Applying Epsilon-Differential Private Query Log Releasing Scheme to Document Retrieval

Sicong Zhang, Hui Yang, Lisa Singh
Department of Computer Science, Georgetown University
sz303@georgetown.edu, {huiyang,singh}@cs.georgetown.edu

ABSTRACT

Web search logs are valuable and widely used for improving Information Retrieval (IR) research. However, these query logs contain sensitive data, which makes them difficult to be released directly even for research purposes. More companies can release their query logs if adequate privacy protection can be placed. This workshop paper introduces our research project on this privacy preserving query log releasing problem. In this paper, we propose a framework using differential privacy on query logs to guarantee high levels of privacy which achieves ϵ-differential privacy. We also provide a proof for why the user IDs for each of the individual search records can not be released in order to achieve differential privacy. Experiments show that our approach is a strong option for maintaining both high privacy and high utility. We believe this project is an important step forward to find practical solutions to this query log releasing problem.

Keywords

Privacy-Preserving IR, Query Log Anonymization, Differential Privacy, Document Retrieval.

1. MOTIVATION

Web search query logs are important and valuable for Information Retrieval (IR) research. Many recent IR methodologies are developed or inspired from the analysis of user behavior in search query logs [2, 8, 15, 18, 21]. However, these search query logs also contain sensitive private information that should not be exposed to the public directly. For instance, AOL released a piece of its search query log data in 2006 in order to help the research community [1]. Although the user ids in that AOL dataset were anonymized by hashed tags, some of the web users from the dataset still got identified [3]. Such data release without enough privacy protection resulted in severe social and legal issues. In order to protect users' privacy and avoid potential legal issues, very few search query logs have been released from major commercial search engines since then.

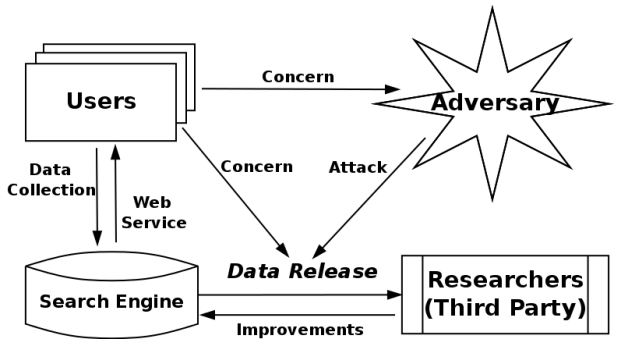

Figure 1: Query log releasing framework for search engine.

Figure 1 shows the relationships among search engines, third party researchers, web users, and potential adversaries. As we can see, the data release process is the key concern and the major problem becomes can search query logs be released without violating user privacy? This motivates us to develop an query log anonymization scheme that will guarantee sufficient level of privacy, while maintaining enough data utility for different IR tasks. The task we focus on in this project is to use a released query log to help document retrieval. Such document retrieval works are based on search logs or click-through data to improve the search results [4, 19, 20].

2. QUERY LOG ANONYMIZATION

The query log anonymization task has been attracting our community's concern since 2006. For years, researchers have proposed many ad-hoc techniques to help preserve privacy in query log anonymization. Cooper [6] made a survey of query log privacy-enhancing techniques from a policy perspective and summarized the major techniques as: *Log Deletion*, *Hashing Queries*, *Identifier Deletion*, *Hashing Identifiers*, *Scrubbing Query Content*, *Deleting Infrequent Queries* and *Shortening Sessions*. However, such techniques based on information removal have been shown to be not private [16].

With better guarantee of the privacy, k-anonymity [5, 13] and differential privacy [9, 10, 11, 12, 17] have been utilized to help in query log anonymizations. However, none of the

existing approach have achieved pure differential privacy.

2.1 Why User ID can not be Released

First of all, we would like to show that, a user-level differential privacy can not be achieved if we want to release the query log in the original format. To be more specific, the user ID can not be released as in the original query log. Here we give a mathematical proof that, if the released query log tuples are marked by its user IDs, a user-level differential private query log releasing scheme will lead to poor IR utility performances, which makes the output containing only random content. In other words, we point out that, a query log tuple's identity can not be released by its user ID. Here is a proof by contradiction.

STATEMENT. A user-level ϵ-differential private query log releasing scheme will lead to poor IR utility, if the released query log tuples are marked with their user IDs.

PROOF. We take the negation of the statement and suppose it to be true. Assume, to the contrary, that $\exists$ a user-level ϵ-differential private query log releasing scheme $A(Q)$, which releases user IDs for each of the query log tuples, and still lead to acceptable IR utility.

We consider this scheme $A(Q)$ as a stochastic algorithm, while otherwise a deterministic algorithm can also be considered as a special form of stochastic algorithm. Then, $A(Q)$ will be a stochastic algorithm satisfies user-level ϵ-differential privacy. Specifically, people can identify whether two query log tuples belong to the same user based on the user IDs.

According to the definition of differential privacy, for all *neighboring* query logs Q_1 and Q_2, and for all possible output Q^*, the following inequality holds:

$$e^{-\epsilon} P[A(Q_2) = Q^*] \leq P[A(Q_1) = Q^*] \leq e^{\epsilon} P[A(Q_2) = Q^*] \quad (1)$$

while *neighboring* means those two query logs differ by at most one specific user's query log.

Without lost of generality, let T_0 be a tuple with a user ID included in Q_2 but not included in Q_1. According to the differential privacy definition, theoretically T_0 can be any individual query log tuple from Q_2. Given the anonymization algorithm $A(Q)$, let's consider it in two cases:

Case 1. $\forall\, Q^*$ satisfies $T_0 \subseteq Q^*$, $P[A(Q_2) = Q^*] = 0$. In this case, the algorithm $A(Q)$ totally removes each individual T_0 from the query log, and makes it impossible to appear in the anonymized output. In other words, this $A(Q)$ removes everything from the original input, and leads to an useless empty output. This contradicts with our assumption that $A(Q)$ leads to acceptable IR utilities.

Case 2. Otherwise, there exists at least one Q^*, satisfies $T_0 \subseteq Q^*$, $P[A(Q_2) = Q^*] \neq 0$. Then, according to equation 1, we have $P[A(Q_1) = Q^*] \neq 0$. However, $T_0 \not\subseteq Q_1$ and actually T_0 can be anything not included in Q_1. No matter what positive value of $P[A(Q_1) = Q^*]$ is, the anonymization algorithm $A(Q)$ should add ALL possible tuples (T_0) into the output of the algorithm each with some positive probability. Theoretically there exists infinite different choices of such T_0. Therefore, this algorithm $A(Q)$ will have to output a query log fulfilled with ALL random tuples. In this case, the output of $A(Q)$ will be fully diluted into a random query log, which also leads to unacceptable IR utility. This contradict with out assumption that $A(Q)$ leads to acceptable IR utilizes.

In summary, the algorithm $A(Q)$ has to fall into one of the two cases we just discussed. In case 1, the algorithm will output nothing from the original query log, which makes the dataset unable to use. In case 2, the algorithm will output a query log fulfilled with all kinds of random tuples with different user IDs, and it fully diluted the tuples from the original query log. In this situation, although the original query log are kept in the output, the severe dilution makes the output query log into a total random query log. Both cases leads to hardly any IR utility, and contradict with our assumption. This contradiction shows that the supposition is false and so the given statement is true. This completes the proof. □

2.2 Releasing Algorithm

In our project, we generate an ϵ-differential private query log releasing algorithm. As we just proved, the user IDs for each of the individual searches can not be released in order to achieve this differential privacy. In addition, previous work [17] has show that it is possible to release queries in text format rather than in hash codes to preserve more privacy. However, a query log releasing algorithm can only be able to achieve ϵ-differential privacy, if it has a chance to release the queries that are not included in the original input query log. Otherwise, it may only be able to achieve (ϵ, δ)-differential privacy. In this section, we introduce our algorithm which is improved from Korolova et al. [17]. By using an external *querypool*, which contains extra potential queries that may be released by a query log releasing algorithm, we give every possible search query a chance to be released from the algorithm no matter it is included or excluded from the input query log Q. Therefore, we are able to improve and develop the algorithm to achieve ϵ-differential privacy.

Our query log releasing algorithm outputs a Q' containing the search queries (in original form) along with the query counts; clicked URLs corresponding to each of the distinct queries along with the clicked counts; and the query transitions with high transition frequencies in the original search sessions. For commercial search engines who are the releaser of real query logs, such an external query pool can be easily generated by a collection of all (or most) of the recorded queries in their database. In our project, we don't have such resource as the major commercial search engines. Hence, we simulate this process by using a collection of frequent English n-grams as the query pool.

Following are the major steps of our releasing algorithm.

- Sensitive information removal. Unique queries and terms with low enough frequency will be filtered out.
- Limiting amount of search queries for each of the users in the input query log. This guarantees each individual

user's search log won't make too much difference to the overall statistics.

- Extend the query candidates for releasing by the external query pool.
- Select queries to release base on the query counts with Laplacian noise.
- Release query counts and click counts with Laplacian noise added.
- Release query transitions information among the queries to be released from previous steps. Noise added to the transition counts.

We can prove that our query log releasing algorithm achieves ϵ-differential privacy. According to the length limitation of this workshop paper, we omit the proof here. In the next section, we will show our project framework on how we evaluate the utility of the released query log using our query log releasing algorithm.

3. PROJECT FRAMEWORK

As proposed in the previous section, we proposed an ϵ-differential private query log releasing algorithm. In order to evaluate how this releasing algorithm can help in typical IR tasks like document retrieval, we generate a framework to implement and evaluate our algorithm. In this framework, we first split the original query log into a training set Q and a test set Q_{Test}. Then, we propose a query log releasing algorithm $A(Q)$ that achieves ϵ-differential privacy. After that, we make use of the released query log $Q' = A(Q)$ to build a query-click graph that connects adjacent search queries in a session and corresponding clicked URLs for each of those queries. This query-click graph helps document retrieval for search queries in Q_{Test}. We compare the retrieved results with the ground truth to evaluate the quality of document retrieval, while the ground truth table is generated based on the actual clicked documents in the query log. In this way, we use mathematical derivation to get the privacy level of the query log releasing algorithm, while evaluating the IR utility of the released query log by how it performs in helping document retrieval in the test set. Now we show the major components of our project framework to evaluate the query log releasing algorithm in the rest of this section.

Data Partitioning. We use the AOL 2006 query log dataset for experiments. We first split the entire original query log into two parts, the training set Q and the test set Q_{Test}. Q is the raw query log from the search engine, and will act as the input for the releasing algorithm A and generates the released query log $Q' = A(Q)$. Q_{Test} will be used to evaluate how Q' can be used to help in document retrieval.

Query Log Releasing Algorithm. As shown in the previous section, we develop a query log releasing algorithm A, which takes the original query log Q as input, and outputs a privacy-preserving query log $Q' = A(Q)$. This is a query log releasing algorithm that achieves pure ϵ-differential privacy, which is improved from a previous approximate $(\epsilon, \delta)-$differential private algorithm from Korolova et al. [17].

Document Retrieval Our query log releasing algorithm partially preserves the sequential search session information by releasing high-frequency adjacent query transitions. Hence, we build a query-click graph to organize the queries and URLs. Nodes are queries, and URLs (documents) and edges, which connects query nodes with their corresponding clicked URL nodes, with clicking frequency as weights. We also connect two query nodes when they are adjacent query pairs according to the released query log, while the weights are the transition frequency. In addition, each query nodes also have highly weighted self loops connects to themselves. We do document retrieval for search queries in Q_{Test} based on a random walk model in the query-click graph, similar with Craswell and Szummer [7]. Starting from the query node, we let it random walk several steps within the query-click graph. Then, we rank the URLs according to the descending order of the probabilities of staying at corresponding URL nodes. By the way, it is worth noting that only those overlapped queries between Q_{Test} and Q' can be retrieved in this way.

Evaluations. We generate a ground truth table which contains query-URL pairs for evaluation use. This is based on the assumption that the actual clicked URLs represents the user's information need. We compare our retrieval results to the ground truth table and evaluate the performance by IR metrics like nDCG@10 [14], Precision@1, Recall@10, etc. Considering nDCG@10 as major evaluation metric, we make further experiments to evaluation our approach and explore the privacy-utility trade-off.

This finishes our project framework. In the next section, we will show our experimental results using our releasing algorithm and other baseline runs.

4. EVALUATION

Our experiments are based on the document retrieval task using the released query log. By comparing document retrieval evaluation results using query logs released with varying privacy guarantees, we can observe how the trade-off between privacy and utility works. Furthermore, we can propose recommendations for commercial search engines about their future query log release using our framework.

A natural baseline is to do document retrieval with the original (not private) query log. In addition to this, we also implement the k-anonymity approach from Carpineto and Romano [5] to compare with. The parameter k in k-anonymity means, only those queries appear in at least k different users can be released. Although k-anonymity preserves some privacy from the query log, it is not as strong as differential privacy. Table 1 shows the results of the experiments.

With different privacy levels, the size of released search log varies among runs, which leads the number of evaluated queries varying from run to run too. Generally, the releasing algorithm with stronger privacy will release smaller size query log and fewer evaluated queries. As we can see from Table 1, more than half (799,830 / 1,502,558) of the queries are released and can be evaluated. While our ϵ-Differential Private run and the k-Anonymity (k=10) run correspond to similar number of evaluated queries, our approach outperforms other runs in IR evaluation metrics like nDCG@10

Table 1: Utility evaluation among our Epsilon-Differential Privacy run, k-Anonymity runs, and the baseline run using original query log.

Runs	# Evaluated Queries	nDCG@10	Precision@1	Recall@10	Notes
ϵ-Differential Private	799,830	**0.6433**	**0.5846**	**0.6979**	Our Algrithm
k-Anonymity [5]	793,339	0.6365	0.5710	0.6977	k=10
k-Anonymity [5]	1,484,040	0.5876	0.5194	0.6526	k=5
Baseline	**1,502,558**	0.5914	0.5237	0.6559	Using raw query log Q, not private.

etc. This shows our ϵ-Differential Private run outperforms the k-Anonymity run in both privacy level and IR utility.

5. CONCLUSIONS

This project addressed the important security concerns in this query log releasing task. We present our ϵ-differential private algorithm to release query logs, and also provide a proof for why the user IDs in each individual search records can not be released in order to achieve such differential privacy. By analyzing the privacy issues during the query log releasing scheme, we make experiments to examine how useful the released query logs are. In this paper, we evaluate the IR utility of our query log releasing schemes based on the document retrieval task. Experiments show that our released query log is still very useful for document retrieval, and it outperforms the k-anonymity releasing scheme in both privacy and utility. More comparative experiments in our project also illustrates the privacy-utility trade-off in query log releasing process. Specifically, the stricter privacy standard we require, the lower utility we can maintain from the released query log. Since the high level privacy has been guaranteed by our ϵ-differential private query log releasing algorithm, we may recommend those commercial search engines to use softer parameter settings in our algorithm in order to maintain high utility of the released query log. We believe this project is an important step towards a final solution of releasing web search logs.

References

[1] E. Adar. User 4xxxxx9: Anonymizing query logs. In *Query Logs Workshop at the 16th WWW*, 2007.

[2] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*.

[3] M. Barbaro and T. Zeller. A face is exposed for AOL searcher no. 4417749. In *New York Times*, Aug 2006.

[4] F. Cai, S. Liang, and M. de Rijke. Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *SIGIR '14*.

[5] C. Carpineto and G. Romano. Semantic search log k-anonymization with generalized k-cores of query concept graph. In *ECIR'13*.

[6] A. Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. *ACM Trans. Web*, 2(4):19:1–19:27, Oct. 2008.

[7] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR '07*.

[8] A. Diriye, R. White, G. Buscher, and S. Dumais. Leaving so soon?: Understanding and predicting web search abandonment rationales. In *CIKM '12*.

[9] C. Dwork. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[10] L. Fan, L. Bonomi, L. Xiong, and V. Sunderam. Monitoring web browsing behavior with differential privacy. In *WWW '14*.

[11] H. A. Feild, J. Allan, and J. Glatt. Crowdlogging: Distributed, private, and anonymous search logging. In *SIGIR '11*.

[12] M. Gotz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Publishing search logs – a comparative study of privacy guarantees. *IEEE Trans. on Knowl. and Data Eng.*, 24(3):520–532, Mar. 2012.

[13] Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri. Effective anonymization of query logs. In *CIKM '09*.

[14] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.

[15] J. Jiang, D. He, and J. Allan. Searching, browsing, and clicking in a search session: Changes in user behavior by task and over time. In *SIGIR '14*.

[16] R. Jones, R. Kumar, B. Pang, and A. Tomkins. "i know what you did last summer": Query logs and user privacy. In *CIKM '07*.

[17] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW '09*.

[18] J. Luo, S. Zhang, and H. Yang. Win-win search: Dual-agent stochastic game in session search. In *SIGIR '14*.

[19] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *KDD '05*.

[20] M. Shokouhi, R. W. White, P. Bennett, and F. Radlinski. Fighting search engine amnesia: Reranking repeated results. In *SIGIR '13*.

[21] S. Zhang, J. Luo, and H. Yang. A pomdp model for content-free document re-ranking. In *SIGIR '14*.