b in a large search engine index. A good cover query should have its IC ratio close to one, so that it cannot be differentiated from the true query in this dimension. We used Microsoft Bing API to find the hit count of a query. Second, we use Microsoft Web Language Model API to compute the probability difference between a query and its cover query at the web scale, and name this metric as query plausibility difference, or ΔP in short. A good cover query should have ΔP close to zero, i.e., it has very similar probability as the true query regarding to a search engine's background knowledge. Note, both metrics are query-level measures about privacy disclosure.

We compare with the following five baseline approaches:

• **Plausible Deniable Search (PDS)** We implemented the PDS [33] model based on the ODP dataset. Following its design, we created 161,553 seed queries and 161,318 canonical queries. In the PDS model, a user's genuine query is not submitted to the search engine; instead, the cover queries are used to retrieve results for the user. This results in very poor retrieval performance in a keyword-based search engine. We modified this model by submitting the original user query along with the generated cover queries to make this model comparable with others in terms of search effectiveness.

• **Knowledge-based Scheme (KBS)** This model focuses on nouns and noun phrases in a user query and generates cover queries based on a predefined lexical ontology [40]. We implemented this method based on WordNet and ODP categories. All the 117,798 nouns from WordNet and 763,378 categories from ODP tree are used. For a given genuine query, the name of nodes within a predefined distance (2 in our experiment) to the inferred query intent nodes on the ontology are selected as cover queries. Like PDS, we modified KBS model to submit true queries to the search engine to make the comparison coherent with others.

• **Embellishing Search Queries (ESQ)** ESQ model [35] embellishes a user query by adding decoy terms with the original query terms. The decoy terms are selected from pre-processed nouns of WordNet, which are organized in buckets of different specificity. We used 117,798 nouns and 82,115 synsets of WordNet to implement this algorithm. ESQ has two parameters, bucket size and segment size, which are set to 4 and 512 respectively. In ESQ, user queries are embellished by first selecting the bucket which contains a genuine search term and then injecting all the other terms from the same bucket as decoy terms to the user query.

• **Topic-based Privacy Protection (TPP)** It is one of the most recent works in privacy preserving personalized search [2], which generates cover queries from a pre-trained statistical topic model. We used the same set of crawled web documents of ODP categories to build a LDA topic model with 100 topics. For each genuine query, TPP first infers its topic and draw query terms independently from the selected topic. TPP also performs client-side reranking to improve search utility.

• **Anonymizing User Profiles (AUP)** AUP [53] is a server-side algorithm for privacy protection. It clusters user profiles based on cosine similarity to form group profiles. To better compute the similarity between user profiles, AUP augments the query terms in a user profile by using synonym and hypernym set from WordNet. AUP balances privacy and search effectiveness by controlling the size of group profiles and introducing diversity during clustering. The AUP model has two parameters: p-linkability and number of steps for hypernym set augmentation. We set $p = 0.2$ and only performed one step augmentation, which results in 116 group profiles over 1000 testing users.

5.2 Experiment Results

• **Comparison on Search Effectiveness.** The average MAP@100 with and without client-side personalization from different privacy protection algorithms is reported in Table 1. We compared IQP with TPP, PDS and KBS models under two different sizes of generated

Table 1: Comparison across different privacy protection solutions. $*$, $\star$ indicate MAP@100 computed by submitting only cover queries to the search engine and by submitting original queries along with cover queries. Statistical significance test is conducted by comparing the best two algorithms and the result is presented in bold-faced (p-value < 0.05).

Settings	Model	MAP@100	MAP@100 [client-side personalization]	KL Divergence	NMI	cIndex	tIndex
$l = 0$	Okapi BM25	0.1236	NA	NA	NA	NA	NA
	BM25 + Personalization	0.1638	NA	NA	NA	NA	NA
	AUP	0.1088	0.1171	0.9636	NA	NA	NA
	ESQ	0.1161	0.1090	0.0912	NA	NA	NA
$l = 2$	IQP	**0.1387**	**0.1486**	0.6866	**0.2156**	**0.5127**	**0.6016**
	TPP	0.1158	0.1174	**0.7558**	0.3922	0.2635	0.5779
	PDS	0.0000^{*} $0.1307^{\star}$	0.1391	0.4467	0.4308	0.3313	0.1936
	KBS	0.0143^{*} $0.1255^{\star}$	0.1474	0.7001	0.2914	0.3775	0.4827
$l = 4$	IQP	0.1331	0.1396	0.8306	**0.2193**	**0.5034**	**0.6116**
	TPP	0.1076	0.1094	**0.9545**	0.3918	0.2565	0.5872
	PDS	0.0000^{*} $0.1179^{\star}$	0.1315	0.5474	0.4337	0.3163	0.2004
	KBS	0.0282^{*} $\mathbf{0.1348}^{\star}$	**0.1411**	0.8814	0.2912	0.3887	0.4149

cover queries, i.e., $l = 2$ and 4. Because in AUP and ESQ, no cover query is generated, their ranking performance is reported in $l = 0$ row of Table 1. We also reported the average MAP@100 of our search engine (denoted as *Okapi BM25*) with its personalization function enabled (denoted as *BM25 + Personalization*) in Table 1. From the results, we can observe that because of the injected cover queries from the privacy protection algorithms, the search engine's ranking performance dropped significantly. And basically the more cover queries were injected, the worse ranking performance one would have. This is expected as the search engine is using inaccurate historical information to personalize the results. We can note that KBS provided improved search utility with more cover queries (with its distance parameter set to 2). We experimented with different distance parameters for KBS, and found MAP decreased with a larger distance parameter, as we can expect. Comparing IQP with other privacy protection algorithms, the decrease of ranking utility is minimized. We attribute it to the client-side re-ranking and the controlled noise that IQP introduces through the cover queries. Note that, we also reported the search performance of PDS and KBS without submitting the true user queries, as in their original designs the genuine queries are not submitted. Clearly, this greatly hurts the search utility of these two algorithms, as more than 90% substituted cover queries contain generic terms from those two algorithms, which provides little utility to users' search intent. And arguably it is meaningless to provide such type of privacy protection, as the users would not gain any utility from the search service.

• **Comparison on Privacy Protection.** We evaluated IQP on privacy disclosure at both query level and task level, and the results are presented in Table 1. IQP adds less amount of accumulated noise comparing to other approaches; therefore, the KL divergence is smaller. This result also partially explains why IQP's ranking performance is better than the other models'. Due to the nature of PDS, it cannot generate cover queries for most of the user query (more than 90% of our testing queries), as the query terms have to exist in a predefined dictionary, and thus it provides very weak privacy protection to the users (true queries have to be submitted without cover queries). Significance test in the NMI results clearly indicates that among every submitted cover query IQP discloses less private information than the other methods. It is evident that adding noise in a controlled manner so that privacy can be protected as well as personalization utility can be retained for users is one of the key advantages of the proposed IQP solution. We also noticed that the results of KL Divergence on keyword-based user profile was not consistent with other evaluation metrics (MAP and NMI), this is because we computed KL divergence between noisy profile and true profile and the KL divergence computation is not symmetric. Moreover, neither NMI nor KL divergence considers the semantic similarity between added terms to a user profile. This result justifies that developing task-level metric is necessary.

The cIndex and tIndex defined in Section 4 are designed to evaluate privacy disclosure at the task level. It can be evaluated after every user task is finished, i.e., to measure how much private information is disclosed in this task. But as we are working with in-session tasks, which are generally short (3-4 queries in average), the changes in entropy and transition probabilities might not be significant. To make the comparison in a more perceptible scale, we computed cIndex and tIndex in each week and reported the results in Table 1. As shown in the results, IQP significantly outperformed all baselines in cIndex and tIndex. As queries in true user tasks are related in their search intents and therefore concentrated on some connected components, a privacy protection algorithm has to generate similar concentrations on the other parts of the intent tree, i.e., plausible cover tasks, to improve cIndex. On the other hand, IQP mimics similar transition patterns as in the true user tasks when generating cover tasks, which leads to a better tIndex. These results prove the effectiveness of IQP in protecting privacy at the task level.

To make a more comprehensive comparison of different algorithms' task-level privacy protection effectiveness, we reported cIndex and tIndex at different time intervals (with $l = 2$) in Figure 3. We can notice that IQP achieved consistent performance in different time intervals; this indicates its generated cover tasks consistently match with the true user tasks under these two metrics. We observed that the TPP baseline achieved very good tIndex in most of comparisons. We looked into its created cover queries and found they tended to distribute uniformly on the intent tree, because of the design of TPP. This leads to coherent transition sequences and a higher tIndex in TPP. But if we compare it with IQP on other metrics, such as cIndex, its performance is clearly unsatisfactory.

Another observation we found is that more private information is disclosed as a user interacts with a search engine for a longer time. One potential reason is that in our current solution, we only

Figure 3: cIndex and tIndex comparisons in different algorithms with different time intervals ($l = 2$).

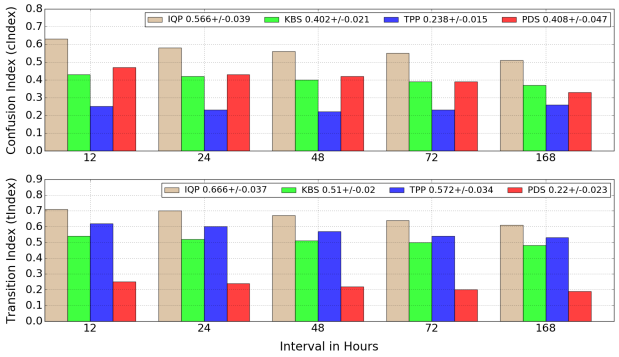
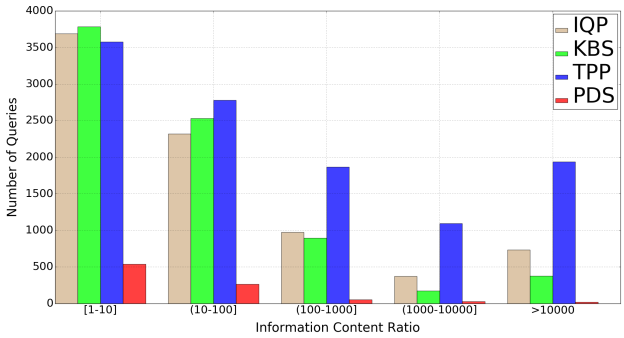


Figure 4: Comparison on *IC ratio* between privacy protection models ($l = 2$).

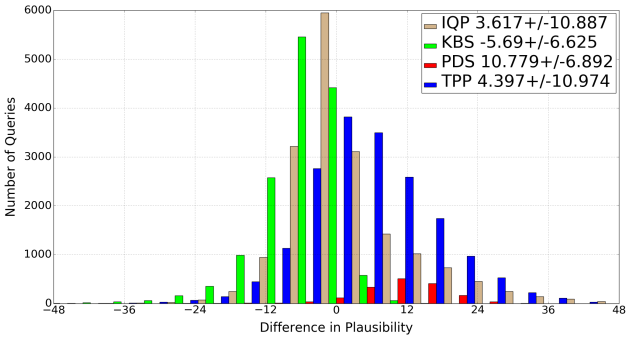


focused on in-session tasks. But it is known that users might return to previous unfinished tasks across sessions [1, 49]. As a result, the generated cover tasks become inconsistent for those cross-session tasks. But our IQP solution can be easily extended to protect cross-session tasks, by incorporating cross-session task extraction algorithms.

• **Comparison on Statistical Query Plausibility.** We compared IQP with TPP, PDS, and KBS in terms of query plausibility using *IC ratio* and *query plausibility difference* (ΔP) on 50 randomly selected testing users. A privacy model that has more cover queries in a low *IC ratio* range and less cover queries in a higher *IC ratio* range is preferred. As depicted in Figure 4, the number of queries belongs to a low *IC ratio* range is larger in IQP compared to those in TPP and PDS but smaller compared to KBS. It indicates cover queries generated by IQP are getting more search results (i.e., more plausible) compared to the cover queries generated by TPP and PDS, but less than KBS. Since KBS uses category names of the ODP ontology and hypernyms/hyponyms from WordNet, cover queries generated by KBS is more generic so that it brings in more search results.

Figure 5 illustrates the comparison on ΔP for IQP, KBS, PDS and TPP models. Positive ΔP indicates cover queries are less plausible compared to the original user queries and vice versus. Therefore, a ΔP value closes to zero represents similar query plausibility; thus, it is more difficult to differentiate true queries from the corresponding cover queries. From the results, we can find KBS constantly generates more plausible cover queries than true queries, because it uses only noun phrases as its cover query. In a long run, this will cause negative disclosure of privacy, as the search engine can filter

Figure 5: Comparison on query plausibility difference between genuine and cover queries ($l = 2$).



the queries that are overly general. Compared to the TPP model, IQP creates more plausible cover queries which make it harder for a search engine to decipher the true queries from the cover queries. On the other hand, the PDS model cannot generate cover queries for most of the genuine user query and its deficiency is clearly evident in Figure 4 and 5.

To better understand how plausible cover queries generated by IQP is, we present an example of in-session user queries from AOL search log, their inferred intent by IQP and one cover query per true query generated by IQP along with their topic in Table 2. The user's geniue task is health related, while the generated cover task is highly related to real estate business.

Table 2: An example of cover queries generated by IQP along with their topic.

Session queries	pregnancy symptoms in the first month abortion pills florida abortion clinics
Query topics	Home/Family/Pregnancy Society/Issues/Abortion Society/Issues/Regional
Cover queries	Low priced compact flash GPS receivers global hotel investment trends commercial real estate development
Cover query topics	Shopping/Consumer_Electronics/Accessories Business/Real_Estate/Property_Management Business/Real_Estate/Development

6 CONCLUSIONS AND FUTURE WORK

We developed an effective intent-aware query obfuscation solution to maintain *Bayes-Optimal Privacy* in a personalized web search environment. Our model handles users' sequentially developed intents in search tasks. Two new metrics measuring task-level privacy disclosure are developed to assess privacy protection quality. Promising results on AOL search log confirmed the utility of our solution in protecting the linkability aspect of privacy in a personalized retrieval system. A chrome plugin [52] is developed based on the query obfuscation technique proposed in this paper.

This research opens a wide spectrum of future research topics. Our current model generates fixed amount of cover queries for each user query, as we assume all user queries are sensitive. Relaxing this assumption and adaptively adjusting the amount of cover queries generated for different queries would better balance the utility of

personalization and privacy protection. Furthermore, it is necessary to perform user studies to understand real users' perception and satisfaction about this type of privacy protection solutions.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments. This work was supported in part by National Science Foundation Grant IIS-1760523 and IIS-1553568.

REFERENCES

[1] Eugene Agichtein, Ryen W White, Susan T Dumais, and Paul N Bennet. 2012. Search, interrupted: understanding and predicting search task continuation. In *Proceedings of the 35th ACM SIGIR*. ACM, 315–324.

[2] Wasi Uddin Ahmad, Md Masudur Rahman, and Hongning Wang. 2016. Topic Model based Privacy Protection in Personalized Web Search. In *Proceedings of the 39th ACM SIGIR*. ACM, 1025–1028.

[3] Avi Arampatzis, George Drosatos, and Pavlos S Efraimidis. 2013. A versatile tool for privacy-enhanced web search. In *European Conference on Information Retrieval*. Springer, 368–379.

[4] Avi Arampatzis, George Drosatos, and Pavlos S Efraimidis. 2015. Versatile Query Scrambling for Private Web Search. *Information Retrieval Journal* 18, 4 (2015), 331–358.

[5] Naveen Farag Awad and MS Krishnan. 2006. The personalization privacy paradox: an empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS quarterly* (2006), 13–28.

[6] Ero Balsa, Carmela Troncoso, and Claudia Diaz. 2012. OB-PWS: Obfuscation-based private web search. In *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 491–505.

[7] Michael Barbaro, Tom Zeller, and Saul Hansell. 2006. A face is exposed for AOL searcher no. 4417749. *New York Times* 9, 2008 (2006), 8For.

[8] Paul N Bennett and Nam Nguyen. 2009. Refined experts: improving classification in large taxonomies. In *Proceedings of the 32nd ACM SIGIR*. ACM, 11–18.

[9] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. 2004. Public key encryption with keyword search. In *Advances in Cryptology-Eurocrypt 2004*. Springer, 506–522.

[10] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou. 2014. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *Parallel and Distributed Systems, IEEE Transactions on* 25, 1 (2014), 222–233.

[11] Jordi Castellà-Roca, Alexandre Viejo, and Jordi Herrera-Joancomartí. 2009. Preserving user's privacy in web search engines. *Computer Communications* 32, 13 (2009), 1541–1551.

[12] Ramnath K Chellappa and Raymond G Sin. 2005. Personalization versus privacy: An empirical examination of the online consumer's dilemma. *Information Technology and Management* 6, 2-3 (2005), 181–202.

[13] Ye Chen, Dmitry Pavlov, and John F Canny. 2009. Large-scale behavioral targeting. In *Proceedings of the 15th ACM SIGKDD*. ACM, 209–218.

[14] Dan Cvrcek, Marek Kumpost, Vashek Matyas, and George Danezis. 2006. A study on the value of location privacy. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*. ACM, 109–118.

[15] Roger Dingledine. 2011. Tor and circumvention: Lessons learned. In *Advances in Cryptology–CRYPTO 2011*. Springer, 485–486.

[16] Josep Domingo-Ferrer, Maria Bras-Amorós, Qianhong Wu, and Jesús Manjón. 2009. User-private information retrieval based on a peer-to-peer community. *Data & Knowledge Engineering* 68, 11 (2009), 1237–1252.

[17] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th WWW*. ACM, 581–590.

[18] Magdalini Eirinaki and Michalis Vazirgiannis. 2003. Web mining for web personalization. *ACM Transactions on Internet Technology (TOIT)* 3, 1 (2003), 1–27.

[19] Liyue Fan, Luca Bonomi, Li Xiong, and Vaidy Sunderam. 2014. Monitoring web browsing behavior with differential privacy. In *Proceedings of the 23rd WWW*. ACM, 177–188.

[20] Michael J Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. 2005. Keyword search and oblivious pseudorandom functions. In *Theory of Cryptography*. Springer, 303–324.

[21] Walter R Gilks and Pascal Wild. 1992. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* (1992), 337–348.

[22] Avi Goldfarb and Catherine E Tucker. 2011. Privacy regulation and online advertising. *Management Science* 57, 1 (2011), 57–71.

[23] Daniel C Howe and Helen Nissenbaum. 2009. TrackMeNot: Resisting surveillance in web search. *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society* 23 (2009), 417–436.

[24] Ali Inan, Murat Kantarcioglu, Gabriel Ghinita, and Elisa Bertino. 2010. Private record matching using differential privacy. In *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, 123–134.

[25] Bernard J Jansen and Amanda Spink. 2005. An analysis of web searching by European AlltheWeb. com users. *Information Processing & Management* 41, 2 (2005), 361–381.

[26] J Jansen Bernard, Amanda Spink, Chris Blakely, and Sherry Koshman. 2007. Defining a session on Web search engines: Research Articles. *Journal of the American Society for Information Science and Technology* 58, 6 (2007), 862–871.

[27] Rosie Jones and Kristina Lisa Klinkner. 2008. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM CIKM*. ACM, 699–708.

[28] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. 2007. "I know what you did last summer" – query logs and user privacy. In *Proceedings of the 16th ACM CIKM*. ACM, 909–914.

[29] Ting-Peng Liang, Hung-Jen Lai, and Yi-Cheng Ku. 2006. Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings. *Journal of Management Information Systems* 23, 3 (2006), 45–70.

[30] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. l-diversity: Privacy beyond k-anonymity. *ACM TKDD* 1, 1 (2007), 3.

[31] David JC MacKay and Linda C Bauman Peto. 1995. A hierarchical Dirichlet language model. *Natural language engineering* 1, 03 (1995), 289–308.

[32] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. 2000. Automatic personalization based on Web usage mining. *Commun. ACM* 43, 8 (2000), 142–151.

[33] Mummoorthy Murugesan and Chris Clifton. 2009. Providing Privacy through Plausibly Deniable Search. In *SDM*. SIAM, 768–779.

[34] Andrew Odlyzko. 2003. Privacy, economics, and price discrimination on the Internet. In *Proceedings of the 5th international conference on Electronic commerce*. ACM, 355–366.

[35] HweeHwa Pang, Xuhua Ding, and Xiaokui Xiao. 2010. Embellishing text search queries to protect user privacy. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 598–607.

[36] Kristin Purcell, Joanna Brenner, and Lee Rainie. 2012. Search engine use 2012. (2012).

[37] Michael G Reed, Paul F Syverson, and David M Goldschlag. 1998. Anonymous connections and onion routing. *Selected Areas in Communications, IEEE Journal on* 16, 4 (1998), 482–494.

[38] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th WWW*. ACM, 521–530.

[39] Felipe Saint-Jean, Aaron Johnson, Dan Boneh, and Joan Feigenbaum. 2007. Private web search. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*. ACM, 84–90.

[40] David Sánchez, Jordi Castellà-Roca, and Alexandre Viejo. 2013. Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines. *Information Sciences* 218 (2013), 17–30.

[41] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th ACM SIGIR*. ACM, 43–50.

[42] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM CIKM*. ACM, 824–831.

[43] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2007. Privacy protection in personalized search. In *ACM SIGIR Forum*, Vol. 41. ACM, 4–17.

[44] Chris Sherman. 2000. Humans Do It Better: Inside the Open Directory Project. *Online* 24, 4 (2000).

[45] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. 2004. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th WWW*. ACM, 675–684.

[46] Bin Tan, Xuehua Shen, and ChengXiang Zhai. 2006. Mining long-term search history to improve search accuracy. In *Proceedings of the 12th ACM SIGKDD*. ACM, 718–723.

[47] Jaime Teevan, Susan T Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th ACM SIGIR*. ACM, 449–456.

[48] Eran Toch, Yang Wang, and Lorrie Faith Cranor. 2012. Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Modeling and User-Adapted Interaction* 22, 1-2 (2012), 203–220.

[49] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ryen W White, and Wei Chu. 2013. Learning to extract cross-session search tasks. In *Proceedings of the 22nd WWW*. ACM, 1353–1364.

[50] Jun Yan, Ning Liu, Gang Wang, Wen Zhang, Yun Jiang, and Zheng Chen. 2009. How much can behavioral targeting help online advertising?. In *Proceedings of the 18th WWW*. ACM, 261–270.

[51] H Peyton Young. 1974. An axiomatization of Borda's rule. *Journal of economic theory* 9, 1 (1974), 43–52.

[52] Puxuan Yu, Wasi Uddin Ahmad, and Hongning Wang. 2018. Hide-n-Seek: An Intent-aware Privacy Protection Plugin for Personalized Web Search. In *Proceedings of the 41st ACM SIGIR*. ACM.

[53] Yun Zhu, Li Xiong, and Christopher Verdery. 2010. Anonymizing user profiles for personalized web search. In *Proceedings of the 19th WWW*. ACM, 1225–1226.