A Review of Ten Year Research on Query Log Privacy

Claudio Carpineto and Giovanni Romano

Fondazione Ugo Bordoni, Rome, Italy
{carpinet, romano}@fub.it

Abstract. The research on search log anonymization is ten years old. Over this time, a number of methods to reduce the risks of user identification and disclosure of sensitive information from search log analysis have been proposed. However, the impact of these findings on the behaviour of data owners and publishers has been very limited so far. In this paper, we present a brief overview and a classification of the main approaches in the literature, arguing that there has been a shift towards a more pragmatic balance between the value of the data published and the risk of an adversary breaching the user's privacy. Then we go on to discuss what are the critical issues that must be tackled before search log anonymization moves out of research laboratories and into operational settings. We also clarify some misconceptions and mistakes in the use of the AOL search query log dataset, which is the main (and virtually only) experimental data source in the field.

1 Introduction

Search log data are extremely valuable for a number of applications but pose privacy risks. The infamous 2006 AOL incident, in which a user was identified from a search log with randomized user identifiers [2], spurred research into search log anonymization. Since then, a number of methods for limiting disclosure of personal information when publishing search logs have been proposed, based on different sanitization strategies such as scrubbing query content, removing infrequent queries, hashing, perturbing queries, injecting noise, and grouping a user's queries. These methods explore the trade-offs that exist between privacy guarantees and data utility, with better protection against information disclosure usually resulting in a reduced amount of useful data retained.

Despite such advances, the collectors of search log data have been reluctant to publish them for new data users. Ten years later the AOL incident, academic researchers continue to use copies of the AOL search query dataset, downloaded from mirror sites. One partial exception of which we are aware is represented by the datasets made available at the Workshops on Web Search Click Data in recent WSDM conferences, which are meant for evaluating search log mining algorithms and are thus fully anonymized (i.e., everything is a number). The concerns about publishing user data combined with a lack of mature privacy preserving information retrieval techniques have affected not only the research on search log analysis but also on other related tasks such as medical information retrieval and microblog retrieval [31].

In this paper we present a review of ten year research on query log privacy, including a fine classification and a discussion of key open problems. The only earlier survey of which we are aware of is [7], which provides a comprehensive discussion of several search log application tasks but focuses only on some early, out of date sanitization techniques. We also clear up a few technical things connected to interpretation and use of the AOL query log.

The remaining of the paper has the following structure. We first review the main sanitization methods, followed by a discussion of empirical evaluation of utility and privacy, and by a classification of approaches. Then we point out some open problems and presents the study concerning the AOL query log. We finally provide some conclusions.

2 Main sanitization approaches

Each record in a query log usually has the structure shown in Table 1. In this section we discuss how several methods delete or modify some of the fields in Table 1 to produce a sanitized query log that better protects the user's personal information. It is important to note that query log privacy is closely related to the older field of database privacy [12] – from which many concepts and frameworks have been borrowed and adapted – but has unique features. In databases, records are described by the same (numerical or categorical) attributes, a restricted set of which are manually labeled as quasi-identifiers (e.g., postal code, age, gender) or sensitive (e.g., salary, diseases, political views). By contrast, in search logs, we have very large sparse data (i.e., the query content), where every item is potentially quasi-identifier or sensitive.

Table 1. Typical structure of a query log record.

IP address	**Cookie ID**	**Query content**	**Timestamp**	**Browser/OS**	**Result Clicked**	**Rank**

2.1 Anonymizing identifiers

This approach consists of replacing well-established external identifiers (such as the user's IP address and/or cookie ID) with a numerical ID, through hashing or randomization, while the query content remains the same. It prevents certain types of privacy leaks such as when an IP in a query log record is correlated with the same IP in some web service, but it can be broken in a relatively easy manner, because the queries of a user are gathered and their contents remain unchanged. Thus, as witnessed by the AOL incident, the data for identifying a user and discover sensitive information can be found in her queries. Because both the user-query association and the query content are maintained, the sanitized log has a high utility.

2.2 Deleting identifiers

A simple approach to query log privacy is to delete the full IP addresses and cookie IDs, thus removing the explicit association between queries and users. This is a powerful tool in protecting user privacy, although there are other information (e.g., the user's browser and operating system configurations, timestamps, and query content) that can still be used to associate multiple queries with the same user. Also, even single queries may be detrimental to privacy if users query their own personal information. On the utility side, this approach does not permit several log analysis application.

2.3 Hashing queries

Hashing can be applied not only to external identifiers but also to single queries or elementary tokens within a query. This approach greatly helps to protect privacy because the original query is removed from the logs, although it may be possible to reverse-engineer particular query replacements by leveraging a statistical analysis of query frequencies gained from other available data sets [22]. The utility of the sanitized log is however clearly limited as most applications rely on the query content.

2.4 User clustering

The idea is to form clusters of at least k users that are similar in terms of their data, and then make all the users in a cluster indistinguishable from each other. Only the clusters are released, each with its own set of queries. Various similarity measures and clustering algorithms have been proposed.

In [16], agglomerative clustering is performed using a bipartite graph built from the queries and the click-through data. Then they create a set of queries for each cluster by adding similar query objects and deleting dissimilar query objects. In [28], they use microaggregation, with a distance function that integrates various types of query log data; e.g., query content, timestamps, clicked URLs. The queries released for each cluster are the centroid of the cluster. Divisive hierarchical clustering driven by WordNet is used in [15], inspired by a similar approach developed for set-valued data [33]. A user's queries are first generalized to the WordNet root concepts and then all the generalized profile queries are recursively partitioned top-down using more specific WordNet concepts to form the subpartitions, until no more partitions with clusters larger than k can be generated. The queries describing the clusters in the most specific partition are released. Another approach is [26], that works at the level of single terms and clusters users based on the similarity of their vocabularies, where the similarity between a pair of terms is topologically measured over a semantic network (e.g., WordNet) containing the terms.

These methods significantly reduce the risk of information disclosure when multiple relatively frequent queries are taken together. However, clustering rearranges the query log destroying the query ordering and creating fictitious sets of queries, which affects the utility of sanitized logs.

2.5 K-anonymity

One of the most fundamental concepts developed in the privacy field is k-anonymity, extensively studied in the the database community to prevent re-identification by multiple databases linking. It is assumed that a subset of attributes are quasi-identifiers and a record is released only if there are at least other k-1 records that share the same values for those attributes, which is usually achieved through generalization and suppression of attribute values [32]. As already remarked, query logs are fundamentally different from set-valued or relational data because there is no explicit distinction between quasi-identifiers and other types of information. Thus, the application of k-anonymity to search query logs [1] requires that a query serves as the quasi-identifier:

A query log L satisfies k-anonymity if for every query in L there exist at least k-1 identical queries in L issued by distinct users.

In this way, there is at most 1/k probability to link a query to a specific individual. However, this method leads to extreme data loss; e.g., about 90% of distinct AOL search log queries were issued by a single user.

To address the limitations of strict k-anonymity, one can try to protect a query with semantically similar rather than equal queries. Adapting earlier work on using WordNet to form privacy-enhanced user clusters [15], in [6] two different queries are replaced with their common WordNet parents and the generalized queries are arranged in hierarchical partitions characterized by decreasing levels of k-anonymity. This method ensures that more queries are released for a certain degree of k-anonymity. However, due to the limited coverage of WordNet, many queries cannot be generalized – e.g., 3,682,195 distinct AOL queries (out of 10,154,742) contain terms that do not match any WordNet concept [6] – and many released generalized queries consist of generic concepts such as 'event', 'thing', or just 'entity'.

Another variant of k-anonymity is k_{Θ}-affinity ([5], [6]), whereby infrequent queries are released based on their similarity to some frequent query of which they are a refinement (controlled by an additional similarity threshold parameter θ). Using this approach, it is possible to release many more queries in a presumably safe manner, given that [18] have estimated that about 40% of search log queries follow a 'Query+Refinement' pattern. K_{Θ}-affinity privacy is modeled as *generalized k-cores* of the graph of Θ-affine queries. In an experiment with the AOL data set, k-anonymity under affinity achieved similar levels of privacy as k-anonymity under equality and under WordNet generalization, while at the same time reducing the data losses to a great extent [6].

The enhanced versions of k-anonymity mitigate the data loss but are still unable to prevent the practical possibility of user identification through combination of multiple, relatively frequent queries. For example, by entering in a search engine like Google the main research interest of the first author of this paper (i.e., web search) and his affiliation, one gets his name several times in the first results page.

2.6 Removing sensitive query content

Personal identifying information such as personal name, email, birth date, address, credit card number, social insurance number, phone number, and others, can be automatically detected and removed from the original query log using several established tecniques such as named entity recognition and relation extraction; see e.g., [20]. However, a query log anonymized by removing certain entities may still be very vulnerable to privacy attacks. Jones et al. [19] showed that a simple classifier trained with registered Yahoo! users can map a sequence of AOL queries without names and numbers into the gender, age, location of the user issuing the queries, thus facilitating user identification.

Machine learning is indeed a viable technology for an adversary to discover leaked sensitive information, but the publisher can use the same means to suppress just those information predicted to be sensitive, thus reducing the potential for information disclosure. This is the approach taken in [23], which ensures a good protection against a state-of-the-art learning algorithm while retaining over 90% of the original data.

2.7 Differential privacy

Differential privacy [10] ensures that the removal or addition of a single database record does not significantly affect the outcome of any analysis. For query log data, differential privacy implies that the amount of knowledge that an attacker can learn about a user is roughly insensitive, according to some privacy parameters set by the data releaser, to omitting or changing the user's search history. This is modeled by requiring that for all pairs of search logs that differ in one user's searches, the probability that any subset is published is approximately the same for both search logs :

$$Pr[A(L_1) \in S] \;\leq\; e^{\epsilon} Pr[A(L_2) \in S] + \delta \tag{1}$$

where $L1$ and $L2$ are the two query logs, A is a differentially private algorithm, $Range(A)$ is the output range of A, and $S \subseteq Range(A)$.

Borrowing on this notion, a differentially private algorithm for publishing a query click graph was proposed in [21]. It consists of three main steps: (1) select a limited number of queries per user, (2) alter their frequencies by injecting Laplacian noise, (3) release only the queries with a frequency higher than a given threshold together with a noisy count of their clicked URLs.

Like k-anonymity, differential privacy does not modify the content of the single queries, but it changes their frequency and remove the association between queries and users.[1] Differential privacy is not an absolute privacy guarantee, but it is very general and powerful because no particular assumptions about an adversary's computational power or ability to access external data are required. On the other hand, for a typical choice of its parameters, it results in the

[1] The differential privacy model does not explicitly rely on the k anonymity parameter, although it can be modified in this direction [14].

suppression of all rare as well as relatively frequent queries, up to frequencies of the order of hundreds [21]. Furthermore, the destruction of the association between users and queries prevents some of the most interesting applications of published search log data. In fact, the utility of differential privacy has been deeply questioned [13] due to the huge involved data loss. A recent proposal extends differential privacy to preserve associations between users and queries, but it requires the specification of a particular objective function to be optimized [17]. Another recent refinement is concerned with reducing the high amount of noise that needs to be added to satisfy differential privacy for text databases, via sensitivity control [9].

3 Evaluation of utility and privacy

The utility of search logs is a broad concept, connected as it is to some benefit gained by analyzing the data. Some studies compare the performance of the original and released logs on certain data mining tasks such as clustering [28], or on some applications such as query substitution [13] and advertisement [4]. Abstracting away from a particular data utility, one common approach is to consider the percentage of released queries (aka impressions), which however provides only a rough indication of the utility of sanitized logs. A more principled approach is the Information Loss Ratio (ILR) [28], based on the difference of entropy between the original and released logs:

$$\mathrm{ILR} = \frac{\mathrm{H}(X) - \mathrm{H}(Y)}{\mathrm{H}(X)}, \qquad \mathrm{H}(X) = -\sum_{x} p(x) \cdot log\, p(x) \tag{2}$$

where X and Y represent, respectively, the set of original and released queries, and H is the entropy.

The need for similar global privacy measures is equally important, because given two sanitized logs produced by different methods one cannot say in which log the user privacy is more protected. One of the few global privacy measure is the Profile Exposure Level (PEL), proposed in [11] and applied to search logs in [28]:

$$\mathrm{PEL} = \frac{\mathrm{I}(X,Y)}{\mathrm{H}(X)} \cdot 100, \qquad \mathrm{I}(X,Y) = \sum_{x,y} p(x|y) \cdot p(y) \cdot log \frac{p(x|y)}{p(x)} \tag{3}$$

where $\mathrm{I}(X,Y)$ is the mutual information between X and Y. The ratio between mutual information and entropy is known in statistics as the uncertainty coefficient, and can be seen as a normalized mutual information. It gives a measure of the information that Y provides about X, normalized with respect to the information of X.

These measures compute some kind of difference between the set of a user's queries before and after sanitization, relating such a difference to the loss of utility (or gain of privacy). However, all queries are treated in the same manner.

It is just the probability distribution of queries that matters, not their content. While this general approach may be suitable for evaluating utility, in the intuitive sense that low difference approximately preserves the value of the original data over various usages,[2] it does not seem very appropriate for privacy. To evaluate the latter, we may be more interested in specific pieces of information that may lead to identify an individual (or disclose their sensitive information), rather than in the overall resemblance of the search logs. Take PEL for instance. It is easy to imagine a situation where the same PEL value corresponds to search logs with very different privacy risks, depending on whether potentially identifying queries with suitable distributions are present or not. Furthermore, the use of average measures for evaluating privacy can be questioned on the ground that the privacy guarantees are individually enforced by the anonymization algorithm (for an in-depth discussion of privacy versus utility measures for relational data see [3] and [25]). In the light of these shortcoming, it can be argued that well founded, empirical utility and privacy measures are yet to be devised.

4 A classification

In Table 2, we provide a classification of the main methods described in Section 2 along several dimensions, including preservation of user-query association and order of queries, degree of empirical privacy and utility (discussed in Section 3), list of enabled search log applications, and computational efficiency. Each method has strengths and weaknesses. The best choice depends on the emphasis on protecting the user privacy or retaining as much utility as possible, especially if the published log is intended to support specific applications.

5 Open problems

Lack of attack model. The database community has precisely defined several types of attacks (record linkage, attribute linkage, table linkage, probabilistic attack) and has provided anonymization techniques for each specific attack, e.g., k-anonymity [32], l-diversity [27], t-closeness [24]. In general, the available techniques can deal with only some of these attacks; see e.g. [12]. For query logs, the modelization of attacks is more vague. It is generally assumed that an attacker has unspecified background knowledge and inference abilities that may lead to user identification and disclosure of sensitive information. Future work should focus on how to model a realistic, resource-limited adversary; e.g., in terms of machine learning or information retrieval tools used to discover or rank sensitive information.

Pragmatic privacy guarantees. We have seen that several recent approaches including variants of k-anonymity and differential privacy have tried

[2] Note however that two logs with entirely different queries may well have the same entropy, thus yielding ILR = 0 and maximal empirical utility, where in fact the utility of one log relative to the other is null. This happens, for instance, if we use query hashing as a sanitization method.

Method	*References*	*User-query association*	*Order of queries*	*Formal privacy guarantees*	*Empirical privacy*	*Empirical utility*	*Task-based utility*	*Efficiency*
Publish as is		Yes	Yes	No	Minimal	Maximal	Ranking improvement, language-based applications, query refinement, personalization, anti-frauds academic sharing, commercial sharing	Maximal
Anonymizing identifiers		Yes	Yes	No	Minimal	Maximal	Ranking improvement, language-based applications, query refinement, personalization, academic sharing	High
Deleting identifiers		No	No	No	N/A	N/A	Weak language-based applications	High
Hashing queries	[22]	Yes	Yes	No	N/A	Maximal	Weak ranking improvement	High
User clustering	[15] [16] [28] [26]	No	No	No	N/A	N/A	Ranking improvement, weak language-based applications, weak query refinement	Low
Deleting sensitive queries	[19] [29] [23]	Yes	Yes	No	Low	High	Ranking improvement, language-based applications, query refinement, anti-frauds academic sharing, commercial sharing	Low
K-anonymity	[1] [5] [6]	Yes	Yes	No	Low	High	Ranking improvement, weak language-based applications, personalization, weak query refinement, anti-frauds, academic sharing, commercial sharing	Low
Differential privacy	[21] [14] [17] [9]	No	No	Yes	High	Low	Weak language-based applications, weak ranking improvement	High

Table 2. A fine classification of several search log sanitization methods.

to trade better levels of utility for reduced formal privacy guarantees. As full de-identification may be theoretically impossible, an interesting open problem is to explicitly relate the privacy guarantees to the inference power of an attacker, with the goal of making information disclosure too hard or costly for him/her. A first step in this direction is provided in [23].

Evaluation methodologies. The sanitization models developed so far enforce different privacy guarantees for different types of output. In addition, each model comes with its own set of parameters. A direct comparison of their results is thus very difficult. This issue is being addressed also in the database field [8]. Empirical measures of privacy and utility for text data, discussed in Section 3, are an attempt at mitigating this problem but more general and reliable evaluation techniques are needed. This problem is compounded by a lack of experimental benchmarks. Another related issue concerns a unifying framework capable to encompass various sanitization models or identification of the desirable theoretical properties of a generic sanitization mechanism.

Experimental datasets. In addition to a lack of benchmarks for experimental evaluation of privacy-preserving algorithms, researchers are often confronted with the paucity of annotated natural language datasets containing sensitive information; e.g., for training classifiers. These datasets are difficult to build, although there are recent works that automate this process to some extent; e.g., [29], [30]. There are also ethical issues involved here. Using only public data and never attempting to indentify users may not be enough, because an adversary could borrow the methods used to assess the privacy risk of an individual to select who to target. Also, publishing explicitly-sensitive anonymous profiles for research purposes poses a risk that someone could attempt to identify the individuals behind those data.

5.1 An examination of the AOL search query log dataset: clarifying and correcting the statistics

The AOL search log data set was retracted by AOL soon after its release due to privacy concerns, but it can still be downloaded from mirror sites. Because it is the only large data set of this kind available for testing to academic researcher, it has been used in a number of experimental studies in the last years. However, we discovered that the data set statistics reported in the literature have been obtained using some hidden and somewhat counter-intuitive assumptions and are not precise, as explained below. The data set contains 36,389,567 lines of data, of the format shown in Table 3. The field *AnonID* is an anonymous user ID number, *Query* and *QueryTime* are, respectively, the query issued by the user and the time at which the query was entered, and the last two fields (i.e., *ItemRank* and *ClickURL*) are present only if the user clicked on a search result. They are, respectively, the rank and the URL of the clicked item. Turning to the data set statistics, it is generally reported[3] that there are 21,011,340 instances

[3] http://www.researchpipeline.com/mediawiki/index.php?title=AOL_Search_Query_Logs .

of new queries (i.e., number of queries with repetitions). However, it is not explained how repetitions are computed. The apparent underlying interpretation is that two identical consecutive queries entered by the same user are seen as just one occurrence of the query, regardless of whether the user has clicked on some results or not and of the elapsed time between the two queries. Even under this interpretation, however, the available statistics are not entirely correct. We found out that the true number is 21,011,338 (instead of 21,011,340) due to a formatting mistake in the original data, as explained below. In line 2,586,379 of file 'user-ct-test-collection-08.txt', the *AnonID* begins with an empty space (see the middle row in Table 3). As the preceding and following queries are identical queries by the same user (i.e., 9403684), the three queries should count for one when computing the statistics. With this caveat, the number of queries amounts to 21,011,338. By contrast, if we used a wrong syntactic check of *AnonID* equality, we would obtain 21,011,338 + 2 = 21,011,340, which is the commonly used statistic. The relevant portion of data is illustrated in table Table 3.

Aside from this mistake, the requirement of non-consecutiveness does not seem to model the user behavior well, because this implies for instance that two identical queries with an elapsed time of 24 hours count for 1, while two identical non consecutive queries with an elapsed time of 30 seconds count for 2. It seems more convenient to update the frequency count even when two identical queries are consecutive, provided that their query time is different. Under this new interpretation, the number of queries grows to 28,898,361. The correct complete statistics are reported in Table 4. We would also like to point out that there are some very long meaningless query terms entered by multiple users. For instance, there are 37 query terms with 50 characters or more entered by at least two users. One of the most surprising query term is that formed by exactly 500 hyphens, entered by 18 users. These data are difficult to explain. One hypothesis is that multiple user ids may have been erroneously associated with the same user.

Table 3. Portion of AOL search query log data containing a formatting mistake that affects the statistics for new queries.

AnonID	Query	QueryTime	ItemRank	ClickURL
9403684	match.com	2006-03-31 06:48:21	1	http://www.match.com
9403684	match.com	2006-03-31 06:55:53	2	http://www.match.com
9403684	match.com	2006-03-31 06:55:53	2	http://www.match.com

6 Conclusions

The research on query log privacy has produced in ten years a number of insights. In this paper we have discussed strengths and weaknesses of existing methods, arguing that there has been a shift towards a more pragmatic approach to balance privacy guarantees and utility of sanitized logs. Today there are techniques

Table 4. The correct AOL search query log statistics. The modified parts are shown in bold, the new parts in italics.

36,389,567	lines of data
21,011,338	instances of new **non-consecutive** queries (w/ or w/o click-through)
28,898,361	*instances of new queries (w/ or w/o click-through)*
7,887,022	requests for next page of results
19,442,629	user click-through events
16,946,938	queries w/o user click-through
10,154,742	unique (normalized) queries
657,426	unique user ID's

that retain most of the utility of the original log at the cost of a very small privacy risk, based on a more realistic assessment of the inferential abilities of an adversary. The question remains as to these advances will be able to affect the visible behavior of the search log data collectors in the next future.

References

1. E. Adar. User 4xxxxx9: Anonymizing query logs. In *WWW Workshop on Query Log Analysis*, 2007.
2. M. Barbaro and T. Zeller. A face is exposed for aol searcher no. 4417749. *New York Times*, 2006.
3. J. Brickell and V. Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *14th SIGKDD*, pages 70–78, 2008.
4. T. Burghardt, K. Böhm, A. Guttman, and C. Clifton. Search-log anonymization and advertisement: are they mutually exclusive? In *CIKM*, pages 1269–1272, 2010.
5. C. Carpineto and G. Romano. Semantic Search Log k-Anonymization with Generalized k-Cores of Query Concept Graph. In *Proceedings of the 35th European Conference on Information Retrieval (ECIR 2013), ECIR 2013 shared Best Paper Award*, 2013.
6. C. Carpineto and G. Romano. K_θ-Affinity Privacy: Releasing Infrequent Query Refinements Safely. *Information Processing & Management*, 51:74–88, 2015.
7. A. Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. *ACM TWEB*, 2(4):1–26, 2008.
8. G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Empirical privacy and empirical utility of anonymized data. In *29th ICDEW'13*, pages 77–82, 2013.
9. W. Y. Day and N. Li. Differentially Private Publishing of High-dimensional Data Using Sensitivity Control. In *(ICCS '15)*, pages 451–462, 2015.
10. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third conference on Theory of Cryptography (TCC'06)*, pages 265–284, 2006.
11. A. Erola, J. Castella-Roca, A. Viejo, and J. Mateo-Sanz. Exploiting social networks to provide privacy in personalized web search. *Journal of Systems and Software*, 84(10):1734–1745, 2012.

12. B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (CSUR)*, 42(4):31–61, 2010.
13. M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Publishing Search Logs: A Comparative Study of Privacy Guarantees. *TKDE*, 24(3):520–532, 2012.
14. Feild H, J. Allan, and J. Glatt. CrowdLogging: distributed, private, and anonymous search logging. In *SIGIR*, pages 375–384, 2011.
15. Y. He and J. F. Naughton. Anonymization of SetValued Data via TopDown, Local Generalization. In *VLDB*, pages 934–945, 2009.
16. Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri. Effective anonymization of query logs. In *CIKM*, pages 1465–1468, 2009.
17. Y. Hong, J. Vaidya, H. Lu, and M. Wu. Differentially private search log sanitization with optimal output utility. In *Proceedings of EDBT 2012*, pages 50–61, 2012.
18. Y. Hu, Y. Qian, H. Li, J. pei, and Q. Zheng. Mining Query Subtopics from Search Log Data. In *SIGIR*, pages 305–314, 2012.
19. R. Jones, R. Kumar, B. Pang, and A. Tomkins. 'I know what you did last summer': query logs and user privacy. In *CIKM*, pages 909–914, 2007.
20. L. Korba, Y. Wang, L. Geng, R. Song, G. Yee, A. S. Patrick, S. Buffet, H. Liu, and Y. You. Private Data Discovery for Privacy Compliance in Collaborative Environments. In *CDVE'08)*, pages 142–150. Springer, 2008.
21. A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and click privately. In *WWW*, pages 171–180, 2009.
22. Ravi Kumar, Jasmine Novak, Bo Pang, and Andrew Tomkins. On anonymizing query logs via token-based hashing. In *WWW*, 2007.
23. B. Li, Y. Vorobeychik, M. Li, and B. Malin. Iterative Classification for Sanitizing Large-Scale Datasets. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM '15)*, pages 841–846, 2015.
24. N. Li, T. Li, and M. Venkitasubramaniam. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, 2008.
25. T. Li and N. Li. On the tradeoff between privacy and utility in data publishing. In *15th SIGKDD*, pages 517–526. ACM Press, 2009.
26. J. Liu and K. Wang. Anonymizing bag-valued sparse data by semantic similarity-based clustering. *Knowledge and Information Systems*, 35(2):435–461, 2013.
27. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd ICDE*, 2006.
28. G. Navarro-Arribas, V. Torra, A. Erola, and J. Castella-Roca. User k-anonymity for privacy preserving data mining of query logs. *IPM*, 48:476–487, 2012.
29. S. T. Peddinti, A. Korolova, E. Bursztein, and G. Sampemane. Cloak and Swagger: Understanding Data Sensitivity Through the Lens of User Anonymity. In *Proceedings of the 2014 IEEE Symposium on Security & Privacy (SP '14)*, pages 493–508, 2014.
30. S.T. Peddinti, K. W. Ross, and J. Cappos. Finding Sensitive Accounts on Twitter: An Automated Approach Based on Follower Anonymity. To appear in the 10th International AAAI Conference on Web and Social Media (ICWSM), 2016.
31. Luo Si and Hui Yang. PIR 2014 The First International Workshop on Privacy-Preserving IR: When Information Retrieval Meets Privacy and Security. *SIGIR Forum*, 48(2):83–88, 2014.
32. L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.
33. M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving Anonymization of Set-valued Data. In *VLDB'08, Auckland, New Zeland*, pages 115–125, 2008.