Query Log Anonymization by Differential Privacy

A Thesis
submitted to the Faculty of the
Graduate School of Arts and Sciences
of Georgetown University
in partial fulfillment of the requirements for the
degree of
Doctor of Philosophy
in Computer Science

By

Sicong Zhang, M.S.

Washington, DC
October 25, 2017

Query Log Anonymization by Differential Privacy

Sicong Zhang, M.S.

Thesis Advisor: Dr. Grace Hui Yang

Abstract

Web search query logs, which record the interactions between the search engine and its users, are valuable resources for Information Retrieval (IR) research. For years, such query logs have been supporting multiple IR applications and have significantly promoted the advance of IR research. However, releasing query logs without proper anonymization may lead to serious violations of user privacy. As a result, concerns about user privacy have become major obstacles preventing these resources from being available for research use. This dissertation addresses the challenge of query log anonymization, in order to keep advancing IR research.

Particularly, this dissertation presents my research on query log anonymization by differential privacy. Anonymization of query logs differs from that of structured data because query logs are generated based on natural language, whose vocabulary is infinite. To mitigate the challenges in query log anonymization, I propose to use a differentially private mechanism to generate anonymized query logs containing sufficient contextual information for existing web search algorithms to use and attain meaningful results. I empirically validate the effectiveness of my framework for generating usable and privacy-preserving logs for web search. Experiments show that it is possible to maintain high utility for this task while guaranteeing sufficient privacy.

In addition, this dissertation also proposes my expended research on query log anonymization to involve session data. My previous work on session search has shown that such search sessions are essential resources to support complex IR tasks. Although researchers have recently proposed approaches to histogram-based data release of query logs, how session data in query logs can be released differentially privately with meaningful utility remains unclear. By proposing a differentially private query log anonymization algorithm to release session data, my research resolves this significant concern about how to properly release and use the session information of query logs. Moreover, I use two typical IR applications, query suggestion and session search, to examine utility of anonymized logs and privacy-utility tradeoff of the session-based query log anonymization work.

In summary, by resolving concerns in both privacy and utility aspects, this dissertation provides theoretical frameworks and practical implementations of query log anonymization by differential privacy. It serves as an important step towards an ultimate solution to the general challenge of data anonymization in real-world IR applications. I hope this work can not only benefit the research in this particular task of query log anonymization but also inspire more research in privacy-preserving Information Retrieval (PPIR) and other data-driven research domains.

INDEX WORDS: Differential Privacy, Information Retrieval, Query Log Anonymization, Document Retrieval

Acknowledgements

It is my great pleasure to express my deepest gratitude to those who made this Ph.D. dissertation possible.

First and foremost, I am deeply grateful to my advisor, Professor Grace Hui Yang for her continuous support and inspiring instruction throughout my entire Ph.D. study. Professor Yang is a great advisor with motivation, enthusiasm, inspiration, and patience. Professor Yang introduced and inspired me to explore the great research area of Information Retrieval and patiently helped me to shape my interests, ideas, and insights. To me, Professor Yang is not only my academic advisor, but also a lifetime mentor.

Besides my advisor, I wish to express my sincerest gratitude to the rest of my thesis committee: Professor Lisa Singh, Professor Micah Sherr and Professor Li Xiong for their insightful comments and valuable advice. More importantly, they are also my most important collaborators in research. I greatly benefitted from the valuable collaborations with each of them. I would also like to express my special gratitude to Professor Lisa Singh for her warm encouragement all these years.

I am indebted to my academic collaborators for our collaborations in research projects and academic events: Jiyun Luo, Dr. Dongyi Guan, Dr. Xuchu Dong, Dr. Yifang Wei, Dr. Ian Soboroff, Dr. Luo Si, Dr. Charles L. A. Clarke, Dr. Simson L. Garfinkel, Dr. Jun Wang, Dr. Marc Sloan, Janet Zhu, Andrew Hian-Cheong, Kevin Tian, Tavish Vaidya, Elchin Asgali, and Lei Cen.

I would like to give my sincere thanks to professors at Georgetown that have offered great lectures to me: Dr. Jeremy Fineman, Dr. Ophir Frieder, Dr. David D. Lewis, Dr. Calvin Newport, Dr. Nazli Goharian, Dr. Mark Maloof, Dr. Adam O'Neill, Dr. Kobbi Nissim, Dr. Nathan Schneider, Dr. Richard Squier, and Dr. Mahlet Tadesse. My sincere thanks also go to professors and staff members in the department for their great support: Dr. Evan Barba, Dr. Jeremy Bolton, Dr. Eric Burger, Dr. Der-Chen Chang, Dr. Bala Kalyanasundaram, Dr. Barrett Koster, Dr. Jami Montgomery, Dr. Justin Thaler, Dr. Clay Shields, Dr. Mahendran Velauthapillai, Dr. Wenchao Zhou, Dr. W. Addison Woods, Woonki Chung, Sheilynn Brown, Larita Williams, and Dejah McCrimmon. I would also like to express my special gratitude to professors at Tsinghua University who ignited my passion in academic research: Dr. Zhiyuan Liu, Dr. Maosong Sun, Dr. Xiaoyan Zhu, Dr. Minglie Huang and Dr.Yiqun Liu.

My sincere thanks also extend to friends in the InfoSense group and in the department for their special support and inspiring discussions that greatly helped my research and life: Shiqi Liu, Razieh Rahimi, Amin Teymorian, Lingqiong Tan, Zhiwen Tang, Chong Zhang, Shuchen Zhu, Christopher Wing, Jianxian Wu, Hongkai Wu, Angela Yang, Hao-Ren Yao, Yuankai Zhang, Jie Zhou, and Yanan Zhu. I would like to express my special gratitude for Yanan's accompany, support and encouragement. I would also like to thank many many more friends from Georgetown University, Tsinghua University, the IR research community, Facebook Inc. and elsewhere for their great support and kind help.

I would like to express my sincere gratitude to organizations that have sponsored or supported my research: National Science Foundation (grant CNS-1223825 and IIS-145374), DARPA (grant FA8750-14-2-0226), SIGIR Student Travel Grant, Georgetown University, and the Department of Computer Science at Georgetown University.

I owe my deepest gratitude to my parents and the entire family for their support, encouragement, love, and understanding. I am the first Ph.D. in my family. My mother insists that I am special, and my father always believe that I will achieve great success in computer science. Without them, this dissertation would not be possible. I dedicate this dissertation to my family.

At the end of this acknowledgement, I would like to share a memorable portion of the famous speech from J.F. Kennedy, with all friends who have started or wish to start a Ph.D. journey: “We choose to go to the Moon in this decade and do the other things, not because they are easy, but because they are hard; because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one we intend to win."

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are living in an era of enormous data sharing and data availability on the Internet. Nowadays, the data pervasiveness has resulted in the emergence of services tailored to extract, search, aggregate, and mine data in meaningful ways [132]. While it is likely that many users of online services understand that they are sharing personal information with strangers, they may not understand the potential risks and implications of doing so. In fact, high levels of exposed information can lead to severe consequences such as stalking [101], identity theft [104], and job loss [114]. Moreover, it has been a growing research topic to understand the data that users are willing to share and the level of sensitivity associated with it. For instance, there has been an emerging interest in linking individuals across online social networks [38, 49, 50, 51, 52, 74, 77, 82, 86, 127]. My previous work [98, 136] also examines this problem of quantifiable measuring of online privacy risks. It is clear that there have been more and more privacy concerns about web users' online data.

Being more specific, the rapid development of big data, social networks, mobile services and the growing popularity of digital communications have also profoundly changed Information Retrieval (IR) research. Many recent advances in IR research rely

on sensitive and private data such as large-scale query logs, users' search history, and location information. It is understandable that the sensitive and private data is kept within the commercial companies without being shared with the research community in general. However, the concern of privacy sometimes is so overwhelming that it has hurt IR research in the past. For instance, the TREC Medical Record Retrieval Tracks [112] are halted because of the privacy issue and the TREC Microblog Tracks [73] could not provide participants with a standard testbed of tweets to ensure a fair comparison. The proper use of privacy techniques to empower privacy-preserving IR [125] research should be studied promptly.

It is perfectly understandable but still disappointing that user privacy has become an obstacle that prevents data release and interferes with the advance of research in IR. It is a boon for IR researchers that the large volume of textual data offers the perfect playground for developing new information retrieval algorithms. However, the sharing of large amounts of data, some of which are sensitive, presents challenges with regards to data privacy.

In this dissertation, I present my research on query log anonymization by differential privacy to address such privacy concerns and resolve obstacles. In this chapter, I first give a general introduction to the background and challenges in query log anonymization. After presenting my high-level solutions to those challenges, I will present the major tasks involved in this thesis. At the end of this chapter, I give the outline of the entire thesis.

1.1 Background

After the release of the web's first primitive search engine *W3Catalog* in 1993, search engines have significantly changed our daily life and became the most popular way to search for information. Nowadays, massive web users all over the world frequently interact with the search engine by submitting search queries, reviewing the search result pages, and clicking some of the retrieved webpages. In the meantime, the search engines record major interactions with their users to form an important type of log data, which is the web search query log. As a type of large-scale online user data, query log becomes very valuable and have been widely used to promote IR research. However, privacy concerns about the data also raised according, especially when releasing the query log to a third party. How privacy risks influence the search engine and its users? How privacy risks influences IR research? What earlier attempts have been proposed for query log anonymization? I provide these background as follows.

1.1.1 How privacy risks influence the search engine and its users

Releasing query logs without proper anonymization may lead to serious violations of users' privacy. As a consequence, the violations of users' privacy may also ruin the reputation of the search engine. A severe incident happened in 2006 when America Online (AOL) released an insufficiently anonymized version of their query log [1] and raised serious social, legal, and financial issues for the company.

The only "anonymization" technique applied to this AOL query log was to replace the user id value by a hash code. This technique was far from keeping the user information anonymized because an adversary can learn a lot by combining all the search records from each user according to the hashed user id. Soon after the AOL release, the identity of an old widow was identified from the log reported by a *New York Times* article as shown in Figure 1.1[1] which says: *"It did not take much investigating to follow that data trail to Thelma Arnold, a 62-year-old widow who lives in Lilburn, Ga., frequently researches her friends' medical ailments and loves her three dogs... AOL removed the search data from its site over the weekend and apologized for its release, saying it was an unauthorized move by a team that had hoped it would benefit academic researchers...'My goodness, it's my whole personal life', she said. 'I had no idea somebody was looking over my shoulder.' In response, she plans to drop her AOL subscription. 'We all have a right to privacy,' she said. 'Nobody should have found this all out.' "* This incident raised severe social and legal issues for AOL due to the massive privacy concern from the public. Actually, as I mentioned earlier, exposed personal information may lead to consequences such as stalking [101], identity theft [104], and job loss [114]. Since then, web search companies have refused to release any query logs, even for research purposes. Table 1.1 shows a sample of the AOL query log.

[1]http://www.nytimes.com/2006/08/09/technology/09aol.html?_r=1&

Table 1.1: A sample of the AOL query log.

UserID	Query	Query Time	Webpage Rank	Clicked Web Page
479	family guy movie references	2006-03-03 22:37:46	1	http://www.familyguyfiles.com
479	top grossing movies of all time	2006-03-03 22:42:42	1	http://movieweb.com
479	top grossing movies of all time	2006-03-03 22:42:42	2	http://www.imdb.com
479	car decals	2006-03-03 23:20:12	4	http://www.decaljunky.com
479	car decals	2006-03-03 23:20:12	1	http://www.modernimage.net
479	car decals	2006-03-03 23:20:12	5	http://www.webdecal.com
479	car window decals	2006-03-03 23:24:05	9	http://www.customautotrim.com
479	car window sponsor decals	2006-03-03 23:27:17	3	http://www.streetglo.net
479	car sponsor decals	2006-03-03 23:28:59		
479	car brand name decals	2006-03-03 23:29:35		
479	brand name decals	2006-03-03 23:29:58		
479	bose	2006-03-03 23:30:11	1	http://www.bose.com
479	bose car decal	2006-03-03 23:31:48	1	http://stickers.signprint.co.uk
479	bose car decal	2006-03-03 23:31:48	1	http://stickers.signprint.co.uk
479	bose car decal	2006-03-03 23:31:48	7	http://www.motorcitydecals.com
479	chicago the mix	2006-03-04 22:11:31	1	http://www.wtmx.com
479	chicago the drive	2006-03-04 22:14:51	2	http://www.wdrv.com
479	chicago radio annoucer whip	2006-03-04 22:16:07		
479	chicago radio whip	2006-03-04 22:16:27		
479	chicago radio brian the whipping boy	2006-03-04 22:17:00	1	http://www.djheadlines.com
479	emma watson	2006-03-04 23:05:53	1	http://www.imdb.com
479	stanford encyclopedia of philosophy	2006-03-06 21:57:14	1	http://plato.stanford.edu
479	internet encyclopedia of philosophy	2006-03-06 21:59:30	1	http://www.iep.utm.edu
479	www library philosophy	2006-03-06 22:01:29	2	http://www.bris.ac.uk
479	allegory of the cave	2006-03-06 22:03:19	1	http://faculty.washington.edu
479	allegory of the cave	2006-03-06 22:03:19	2	http://www.wsu.edu:8080
479	allegory of the cave	2006-03-06 22:03:19	6	http://en.wikipedia.org
479	citation machine	2006-03-06 22:57:22	1	http://citationmachine.net
479	howard stern lawsuit	2006-03-08 00:14:55		
479	sirius playboy	2006-03-08 17:23:07	3	http://www.orbitcast.com
479	opec	2006-03-09 09:19:29		
479	citation machine	2006-03-13 18:25:12	1	http://citationmachine.net
479	wto history	2006-03-13 18:26:09	1	http://depts.washington.edu
479	wto history	2006-03-13 18:26:09	3	http://www.pbs.org
479	wto history	2006-03-13 18:26:09	4	http://www.wto.org
479	wikpedia	2006-03-13 18:29:21	1	http://en.wikipedia.org
479	britannica	2006-03-13 18:41:09	1	http://www.britannica.com
479	wto	2006-03-13 18:44:23	1	http://www.wto.org
479	wto history	2006-03-13 18:48:27		
479	wto history	2006-03-13 18:48:35	1	http://depts.washington.edu
479	wto history	2006-03-13 18:48:35	4	http://www.wto.org
479	wto history	2006-03-13 18:48:35	6	http://www2.netdoor.com
479	wto history	2006-03-13 18:48:35	7	http://www.econ.iastate.edu
479	wto history	2006-03-13 18:48:35	9	http://cyberjournal.org
479	wto history	2006-03-13 18:48:35	2	http://depts.washington.edu
479	wto history	2006-03-13 18:48:35	4	http://www.wto.org
479	www.galleryhost.com	2006-03-15 01:13:14		
479	prairie state college student email	2006-03-20 23:15:59	3	http://students.prairiestate.edu
479	ronny van zant	2006-03-22 00:23:17		
479	ronnie van zant	2006-03-22 00:23:55		
479	zack wylde	2006-03-22 00:25:01		
479	zack wylde lead singer	2006-03-22 00:26:26		
479	ozzy guitarist	2006-03-22 00:27:44	1	http://www.ultimate-guitar.com
479	www.whitesox.xom	2006-03-22 11:38:16		
...	...	...	...	...

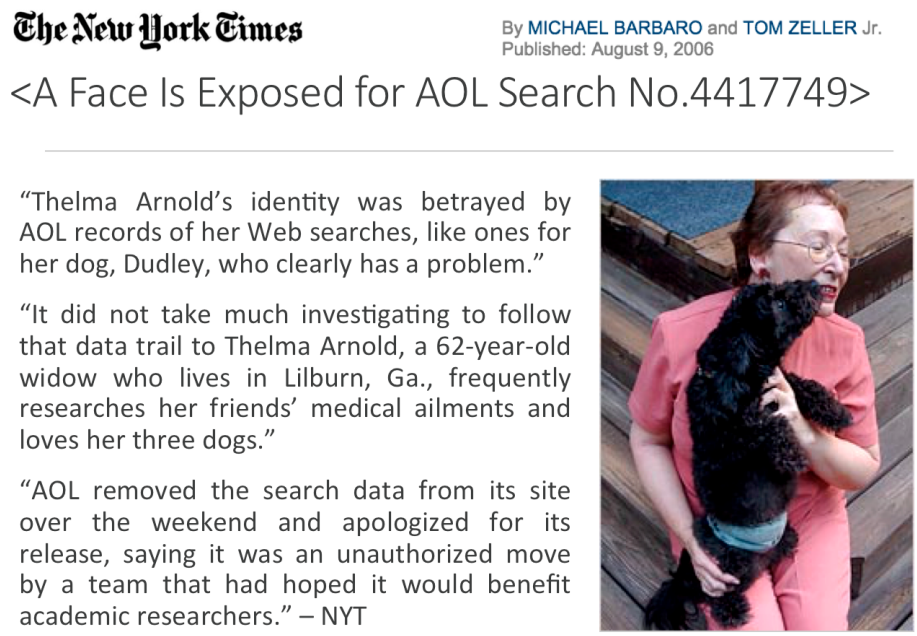


Figure 1.1: A user's identity was identified from the released AOL query log, according to an article from *The New York Times*.

1.1.2 How privacy risks influence IR research

Privacy issue have significantly influenced the advance of IR research. For years, due to the lack of mature techniques in privacy-preserving IR, concerns about information privacy and security have become serious obstacles that prevent valuable user data from being used in IR research such as studies about the increased information leakage from text [103, 132], public information exposure on social media and online platforms [38, 50, 52, 97, 98, 127, 136], query log anonymization [1, 21, 39, 66, 134, 135], and medical research work [17]. The situation needs to be improved promptly.

Consequently, large and real query logs can be found within only a few commercial companies. Such data dominance might have already impacted negatively on the research field as a whole. I suspect an increasing split in our community – academic researchers who have no access to query logs and could not conduct related research vs. industrial researchers who have the data but miss the opportunities to learn more diversified ideas from their academic colleagues. I believe that this split could be one of the reasons that premium IR conferences such as SIGIR (Special Interest Group on Information Retrieval) are experiencing a decline.[2] Researchers probably won't be able to change the situation in one day. But I hope the issue of privacy in IR could be alleviated at least from a purely technical point of view. I am therefore highly motivated to propose query log anonymization methods to enable data release for research in IR.

1.1.3 Earlier attempts for query log anonymization

Existing work on query log anonymization has attempted to protect the privacy of search logs in many ways. However, those existing attempts are far from satisfactory. For instance, Adar [1] and Carpineto et al. [15] used clustering techniques and k-anonymity, which assumes each query to be issued by at least k different users, to anonymize query logs. The limitation of a k-anonymity approach is that its privacy guarantee can be easily broken when an adversary knows information about the users from an unexpected source. When an adversary knows more about the user than what

[2] http://sigir.org/files/forum/2016J/p001.pdf

Table 1.2: A sample of the anonymized query log from Web Search Click Data workshop. (WSCD 2014).

SessionID	SERPID	QueryID	ListOfURLs		
34573630	0	10509813	34175267	34171511	35444452
			15370141	31342884	43630531
			26065978	29902424	39016998
			62861215		
34573635	0	8447254	44298735	41815016	62677540
			13753389	3336907	67724115
			22354391	4606079	37985498
			53161116		
...	...	...	...		

the k-anonymity algorithm assumes, the adversary could join the unexpected source with existing ones and break the privacy guarantee.

A few other attempts have also been made to alleviate the lack of available search log data. For example, in 2014 Yandex shared an anonymized query log (Table 1.2) for a web search challenge at the Web Search Click Data (WSCD) 2014 workshop to support document re-ranking.[3] In this released query log, all words were converted to hash codes, reducing the utility of the released log significantly. The only web search task that can be researched with this data set is document re-ranking [11, 85, 95]. Because the contextual data has been removed, the data set is not useful for any other external IR use. Furthermore, this anonymized query log still suffers from privacy risks. For instance, if some hash codes are matched to original data according to frequency distributions, the original search logs from the user may be easily re-

[3] `http://research.microsoft.com/en-us/um/people/nickcr/wscd2014/`

identified. A stronger privacy notation is still needed in query log anonymization to achieve a good balance between privacy and utility.

Recently, differential privacy (DP) has been emerging in query log anonymization. DP is effective in anonymizing statistical data. It is nice that its privacy guarantee can be mathematically proved. By adding noise to samples of data in the dataset, the goal of DP is to create a disturbed (anonymized) dataset that is able to hide the information of each individual in such a way that no one could tell whether the individual exists in the dataset or not. The intuitive idea of differential privacy is that if no one could tell if a user exists in a dataset, it would be even more difficult to find the user out. It is like what is described in *Platform Sutra – if fundamentally there is not a single thing, where could any dust be attracted?* This is perhaps the strictest way to protect data privacy since the data "seems" to no longer exist. Due to its strong privacy protection, there is no need for DP to make any assumption about an adversary's knowledge or the method an adversary would use to attack the data. It makes DP feasible in practice to help query log anonymization.

Furthermore, although there exists recent research work on query log anonymization, how session data could be properly anonymized and utilized in IR applications remains an open question. The session data, as a special form of sequential data in the query log, contains important information about the original web search retrieval process. It is a helpful complement to the click-through data in query logs. Without session data, a properly anonymized query log may be used to support

simple IR applications such as ad-hoc search or simple web mining applications such as website clustering. A properly anonymized query log containing both click-through data and session data can be used to support more complex IR applications such as query suggestion and session search. Therefore, I am motivated to research query log anonymization for both click-through data and session data.

1.2 Challenges

The challenges of query log anonymization rise from privacy guarantee, data utility, and risk quantifiability.

A major challenge in query log anonymization is to provide a mathematically proved sufficient privacy guarantee. The AOL incident has shown that the removal (hashing) of user id is not working at all. A good privacy-preserving query log anonymization mechanism should provide proved privacy so that any adversary could not gain much information about the identity of each person whose data appeared in the log. This challenge is hard to achieve because the potential adversary may use unexpected external data to analyze or attack our anonymized query log. For instance, a query frequency distribution may be used to attack an anonymized query frequency distribution even if the query itself was removed or hashed. Another example is that the adversary may even get the exact search log of a specific web user from unknown sources. Then, the user's identity may be revealed if a piece of our anonymized query

log matches exactly with the information obtained by the adversary. A strong privacy mechanism must be used to resolve this challenge.

Another challenge in query log anonymization is to maintain enough IR utility of the data. It is important since a privacy-preserving technique only makes sense if enough utility of the data can be maintained. For instance, a trivial privacy-preserving "technique" that many data owners are currently using is to "release nothing" to achieve absolute privacy. However, no utility is left after such a "technique". Another example is the Yandex example I just mentioned (Table 1.2). While transforming everything into hash codes, the anonymized query log contains no textual data with real natural language meaning, which makes the query log almost useless for other research tasks depending on natural text. An ideal privacy-preserving technique should maintain enough utility in the output data to support research in different applications.

Last but not least, the privacy risk of the query log anonymization mechanism should be quantifiable. In other words, the privacy level of a good privacy-preserving mechanism should be clearly quantified by privacy parameters. To address privacy concerns from the public, such quantifiability is the guarantee that a good privacy-preserving technique can be accepted by the public or any individual user of the corresponding web service. Practically, when the data owner is preparing for a data release, the tradeoff between privacy and utility must be considered. Quantifiable

privacy risk may provide the data owner an opportunity to make better judgments of the privacy-utility tradeoff.

With all the challenges in mind, I explore the interesting topic of query log anonymization and propose my solutions to address such challenges.

1.3 Solutions

In this thesis, I propose to use differential privacy [28, 66] to anonymize a query log. Differential privacy is the state-of-the-art approach which provides a strong privacy notion. It has been widely used in the database and data mining communities [71, 93, 106]. Differential privacy provides guarantees which can be theoretically proved that every individual user in the datasets would not be identified. Unlike k-anonymity, differential privacy does not make assumptions about the amount and scope of an adversary's background knowledge.

Moreover, most existing work in query log anonymization [66] measured the utility of the anonymization output regarding the size of the remaining logs, without systematically measuring the utility that is directly related to retrieval performance. It is thus difficult to tell how much utility is left in the query logs after anonymization regarding how useful the logs are when we use them to retrieve relevant documents in a web search algorithm. In this work, I propose the retrieval utility function from the viewpoint of a search engine to report the actual usefulness of query logs after anonymized with DP.

To improve the privacy level of query logs to match the specifications of DP, a search record might be removed or modified into a set of statistics. However, we need to be aware that such changes made on the original data only make sense if the remaining logs can provide enough information to be useful, in our case, to still be able to support web search.

I propose that we need to keep the web search queries in natural language form in the anonymized query logs. They are kept in the textual format as they are. Low-frequency queries are removed since they are too unique and greatly increase the chance to break the privacy guarantee if they stay. Next, the click-throughs are also key data in a query log. However, they can only be released in a statistical format in order to achieve differential privacy. I aggregate all the click-throughs and show them as summary counts. Furthermore, highly identifiable features such as the user ids are removed during anonymization. Therefore, they are not shown in the output log. Moreover, I develop a query log anonymization mechanism to maintain session data in the anonymized logs. This allows researchers to make use of the anonymized query logs in more complex research tasks that require session data or query sequence data.

1.4 Tasks

In this dissertation, I focus on the general task of query log anonymization for single queries first and then move on to tackle query log anonymization for sessions.

1.4.1 Query log anonymization for single queries

Compared to databases or structured data, query logs have an infinite vocabulary. This unstructured, unbounded dataset poses different challenges. To mitigate these challenges, I propose to use a differential privacy framework to generate anonymized query logs that contain sufficient contextual information to allow existing web search and web mining algorithms to use the data and attain meaningful results. I empirically validate the effectiveness of our framework for generating usable, privacy-preserving logs for web search and demonstrate that it is possible to maintain high utility for web search while providing sufficient levels of privacy. Furthermore, in order to examine how these anonymized logs can be used to support other web mining tasks, I also show some preliminary results for web document clustering as future work. Following are the major challenges in query log anonymization for single queries:

- What is the best way to anonymize query logs with differential privacy?
- How can we deal with the infinite domain of queries, since a web search query could be any combination of free text?
- How can we use real IR applications to evaluate the utility of anonymized query logs?
- How can we get a good balance between privacy and utility?

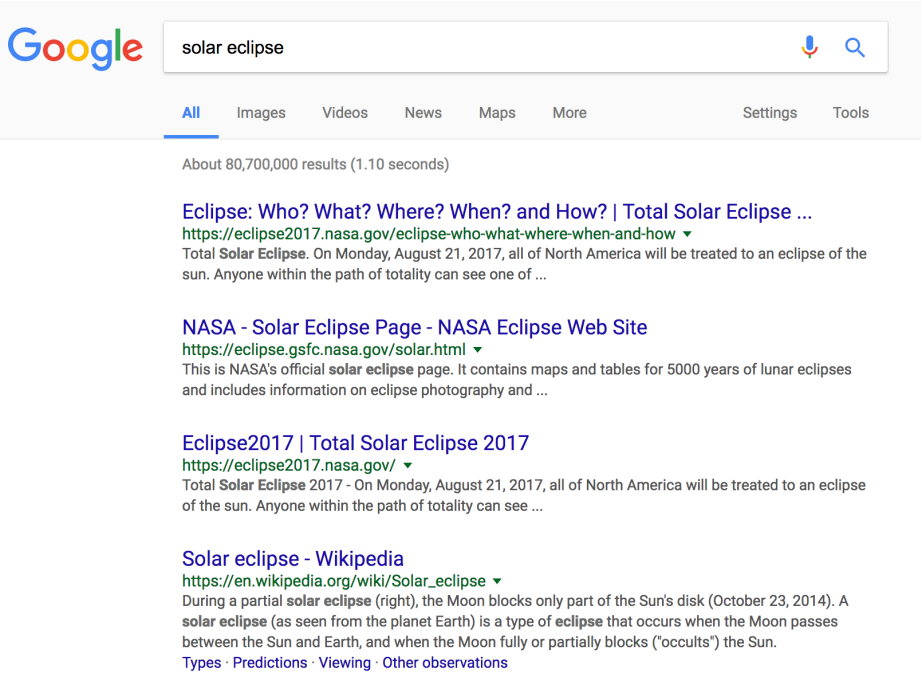


Figure 1.2: Ad-hoc web search results by Google on Sep 8th, 2017.

I use the typical IR applications of ad-hoc web search to evaluate the utility of the anonymized query log in this task. Figure 1.2 presents an example of ad-hoc search by Google in 2017.

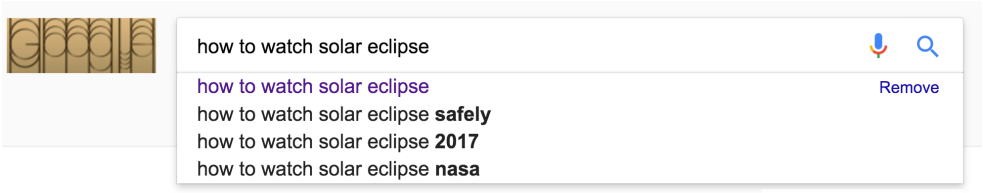
1.4.2 Query log anonymization for sessions

In addition to query log anonymization for single queries, I expand the research to involve session data. Search sessions are special log data that may reveal the intention hierarchy of users' online behavior. The session-based query log anonymization algorithm will also release search sessions as additional output. It will be able to support

complex IR applications that require query sequences. My recent work on session search have shown that such search sessions are very important resources to support complex IR tasks and advanced approaches. Although researchers in our community have recently proposed a few approaches on histogram-based data release of query logs, how session data in the query log can be released differentially privately with meaningful utility remains unclear. My research resolves such a major concern about how to properly release and use the search session information of query logs. I use two typical IR applications, web search and query suggestions, to examine the privacy-utility feedback of the session-based differentially private query log anonymization work. Following are the major challenges in this task:

- How can we keep session information or sequential data in differentially privately anonymized logs?
- How well can anonymized query logs containing frequent search sessions be used to support complex IR applications such as query suggestion and session search?

I use query suggestion as the major IR application to evaluate the utility of the anonymized session log. Query suggestion is a typical task in IR. The general setting of query suggestion is to predict the following query (or queries) that the user is going to submit to the search engine, given the previous search log of the user. Figure 1.3 presents common query suggestion examples from Google. In typical search engines such as Google, the results of query suggestion usually appear in the search box as in



(a) Query suggestion at the search box

(b) Query suggestion at the end of the search page

Figure 1.3: Query suggestion examples by Google on Sep 8th, 2017.

Figure 1.3(a) or at the end of the Search Engine Results Page (SERP) as in Figure 1.3(b).

I use session search as another IR application to evaluate the utility of the anonymized session log. Session search is a complex search task that makes use of all queries and user interactions in a search session to improve retrieval effectiveness for the whole session. Figure 1.4 presents examples of the interactive process of session search between the web user and the search engine in the Text RetriEval Conference (TREC) 2012 Session Track dataset [62].

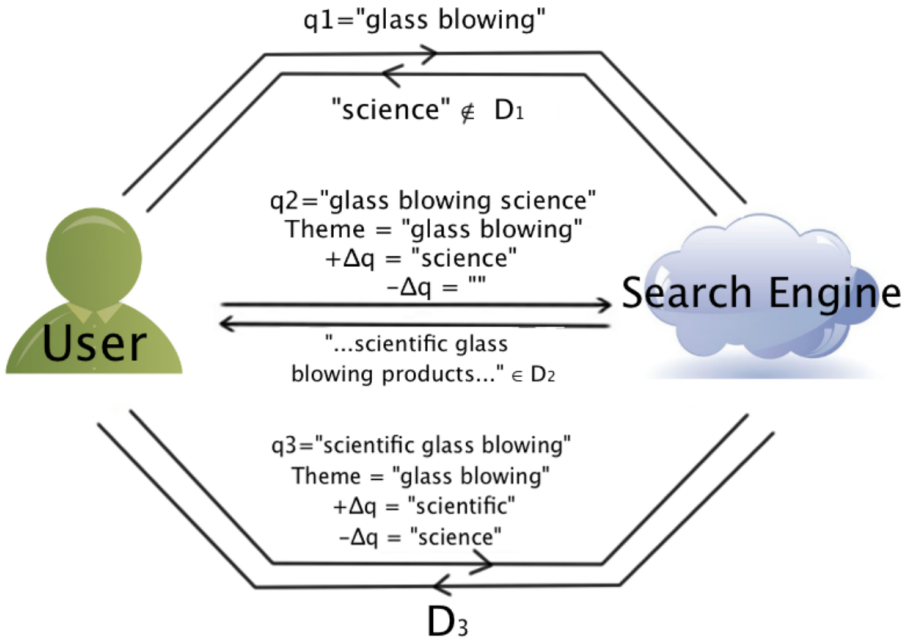


Figure 1.4: Search Session: The interactive process between the web user and the search engine. (S85 From the TREC 2012 Session Track)

1.5 Outline

By providing theoretical frameworks and practical implementations of query log anonymization, this dissertation shows query logs can be anonymized differentially privately to guarantee privacy, while such query logs anonymized by differential privacy contain enough utility to support real IR applications as well. In addition, I present research that can properly anonymize session logs with differential privacy as well. This thesis research serves as an important step towards a solution to the

challenge of query log anonymization with the balance of sufficient privacy guarantee and good utility in typical real-world IR applications.

The remainder of this dissertation thesis is organized as follows: Chapter 2 presents related work of this dissertation. Chapter 3 addresses the task of query log anonymization for single queries. I present my research on query log anonymization and how well the anonymized query logs can be used to support ad-hoc web search. Chapter 4 presents my research on session log anonymization and how well the anonymized log can be used to support complex IR tasks such as query suggestion and session search. Chapter 5 provides the proofs of differential privacy for the algorithms I present in Chapters 3 and 4. Finally, Chapter 6 concludes the dissertation with discussions.

Chapter 2

Related Work

This chapter presents the related work to this thesis. Chapter 2.1 introduces the basic ad-hoc search. Chapter 2.2 presents the related work for query suggestion. Chapter 2.3 shows the related work for session search. Chapter 2.4 summarizes the early techniques in privacy-preserving Information Retrieval (IR). Chapter 2.5 introduces the related work in differential privacy. Chapter 2.6 presents the existing work in query log anonymization. Chapter 2.7 is the chapter summary.

2.1 Ad-hoc Search

Ad-hoc search is a standard web search retrieval task [84]. In an ad-hoc search task, the user specifies an information need through a single query which initiates a search for documents which are likely to be relevant to the user's information need. The search engine retrieves documents from a corpus with certain retrieval algorithms and models. Finally, the retrieved objects will usually be evaluated based on their fulfillment of the user's information need. Following are some typical classic ad-hoc retrieval models.

Statistical modeling in Information Retrieval has largely been confined to solving static problems; examples include generating estimates for the probability of relevance using the BM25 formula [88], using Latent Dirichlet Allocation (LDA) to create topic models [115] or using link analysis to determine PageRank scores [84]. The data is used in the building and testing of the model and its parameters remain fixed for its application in a real-world scenario.

In the relevance feedback literature, classic techniques usually include the Rocchio algorithm [58], pseudo and implicit relevance feedback [12, 99] and its roots in modern interactive retrieval [90]. These are examples of dynamic IR systems that directly respond to user behavior. Likewise, we can model dynamic interaction over singular search queries by re-ranking [23] or optimizing over multiple pages of search rankings based on implicit and explicit feedback [57].

Since ad-hoc retrieval is the most simple web search process with one single search query, theoretically, any reasonable ranking (re-ranking) model for a corpus or a scoring system for the documents could be developed into an ad-hoc retrieval algorithm. As the most classic retrieval task in IR, ad-hoc retrieval is the foundation of the other retrieval tasks. For example, other IR tasks may involve multiple queries or social network data, but all of them were developed from this simple process of ad-hoc retrieval.

2.2 Query Suggestion

Query log anonymization for session data is an important part of my research which I will present in Chapter 4. Therefore, IR applications more complex than ad-hoc retrieval should be involved to examine the utility of the anonymized session log. I use query suggestion [45] as the major IR application to examine how the anonymized session log can be used to support the task. I present the related work for query suggestion here.

Although under specific settings, query suggestion can be done in the absence of query logs [8], most query suggestion approaches utilize query logs to suggest queries [3, 9, 13, 47, 64].

There are two main types of log-based query suggestion approaches. The first type clusters query based on a click-through bipartite graph. Wen et al. [116] analyzed query contents and click-through bipartite graphs from the query log. They generate similarities between two queries from the common documents the users selected for them and apply a density-based algorithm to form query clusters. Beeferman and Berger [6] proposed a hierarchical agglomerative clustering algorithm to find query clusters based on the bipartite graph. Similarly, Baeza-Yates et al. [3] proposed a method based on query clustering which semantically groups similar queries together. Feild and Allan [35] proposed a task-aware model for query recommendation using random walk over a term-query graph from query logs.

Another type employs query sequences to predict the following queries. For instance, Boldi et al. [9] utilized the concept of a query-flow graph to generate query suggestion results. The following queries are suggested according to the likelihood of query transitions from the previous query in the search session. Cao et al. [13] proposed an approach using both the click-through data and the session data. Unlike previous methods, this approach considers not only the last query but also several recent queries in the same session to provide better suggestions. Song and He [99] worked on optimal rare query suggestion also using random walk and implicit feedback in logs. Song et al. [100] proposed a query suggestion framework considering user re-query feedbacks from the query transition logs. This work modeled the term-level query reformulation activities through the query sequence to generate better suggestion results. Shinde and Joshi [94] gave a survey of various other recent query suggestion systems.

In summary, the most valuable information from query logs for query suggestion is the click-through data and query sequences in the search session. Both of them are available in my anonymized query log in chapter 4. Hence, I will use query suggestion as the major application to examine the utility of the anonymized log.

2.3 Session Search

Session search is another challenging IR task [16, 42, 43, 54, 55, 56, 59, 63, 67, 75, 78, 87, 117] that I use to examine the utility of the anonymized session log.

Session search has attracted a great amount of research from a variety of views. Generally, they can be grouped into log-based methods and content-based methods.

There is a large body of work using query logs to study queries and sessions. Wang et al. [113] utilized a semi-supervised clustering model based on latent structural Support Vector Machine (SVM) to extract cross-session search tasks. Our state-of-the-art research work modeled the dynamic process of session search as Markov Decision Processes (MDP) [43, 124, 130] and Partially Observable Markov Decision Processes (POMDP) [78, 131]. When a user keeps reformulating queries for a complex information need, the entire sequence of queries taken into account and the search results from queries in the session are aggregated to produce session-wide results. All these dynamic processes, especially the query sequences and click-through logs, are kept by the web search query logs. Many other log-based approaches also appear in the Web Search Click Data (WCSD) workshop series.[1]

Content-based methods directly study the content of the query and the document. For instance, Raman et al. [87] studied a particular case in session search where the search topics contain intrinsically diversified tasks, which typically require multi-session searches on different aspects of an information need. They applied techniques used in diversity Web search to session search. Content-based session search also includes most research generated from the recent TREC Session Tracks. Guan et al. [42] organized phrase structure in queries within a session to improve retrieval

[1]http://research.microsoft.com/en-us/um/people/nickcr/wscd2014/

effectiveness. Jiang et al. [56] proposed an adaptive browsing model that handles novelty in session search. Jiang and He [54] further analyzed the effects of past queries and click-through information on whole-session search effectiveness.

Others study even more complicated search – search across multiple sessions [67, 75, 113]. Kotov et al. [67] proposed methods for modeling and analyzing users' search behaviors that extend over multiple search sessions. Wang et al. [113] targeted the identification of cross-session (long-term) search tasks by investigating inter-query dependencies learned from user behaviors.

In summary, existing session search approaches share the common idea of identifying a session-wide information need from past queries in the session. Therefore, the query logs can be well used to support the task of session search.

I have published Session Search papers in SIGIR'13 [43, 130], TREC'13 [129], SIGIR'14 [78, 131], ECIR'15 [79] and TOIS [124].

2.4 Early Privacy-Preserving Information Retrieval Techniques

In 2006, an outbreak of privacy concerns was triggered when users were re-identified from a query log released by America Online (AOL) [4]. The AOL log was "anonymized" by hash coding the ID of each user, and it was soon proved how poor a mechanism it was. Since then, privacy-enhancing techniques [1, 21, 39, 61] have been tried on query logs. Early attempts include *Log Deletion, Hashing Queries,*

Identifier Deletion, Hashing Identifiers, Scrubbing Query Content, Deleting Infrequent Queries and *Shortening Sessions* [21].

The commonality shared by these techniques is that they tried to directly modify or remove individual data entries. However, there would always be some other traits left to re-identify the individual. By analogy you may still recognize the identity of a friend standing in front of you, even if he covered his name tag (hashing identifier) or wore a mask (scrubbing query content). Regarding these early techniques, now researchers have reached a consensus that they do not work [1, 39, 61]. Jones et al. [61] showed that by using a simple classifier, after removing unique terms from the query log, personal information can still be re-identified with an accuracy as high as 97.5%. Apparently, such naive privacy-preserving techniques are not good enough.

K-Anonymity [15, 105] has been a popular privacy protection technique since 2002. It achieves a certain level of privacy by blending the owner of released records into a crowd. The main idea is that any released record for an individual cannot be distinguished from at least k-1 other individuals whose information is also released. Sweeney [105] provides a detailed discussion about k-anonymity.

The k-anonymity mechanism doesn't limit the quantity of records each user may provide, which would result in more raw data being kept in an anonymized log. However, the privacy of k-anonymity is not strong because it depends on assumptions made about an adversary [105]. Years later, l-diversity [80] and t-closeness [72] were also proposed as new privacy mechanisms improved based on k-anonymity.

In order to develop data protection mechanisms with strong enough proved privacy guarantee in varying scenarios, our community started to get together and discuss the future of privacy-preserving IR. In the past few years, we organized the Privacy-Preserving IR workshops in SIGIR 2014 (PIR 2014 [96]), SIGIR 2015 (PIR 2015 [122]), and SIGIR 2016 (PIR 2016 [125]). This series of workshops focused on exploring and understanding the privacy and security risks in information retrieval. Talks and presentations in the workshop aimed at connecting the disciplines of IR, privacy, and security.

2.5 Differential Privacy

Differential privacy [33] is the state-of-the-art privacy protection concept [26, 28, 29, 30, 31, 32, 40, 66, 71, 83, 118, 119, 123, 134, 135]. It offers the strongest privacy guarantee for statistical data release. The key idea of differential privacy is to make the released statistics affected very limited by adding or removing any single record or any single user. Whether record level DP or user-level DP is achieved depends on the neighboring dataset definition. In this thesis, we work on user-level DP.

Recent work [40, 66, 134] has been proposed to use histogram based DP algorithms for query log anonymization. Gotz et al. [40] and Korolova et al. [66] proposed to release queries and clicks based on their frequencies. They are one of the first to release queries as natural language words instead of hash codes. Their work achieved (ϵ, δ)-differential privacy. They showed that more than 10% of the search volume and

0.75% of distinct queries from the original log could be privately released. However, in the existing work, the utility of the released query log was evaluated by the quantity of queries that could be released and how similar the released statistics are to the original statistics, instead of how these anonymized query logs could be really used to support research applications such as IR applications. The research on query log anonymization has just started and has much more to be complete.

I would also like to briefly mention the use of differential privacy in a few domains closely related to IR, which includes domains such as Text Mining [37, 120], Data Mining (DM) [37, 120] and Natural Language Processing (NLP) [14, 20, 37, 70, 93, 120]. Typical such differentially private approaches include differential privacy in social network analysis [106], histogram publication for dynamic datasets [71], frequent graph pattern mining [93], privacy-preserving inference from geo-location data [92, 107], geographic IR [108, 109, 110] or the use of differential privacy in data mining in general [37, 120]. Tutorials about the use of differential privacy in such related fields include [22, 46, 76, 81, 91, 126].

Specifically, frequent sequence mining [7, 10, 18, 102, 121] in data mining (DM) is relevant to our work because it studies how to anonymize sequences of data. Its techniques can be grouped by types of sequences being released – consecutive subsequence mining [10, 18], unconstrained subsequence mining [121], and frequent itemset mining [102].

It is worth noting that these frequent sequence mining techniques cannot be directly applied to query log anonymization. The reason lies deeply in the difference between IR and DM/ Database(DB). IR handles free text data in natural language, which can be considered an infinite domain. On the contrary, DM and DB handle structured data, which is generated from a limited domain, also called a limited vocabulary. Even with less than hundreds or thousands of unique items, most frequent sequence mining approaches would function at a very high cost regarding computational complexity. If they are applied to an unlimited domain as in IR, where theoretically any word sequence could be a search query, these algorithms' computational costs would be too high to be applicable.

2.6 Query Log Anonymization

The query log anonymization task came on researchers' radar in 2006 when a user was identified from the released AOL search log [4]. For years, researchers have proposed many ad-hoc techniques to help preserve privacy in query logs. [1] and [39] proposed anonymizing query logs by removing unique queries and segmenting the search sessions. Jones et al. [60, 61] studied the application of simple classifiers for identifying gender, age, and location, which can largely reduce the size of user candidates for portions of the query log. They found that the re-identification approach remains very accurate even after removing unique terms from the query log. These works verified the need for more robust anonymization techniques for query log anonymization.

Though with limitations, k-anonymity [1, 15, 48] provides specific privacy guarantees and has been utilized to help in this query log anonymization task. [15] and [48] proposed methodologies to reduce the large-scale data losses and utility in query log anonymization. However, the privacy of k-anonymity is based on assumptions about the background knowledge of the adversary. An approach that does not require such strict assumptions would be preferred.

Differential privacy [28, 34, 36, 39, 66] is a promising option since it does not make assumptions about the adversary. Although none of the previous research has achieved a proven private query log anonymization scheme that can publish the query log in its original plain format, some have begun investigating differential privacy in this context. For instance, [66] proposed an algorithm that releases a query-click graph containing queries, clicked URLs with each query, and the corresponding counts. They gave an (ϵ, δ)-differential privacy approach which maintains some utility. However, as mentioned in their paper as a limitation, their framework cannot output queries that were not included in the original query log, which also means that they cannot achieve ϵ-differential privacy. In this work, we filled this important gap by 1) proposing a method that preserves more contextual information than previous methods, 2) proposing a utility function that is specific to the primary task of query logs (web search) and leads to a more comprehensive evaluation, and 3) achieving ϵ-differential privacy by incorporating the idea of an external query pool.

A closely related research topic to query log anonymization is publishing transaction or sequence data with differential privacy [19, 68]. Query log can be viewed as a set-valued or sequence dataset where each user's record corresponds to a set or sequence of query terms and URLs (items), and the goal is to publish the count of query terms or sequences (itemsets). While many algorithms have been proposed in the literature for frequent itemset mining and frequent sequence mining with DP, a fundamental difference is that these methods assume a finite domain for the items, begin with all items in the domain as candidate itemsets and compute their noisy count. In query logs, there is an infinite possible set of query terms. Hence, the previous work only achieves the weaker epsilon-delta DP by releasing the noisy count of a subset of query terms from the query log (as opposed to the entire domain of possible query terms).

2.7 Chapter Summary

This chapter reviews major related work to my dissertation. In the IR utility aspect, I introduce the major IR applications that are involved in this dissertation including ad-hoc search, query suggestion, and session search. In the privacy aspect, I go through the privacy work from the earlier privacy-preserving techniques, differential privacy to the task of query log anonymization. In the following chapters, I will present my detailed query log anonymization approaches.

Chapter 3

Query Log Anonymization for Single Queries

Query logs can be very useful for advancing web search research. Since these web query logs contain private, possibly sensitive data, they need to be effectively anonymized before they can be released for research use. In this chapter, I propose using a differential privacy framework called Safelog to generate anonymized query logs that contain sufficient contextual information to allow existing web search algorithms to use the data and attain meaningful results. I evaluate the effectiveness of my framework for generating usable, privacy-preserving logs for web search and demonstrate that it is possible to maintain high utility for this task while guaranteeing sufficient privacy.

This chapter mostly involves contents from my publications [133, 134, 135].

Web query logs have been used to guide the development of new retrieval methods [2, 27, 78, 131]. While not obvious, anonymization is more difficult for query logs than for other more structured data sets because query logs are generated from billions of individual users' natural language. The associated vocabulary domain for these queries is, therefore, infinite. This is a sharp contrast to the finite domains of more traditional data, e.g. itemset mining of a finite domain of items [19, 68].

In this chapter, I develop a novel ϵ-differential privacy framework called Safelog that sanitizes and anonymizes a query log. The generated query log maintains utility for web search and web mining algorithms while maintaining strong privacy guarantees. The key to achieving the strong privacy guarantees is the introduction of a *query pool* for augmenting the query log during the anonymization process. I explain and empirically show the privacy guarantee and how to measure the actual retrieval utility for the task of web search (the primary task that uses query logs). I also consider other web mining tasks that can be supported by these anonymized logs and show some preliminary results for a clustering task.

To summarize, the main contributions in this chapter are as follows:

(1) It is the first to evaluate the utility of differentially private anonymized query logs on the task of web search. I present Safelog, an effective framework for implementing and evaluating both the privacy and the utility of an anonymized query log. To better evaluate the effectiveness of the anonymized query log, I propose a new utility function that is tailored to this task.

(2) I demonstrate how a log anonymization algorithm achieves ϵ-differential privacy, improving the state of the art in this area from (ϵ, δ)-differential privacy [66].

(3) I present an empirical analysis that highlights the effectiveness of my framework for document retrieval on real world data. I also analyze the privacy-utility tradeoff so that companies can decide on the level of privacy that is acceptable to them. Based on both the theoretical and empirical findings, I make practical recom-

mendations for companies interested in releasing anonymized query logs that include a detailed discussion of how to set parameters.

3.1 Preliminaries of Differential Privacy

Table 3.1 uses a toy example to explain how DP works. Suppose we would like to release a dataset about users' possession of apples. We would like to release a summary statistic of the dataset to the public. Would there be any privacy concern when we release the sum as a statistic? The answer is yes. Suppose we release Dataset 1 or Dataset 2. Dataset 1 and Dataset 2 are two datasets that differ by exactly one user, Carol. The summary statistic, the sum of apples, for the two datasets also differs by what is contributed by Carol (11-9=2). If an adversary happens to know that (a) Carol has 2 apples and (b) everybody else has either 4 or 5 apples, then it is easy to identify that Carol is in Dataset 1 and not in Dataset 2. The re-identification can be done by calculating the possible decomposition for the released sum. For 11 the only possible decomposition is 5+4+2 and for 9 it is 5+4. Hence, Carol must be included in Dataset 1 and excluded from Dataset 2. The adversary can thus identify which dataset Carol is in. If more information about Carol is stored in the dataset, the adversary could possibly find it out once knowing where Carol is. To obscure an individual's identity, DP adds randomized noise to the summary statistic. The anonymized data then follows a distribution whose mean is equal to the original mean, but the end results (10) would not be distinguishable between datasets with or without Carol. In

Table 3.1: Toy example for Differential Privacy. Given only the anonymized data (10 total apples), it is difficult to tell whether the raw data includes Carol (Dataset 1) or excludes Carol (Dataset 2).

	Dataset 1	Dataset 2
Raw Data	Alice has 5 apples Bob has 4 apples Carol has 2 apples	Alice has 5 apples Bob has 4 apples
Sum of apples	5+4+2=11	5+4=9
Anonymized Sum	11+Noise=10	9+Noise =10

other words, with DP, the adversary would not be able to figure out whether or not Carol is in a given dataset based on the released anonymized statistics.

Neighboring is a concept upon which differential privacy is defined. Therefore, we start the definition for differential privacy from defining what neighboring datasets are.

Definition 1: Neighboring. Two query logs, or more generally two datasets, Q_1 and Q_2, are said to be neighboring to each other if they differ by at most one user.

The toy example we presented in Table 3.1 shows two neighboring datasets. The only difference between them is information from one person, Carol. When we define differential privacy later, any possible pairs of neighboring datasets are under consideration. It means the datasets could also differ by Alice or by Bob.

Definition 2: Differential Privacy. A randomized query log anonymization algorithm A satisfies differential privacy, or more specifically (ϵ, δ)-differential privacy,

iff. for all neighboring query logs Q_1 and Q_2, and for all possible output Q', the following inequality holds:

$$Pr[A(Q_1) = Q'] \leq e^{\epsilon} \times Pr[A(Q_2) = Q'] + \delta \tag{3.1}$$

where A is the randomized anonymization algorithm that takes an original query log as the input and outputs an anonymized query log Q'. ϵ and δ indicate privacy levels of A and control the probabilities of getting certain outputs from neighboring inputs. The ranges of them are $0 \leq \epsilon \leq \infty$ and $0 \leq \delta \leq 1$.

In Eq. 3.1, smaller values of ϵ and δ would lead to stronger privacy guarantee. Therefore, we usually expect ϵ and δ to be much smaller than their upper bounds. In general, there is no hard rule for selecting values of ϵ and δ. Proper settings of them vary depending on actual applications. However, usually we consider an ϵ no more than 10 and a δ less than or around (1/#of released users) to be acceptable.

According to Eq. 3.1, the values of ϵ and δ directly affect how different two neighboring query logs are after applying the anonymization algorithm A. Therefore it is necessary to quantitively define the difference between two neighboring input query logs Q_1 and Q_2. This is the concept of *sensitivity*:

Definition 3: Sensitivity. Given a function f that takes a query log Q as the input and a numeric vector as the output, the sensitivity of the function is denoted as Δf:

$$\Delta f = \max_{\forall \text{ neighboring } Q_1, Q_2} ||f(Q_1) - f(Q_2)||_1 \tag{3.2}$$

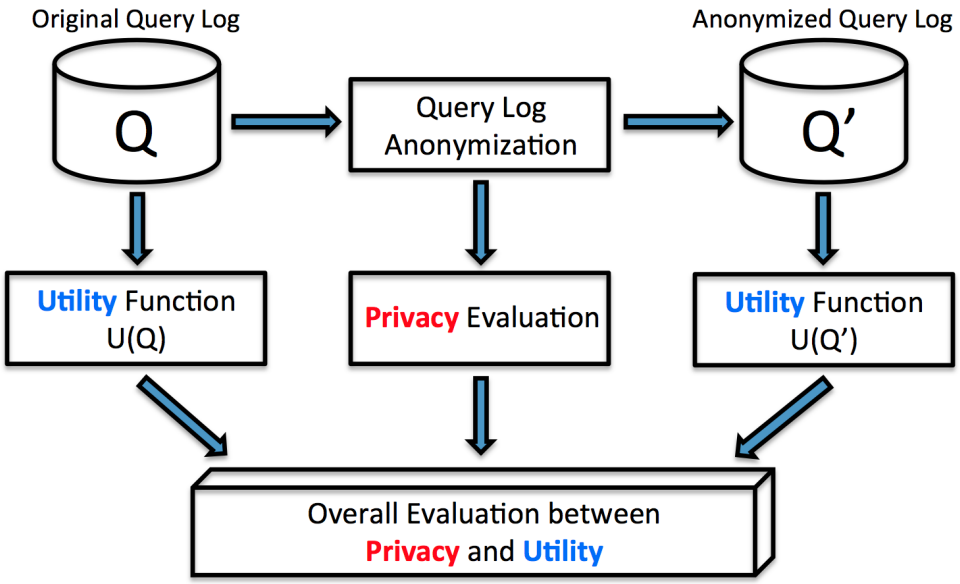


Figure 3.1: General framework of query log anonymization.

where $||.||_1$ is the l_1 norm. The maximum is taken over all pairs of neighboring query logs Q_1 and Q_2. In the scenario of query log anonymization, the numeric vector output of function $f(Q)$ is the vector of raw count statistics generated from query log Q. The sensitivity value Δf shows the maximum overall statistical difference between the two neighboring inputs Q_1 and Q_2, which largely influence the values of ϵ and δ.

3.2 Problem Formulation

This section presents my formulation of the query log anonymization problem for single queries. Figure 3.1 presents a general framework of query log anonymization along with how we evaluate it.

Query Log Q: Query log Q is a textual document that records query data between the search engine and its users. Usually, it contains a record for each user including the user's ID, the query, a ranked list of URLs that the search engine returns to the user, click-through information, and timestamps for all user actions.

The Task of Query Log Anonymization: Given an input query log Q, the task is to produce a version of the log in which the identifiable data is removed and the remaining data is adequately anonymized so as to reduce the likelihood of re-identification of users. The output of this task is an anonymized query log Q', with a guaranteed degree of privacy.

Privacy Function A: An anonymized query log Q' is generated by applying a privacy function A on the original query log Q. That is, $Q' = A(Q)$. Usually, A is parameterized to indicate the level of privacy that Q' can achieve. For example, in differential privacy, ϵ and δ is the parameter in A, i.e., $Q' = A(\epsilon, \delta, Q)$. Smaller ϵ and δ values indicate higher levels of privacy protection. In the scenario of query log anonymization, we may also refer to the privacy function A as the query log anonymization "algorithm" or the query log anonymization "mechanism".

Utility Function U: In privacy-related research, the remaining utility of the data after applying a privacy function on it is an indispensable part of the research. Usually, a utility function U needs to be domain specific to be able to evaluate the usefulness of the data in a domain. The utility function can be applied on both the original data $U(Q)$ and the anonymized data $U(Q')$ to compare the utility deduction.

Web Search Using Query Logs: Given a query q and a query log Q, the task of web search is to provide a ranked list of documents or URLs D that is relevant to q, from a set of documents or URLs that are built into a pre-indexed corpus C. Most Web search algorithms fit into this setting. User clicks, query reformulations, time spent examining the returned documents, and clicked documents on similar queries shared by multiple users are often the key elements used in a modern Web search algorithm.

Utility Function for Web Search: In the context of information retrieval, a utility function U could be a two-step process – the first is to use the query log for document retrieval, i.e., to retrieve a set of ranked documents D for any $q \in Q$, where $D = R(q), q \in D$ and R is a retrieval algorithm. The second is to use IR evaluation metrics E to measure how good the retrieved document list D is with respect to each q being evaluated; that is $E : E(D)$. Therefore, the utility function of a query log Q can be represented as $U(Q) = E(R(Q))$, where E is a retrieval effectiveness measure for search results generated by R.

Goal: The goal of a successful query log anonymization algorithm is to have

$$|U(Q) - U(Q')| < \sigma \tag{3.3}$$

where σ is kept small. At the same time, a successful query log anonymization algorithm should ensure that the privacy level ϵ and δ: $Q' = A(\epsilon, \delta, Q)$ are small enough to provide high privacy guarantee.

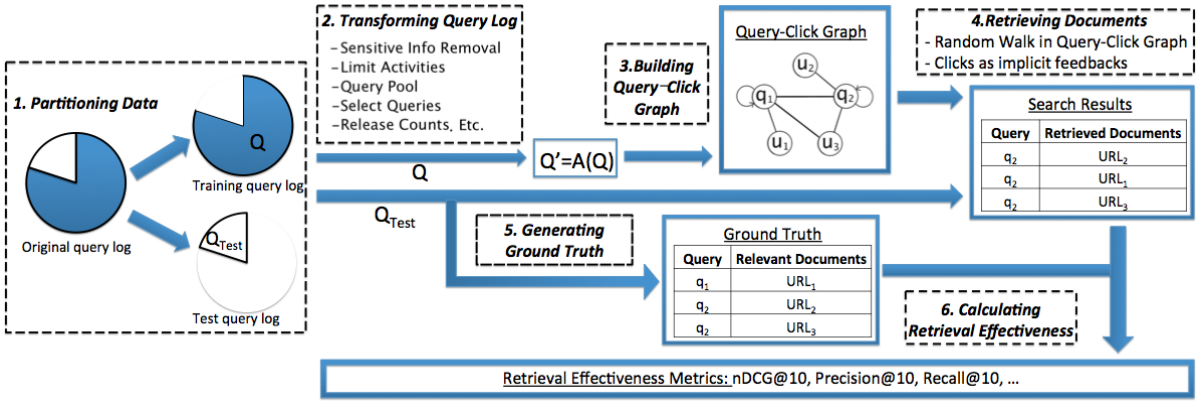


Figure 3.2: Framework overview: My approach.

3.3 Anonymization Algorithm for Single Queries

Figure 3.2 shows my framework for query log anonymization. The Figure focuses on a workflow of creating anonymized query logs as well as measuring the logs' utility in a complete pipeline for the task of web search. I first partition data using 5-fold cross validation. In each run, I use 80% of the data as the training set Q and the remaining as the test set Q_{Test}. Q acts as the raw query log from the search engine that is the input to the query log anonymization algorithm. Q_{Test} is used to evaluate the utility of my approach on the document retrieval task. Then I transform the query log Q into $Q' = A(Q)$ in a privacy-preserving way. After that I build a query-click graph, where nodes are queries and URLs (documents) while edges connect query nodes with their clicked URL nodes.

Algorithm 1 A_{Click}: Query Log Anonymization Algorithm

1: **Input.** Q: Original query log collected by search engine; Q_p: Set of external search queries; τ :query filtering parameter; q_f, c_f: limiting activity parameters for number of queries per user and number of clicked urls per user; b, b_q, b_c: noise parameters; K: threshold of tail.
2: **Output.** Q'.
3: Q_{clean} = removeSensitiveData(Q, τ)
4: Q_{clean} = limitUserActivity(Q_{clean},q_f, c_f)
5: $Q^+ = Q_{clean} + Q_p$, considering queries from Q_p as queries with 0 frequency as in Q_{clean}.
6: $Q_{reduced}$ = selectFinalQuerySet(Q^+,b, K)
7: Q' = generateLogStats($Q_{reduced}$,b_q, b_c, K)
8: return Q'

Algorithm A_{Click} shows my high-level algorithm for query log anonymization. The remainder of this subsection describes the main components of the proposed algorithm.

1) The key input of the algorithm includes the original query log Q. Other input of the algorithm includes: the set of external search queries Q_p, the query filtering parameter τ, the parameters for max number of queries per user q_f and max number of clicked URLs per user c_f, noise parameters b, b_q, b_c, and frequency threshold parameter K.

2) The output of the algorithm is an anonymized query log Q'. The output contains click-through data as shown in Table 3.2.

Table 3.2: Anonymized AOL query log: Click-through data.

Query	Clicked URL	Counts
weather	http://www.weather.com	4190
weather	http://weather.yahoo.com	1035
aol weather	http://weather.aol.com	30
aol weather	http://aolsvc.weather.aol.com	16
blue book	http://www.kbb.com	33
blue book	http://www.nadaguides.com	1
hairstyles	http://www.hairfinder.com	5
hairstyles	http://www.1001-hairstyles.com	19
hairstyles	http://www.hair-styles.org	21
hairstyles	http://hairstyles.free-beauty-tips.com	16
...	...	...

3) Preprocessing. As a preprocessing step, step 3 of the algorithm first empirically remove all queries with a frequency less than 5 from the corpus in order to prevent the release of unique sensitive data. This step also removes typos. I refer to the output of this step as Q_{clean}.

It is worth noting that step 3 is acting as a "double insurance" of the privacy guarantee. In fact, the DP does not require such a preprocessing step. Since there may be specific privacy concerns on the unique queries with very low frequency, which are more likely to contain personal sensitive data such as one's Social Security Number (SSN), I add this preprocessing step to make sure that unique information will not be released from the algorithm. In addition to the removal of very low frequency data, this preprocessing sanitation can be further enhanced by removing data containing certain named entities from sensitive domains. Practically, I add another preprocessing step to remove all search queries containing email addresses, phone numbers,

Pattern-1: Someone elses bday w/ mention	$\{happy\|Happy\|HAPPY\}\{birthday\|Birthday\|BIRTHDAY\}@SOMEONE$
Pattern-2: Someone elses bday in retweet w/ mention	$\{@SOMEONE\}^n * \{happy\|Happy\|HAPPY\}$ $\{birthday\|Birthday\|BIRTHDAY\} * \{@SOMEONE\}^m$
Pattern-3: Person's own birthday	my birthday is $\{in\|on\|\epsilon\}[TimeExpression]$

Figure 3.3: Sensitive information removal example. Lexicon-syntactic patterns for BIRTHDAY on Tweets.

and SSN numbers. However, the anonymized query logs are not influenced by this second preprocessing step since there are 0 occurrences of such sensitive data from these domains in this AOL dataset. Therefore, I only keep the query frequency based preprocessing step to make the algorithm neat.

Actually, removing sensitive attribute information from text data is another research topic. In one of our recent publications [97], I proposed a pattern-based attribute detection algorithm to detect information in certain attribute topics such as birthday or location from a tweet dataset. Figure 3.3 presents some lexicon-syntactic patterns for birthday attributes on tweets, while Table 3.3 shows the coverage and precision of such pattern-based extraction on the tweet dataset. Practically, the owner of the query logs may choose to remove certain information from the query log in its own way as an extra part of the preprocessing step to reduce the privacy risk even more. Since this is not the focus of this thesis, I do not go into much detail here.

4) Limiting User Activity. In this step, I reduce each user's sample in the query log by limiting the number of queries and URL clicks of each user. Specifically, the algorithm only keeps the first q_f queries and the first c_f URL clicks of each user from

Table 3.3: Sensitive information removal example. Coverage and precision of pattern-based extraction on tweet dataset.

Attribute	**Pattern Interpretation**	**# of Posts w Pattern**	**Precision**
Birthday	Pattern 1	36	35/36=97%
Birthday	Pattern 2	207	84/100=84%
Birthday	Pattern 3	36	33/36=92%
Brand	Concern/interest in brand	11,095	87/100=87%
Sports team	Interest in sport's team	572	99/100=99%
Location	Visited location	34,296	51/100=51%

Q and removes the rest. Intuitively, this step allows us to guarantee that the removal or addition of a single individual in Q has a limited effect on the query log. I will give an experiment later about the values of q_f and c_f.

5) Query Log Expansion. This step is based on an assumption of the use of a query pool, which I will give a detailed discussion about it later. In order to overcome the challenge of the infinite domain in query logs, the key idea of step 5 is to use an external stochastic query pool to augment the query terms already in the query log. In other words, the query term domain can be viewed as a sampled set of terms S from the set of all possible query terms in the population P. I will show formally that using an external stochastic query pool Q_p to augment the query log with additional queries improves the overall privacy and allows us to achieve pure DP. I refer to the expanded set of queries as Q^+, where $Q^+ = Q_{clean} + Q_p$. In the next section, I will discuss different query pool generation strategies.

6) Selecting Final Query Set to Release. After that, I select the final set of queries to release in step 6. Using $Lap(b)$ to represent a random real value drawn independently from the Laplace distribution with mean 0 and scale parameter b [33], I define perturbed counts to be query counts after applying Laplacian noise. I choose to release a query q when its perturbed query count $(M(q, Q^+) + Lap(b))$ is greater than a threshold K, where $M(q, Q^+)$ is the frequency of query q in Q^+. Specifically, for each query q added from the query pool, $M(q, Q^+) = 0$. However, its perturbed query count $(M(q, Q^+) + Lap(b))$ still gets a chance to pass the threshold K and therefore be included in the output of this step. Theoretically, since every query on Q^+ has a chance of being selected in the final log, the algorithm can achieve ϵ-differential privacy. The final query set generated after this step is referred to as $Q_{reduced}$.

7) Releasing Click-through Data. As previously mentioned, I release the perturbed query counts $(M(q, Q+) + Lap(b_q))$ for each query. It is worth noting that for the perturbed query counts, the algorithm adds noise again using another parameter b_q in step 7. Although b_q does not necessarily differ from b, this process reduces the impact of the cut-off threshold K from the previous step. I also release the perturbed click counts for each URL: <q, u, #u was clicked when q was posted + $Lap(b_c)$>.

3.3.1 The anonymized query log

Let K, q_f, c_f, b, b_q and b_c be parameters in my algorithm as defined previously. Let Q be the original query log as input to the algorithm and Q_{clean} be the set of queries from

Q that are possible options for release because they occur often enough while keeping at most q_f queries and c_f clicks from each user. Let Q_p be an externally generated stochastic query pool containing a large set of queries. Suppose each possible query q in the infinite domain has a probability of $p_g \in [0,1]$ to be included in the pool Q_p. In practice, the value of p_g depends on the source that is used to generate the query pool. While I provide an approach for generating Q_p, major commercial search engines that have access to a large number of historic queries in their system can create a large pool Q_p satisfying a p_g value close to 1. Here I state the following theorem:

Theorem 1: The query log anonymization algorithm presented in Algorithm A_{Click} satisfies ϵ-differential privacy, where ϵ is defined as:

$$\begin{aligned} \alpha &= Max\{\frac{e^{1/b}}{p_g}, 1 + \frac{1}{2e^{(K-1)/b} - 1}\} \\ \epsilon &= q_f \cdot ln(\alpha) + q_f/b_q + c_f/b_c \end{aligned} \tag{3.4}$$

I will give the proof in chapter 5.

As I have presented, the query log anonymization algorithm introduced an external stochastic query pool in order to overcome the challenge of the infinite domain in query logs. The proper use of such an external query pool is based on an assumption that the involvement of the query pool will not raise an extra privacy issue. Here I give interpretations about the solution with such further assumptions in section 3.3.2 and solution without such further assumptions in section 3.3.3.

3.3.2 A solution with further assumptions

Having the assumption that the query pool will not raise an extra privacy issue, I give the following detail about how we make use of the query pool.

Formally, I define Q_p as an external query pool generated using an external set of search queries that are independent of the queries in the original query log Q. Q_p serves as a proxy for the full set of queries that exist in the population P. Each query in Q_p has an equal probability of being included in the query pool. When a commercial search engine uses my algorithm, this query pool Q_p can be generated using a random sample of all their recorded queries or by using queries from a different period. If the previous set of recorded queries is insufficient to represent P, query terms can be randomly extracted from a random set of web pages. Then, I can expand the query log as $Q^+ = Q_{clean} + Q_p$. Queries added to Q^+ from Q_p are queries with a click count of 0 in the original query log.

Because academic researchers do not have access to an extensive query set like commercial companies, I must have an approach for simulating the query pool construction process. Therefore, I propose a simulation algorithm that generates a query pool using artificial queries constructed by randomly sampling and combining high-frequency n-grams present in the English language. In my experiments, I use the Corpus of Contemporary American English (COCA)[1], which includes approximately 450 million words and 190,000 texts. Using this corpus, Davies [25] published the

[1]http://corpus.byu.edu/coca/

(approximately) 1 million most frequent n-grams each for n=2, 3, 4 and 5. I identified 1,159,938 n-grams from this list that end with a noun since nouns are more likely to be part of search engine queries. I sample these n-grams to generate the final query pool Q_p. In other words, I combine the query terms from two independent samples, making it difficult for an adversary to know clearly which queries are real and which ones are not. Using a query pool to maintain log privacy is one of the main contributions of this work. I will show in my empirical evaluation that even with the addition of these noisy, external data, I can still maintain reasonable utility for web search queries.

3.3.3 Solutions without further assumptions

The use of the query pool I just introduced is based on the assumption that the query pool itself will not raise other privacy issues. However, if the query pool is sampled from another private collection of search queries, the situation will be different. Improper use of the query pool may lead to a privacy leak of the private dataset that the query pool is sampled from. Therefore, the privacy risk of exposing that private collection of search queries should be carefully analyzed.

In this section, I propose two potential alternate solutions to address the challenge when such a query pool of all search queries is not publicly available.

Solution 1: Generate a publicly available superset of the query pool with Natural Language Generation (NLG) [5]. An alternate solution is to replace the

query pool by its superset generated from a dictionary or a public collection of search queries. If we generate the query pool from a collection of search queries that are publicly available, there won't be new privacy issues from the use of the query pool. Alternately, if we randomly generate a query pool from a dictionary, although there will not be privacy issues raised from the query pool, the effectiveness of the query pool may decrease. According to the sparsity of natural language, we may have to generate an exponentially increasing amount of term combinations in order to cover most of the potential queries. In other words, the noise scale may be too large.

Although the exact dataset (query pool Q_p) of all recorded queries from the search engine may be considered sensitive data by the search engine company, it is still possible to generate a superset Q'_p of it and make it publicly available. Specifically, Q'_p is a disordered superset of Q_p containing both original search queries as well as artificially generated queries [5]. The major difference between this NLG-based Q'_p and an artificial query pool purely generated from a dictionary is that Q'_p is a finite superset of all recorded queries. Although most commercial search engine companies are currently hesitating to release their data for different reasons, the release of such Q'_p is possible in the near future since it is only a disturbed collection of queries.

By using such an NLG-based superset Q'_p of the query pool Q_p, my algorithm A_{Click} can be implemented in the exact same way. Practically, as a tradeoff of involving more artificial queries, the privacy level of the algorithm may be reduced with a greater

ϵ value. In summary, the challenge may be resolved if any non-trivial supersets of web search queries become publicly available in the near future.

Solution 2: Map the infinite domain into a finite domain. As I have mentioned, a key challenge of query log anonymization as well as sequential anonymization in the IR domain is how to address the sparsity of natural language distributions. Theoretically, any combination of words may generate a legal query, while any combination of queries may generate a query sequence as a search session. Although there has been some differentially private work in related fields such as Data Mining [37], consecutive subsequence mining [10, 18], unconstrained subsequence mining [121], and frequent itemset mining [102], they can not be directly implemented in IR since their high computational complexity is not affordable in the IR scenario with more than millions of different search queries.

However, a potential solution of using those privacy-preserving Data Mining algorithms in IR may be possible if we can properly map the infinite domain of search queries into a finite domain. This may be especially helpful to session-based IR tasks that require sequential information of the search sessions.

To be specific, not all IR applications require raw data of the original search queries during the search process. For instance, as a toy example, if my IR task is to research the user patterns of generating "long" queries during a search session, the exact content of the search queries may not be necessary in this research. Actually, one binary bit L is enough to provide the required information for a search query: L

takes 1 when the query is "long", while it takes 0 when the query is not "long". Hence, a search session with 6 queries may be simply represented as $(1, 0, 1, 1, 0, 0)$. By simply adding a limitation that a search session can consider no more than 20 queries, we can map any search session from a domain space of ∞^{20} into a domain space of 2^{20}. Practically, many of the mechanisms such as model-based prefix tree mining [10] with higher computational complexity may became affordable in this domain scale.

Generally, how to map the infinite domain of queries into a finite domain is task dependent. For instance, a topic classifier may map a search query as one of the predefined topics, which may be used to support the IR research on topic drifts in session search [78]. Another example is that a mapping from a search query to a concept in a knowledge graph or ontology may be used to support multiple tasks in NLP and IR research. The major advantage of this mapping-based solution is that many of the existing differentially private Data Mining approaches may be applied to address IR tasks, which greatly expands the arsenal of methodologies available to privacy-preserving IR researchers. However, the major limitation of it is that the anonymized dataset can only be used to support very specific IR applications since the raw content of queries has been compromised.

In summary, although this solution is not universal to support all IR applications, it is still meaningful when the IR application only requires certain information in a finite domain. I hope this can inspire more follow up work on this path of privacy-preserving Information Retrieval.

I measure the utility of the anonymized logs by using them to help the task of ad-hoc retrieval in web search.

Retrieving Documents. We retrieve documents using three different algorithms [2, 24, 111] for queries in Q_{Test} in order to evaluate the utility of the released query log Q'.

The first retrieval algorithm is based on a random walk on the query-click graph. In the graph, nodes are queries and documents (URLs), while the transition weights between nodes are defined by their relationship in the released data. The most common transition type links a query node to a document node. Another type of transition we consider is between two query nodes in the query-click graph. Each query node also has a transition weight to itself as a self-loop. I calculate the transition probability $P(k|j)$ from a query node j to document node or another query node k in a slightly different way from a popular random walk click model proposed by Craswell and Szummer [24]. Then, I rank the URLs according to the descending order of the probabilities of staying at corresponding URL nodes. $P(k|j)$ is calculated by:

$$P(k|j) = \begin{cases} (1-s)C_{jk}/\Sigma_i C_{ji} & , \forall k \neq j \\ s & , k = j \end{cases} \tag{3.5}$$

where C_{jk} is the weight between nodes j and k given by Q', and s is the self-transition probability. If both nodes j and k are query nodes, weight C_{jk} is defined as the query transition counts from j to k as specified in Q'; otherwise, if j is a query node while k is a document node, weight C_{jk} is defined as the click-through counts for this query-document pair as specified in Q'. In this approach, I empirically set the self-loop probability $s = 0.1$. After that, I can rank documents in descending order by the probability of being the stopping node.

The second algorithm uses the impact factor of each web page to enhance the random walk model. It is based on [111]. In this setting, more popular websites gain greater probabilities for the random walker to walk into. An impact factor F for each web page is introduced to the above random walk model, resulting in greater probabilities for the walker to walk into nodes with greater F values. In the implementation, I define the impact factor F_i of a web page (document node i) as a smoothed sum of its click counts as

$$F_i = 1 + \log(1 + \Sigma_j C_{ji}) \tag{3.6}$$

where C_{ji} is the weight between node j and i given by Q', and the impact factors for the other query nodes are set to be a constant 1. The new transition probability $P'(k|j)$ from node j to node k can be calculated by

$$P'(k|j) = Normalized(\frac{F_k}{\sum\limits_{i \in L_j^+} F_i} \times P(k|j)) \tag{3.7}$$

, where L_j^+ is the set of outbound links of node j and $P(k|j)$ and F are calculated as earlier.

Besides using query graphs for document retrieval, we can also consider using user clicks as feedback. This additional query log data is useful for some web search algorithms. The third algorithm is a modified version of the implicit feedback model proposed in [2]. It merges the original rankings with the implicit feedback, in this case the user clicks. Previous literature has proposed methods that incorporate user behavior data to help to improve the order of retrieved documents. In this work, I implement a variant of [2].

Given a query q, the relevance score $S(d)$ for each document d is calculated as:

$$S(d) = \begin{cases} \lambda \frac{1}{I_d+1} + (1-\lambda)\frac{1}{O_d+1}, & \text{if implicit feedback exists for document d} \\ \frac{1}{O_d+1}, & \text{otherwise} \end{cases} \tag{3.8}$$

where O_d represents the original rank of document d, I_d represents the implicit feedback rank in Q_{test}, and λ is a parameter to weigh the importance of the implicit feedback. In this approach, the original rank O_d is ranked by the order of click-through counts of document d for query q, according to Q'. I_d is the rank of d from Q_{Test} when the user makes a click. I empirically set $\lambda = 0.6$. Finally, the documents are ranked in the descending order of $S(d)$ scores for each query q. In addition, I generate a ground truth set of documents based on the actual clicking information in Q_{Test} to evaluate the document retrieval results obtained from the previous step. For each tested query in Q_{Test}, the corresponding ground truth contains a set of relevant

documents (URLs), where relevant means that they have been clicked on in Q_{Test}. Actual search engines may choose to replace my approach in this step since they have more detailed data about users' online activity (for instance, the dwell time on each returned page) than I do.

Calculating Retrieval Effectiveness. I compare my retrieval results to the ground truth and evaluate the retrieval effectiveness using multiple IR metrics, including nDCG (normalized discounted cumulated gain) [53], MAP (Mean Average Precision) [89], Precision, and Recall. Among them, nDCG at rank position 10 (nDCG@10) is the most widely used IR evaluation metric for web search in both commercial companies and academia. It measures the retrieval effectiveness for a ranked list of retrieved documents in the first ten results, which form a SERP that a searcher cares most about. I also use nDCG@10 as the predominant evaluation metric. For each query q in a query log Q, I can calculate its nDCG@10 as:

$$nDCG@10(q, D) = \{\sum_{i=1}^{10} \frac{rel_i}{log_2(i+1)}\}/\{\sum_{i=1}^{N_{Rel}} \frac{1}{log_2(i+1)}\} \tag{3.9}$$

where D is the ranked list of documents retrieved for query q by ranking algorithm R such that $D = R(q)$. rel_i is 1 when the i^{th} retrieved document $d_i \in D$ is relevant. Otherwise, it is 0. N_{Rel} is the smaller value of the total number of relevant documents for q and 10.

This work is the first to evaluate the utility of a query log using actual IR evaluation metrics. The total utility function $U(Q)$ for a query log Q is:

$$U(Q) = \frac{1}{|Q|} \sum_{q \in Q} nDCG@10(q, D) \tag{3.10}$$

Note that not all queries in Q_{Test} can be found in the anonymized query log Q'. Most of the added queries in Q' from the query pool may not be included in Q_{Test} either. Therefore, when this utility function is used on the anonymized query log Q' to test on retrieval on Q_{Test}, only those queries in the intersection of Q_{Test} and Q' can be evaluated.

Table 3.4: Statistics of the AOL query log.

Statistics	Counts
Total number of records	36,389,567
Log size (GB)	2.2
# of unique user IDs	657,426
# of unique queries	10,154,742
# of clicks	19,442,629
Avg. clicks per user	29.57

3.5 Experiments

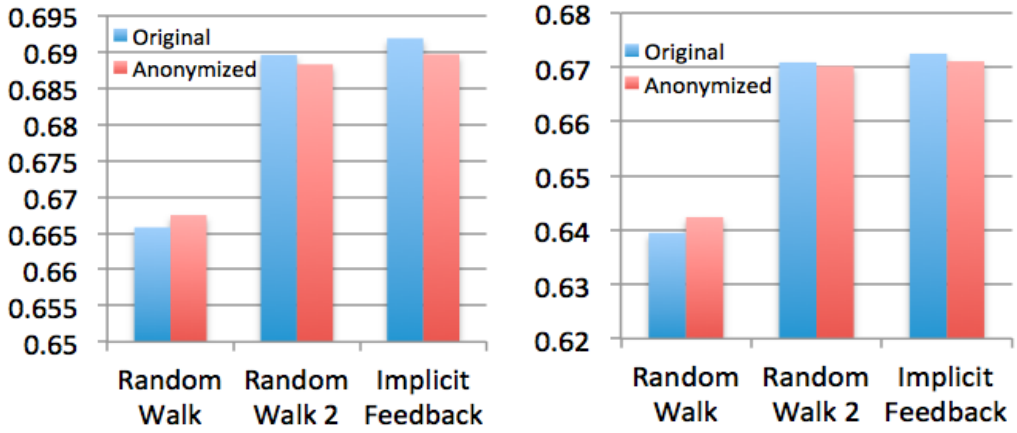
In this section, I evaluate the privacy-utility tradeoff of the query log anonymization algorithm based on the AOL query log dataset. I use retrieval effectiveness for web search to evaluate the utility of anonymized query logs. This section presents my empirical results and analysis.

3.5.1 Experimental setup

I use the released AOL query log dataset [1] in my experiments. The dataset is a query log containing 36,389,567 search records. Table 3.4 presents some more statistics of the AOL query log dataset. At this stage, the AOL query log is the only available query log for privacy-related research like ours. Table 1.1 gives a sample of the original AOL query log.

As detailed in the previous section, I first partition the input query log into a training set Q and a test set Q_{Test}. Then I use my query log anonymization algorithm to generate the anonymized query log Q'. After that, I use the presented document retrieval algorithms to retrieve documents for queries in Q_{Test}. Finally, the utilities are calculated by comparing search results against the ground truth. During parameter tuning of the query log anonymization algorithm, I ran different combinations of the major parameters, including K, b, q_f, and c_f. To better control and compare the major parameters, I set constant values for some other parameters such as b_c and b_q.

In the following section, I present my detailed experiments. Firstly, I evaluate the utility by retrieval effectiveness to verify that the anonymized query log can be as useful as the original query log. Then, I design and implement experiments to investigate the impact of major parameters K, b, q_f and c_f on privacy and utility. Furthermore, I implement experiments to research the privacy-utility trade-off. Finally, I give overall analysis and generate more experiments to give my recommendations of parameter selection during the query log anonymization process.



(a) nDCG@10 (b) MAP

Figure 3.4: Utility by retrieval effectiveness.

3.5.2 Utility by retrieval effectiveness

First of all, we need to show that the anonymized query log is useful. Specifically, I implement this experiment to illustrate that the utility of the anonymized query log Q' can be comparable to the utility of the original query log Q according to their retrieval effectiveness. I use multiple retrieval algorithms to do document retrieval based on Q and Q' and compare their utility scores. Figure 4.3 shows the utility across different retrieval evaluation metrics including nDCG@10 and MAP (Mean Average Precision). For each of the three algorithms I presented in the previous section, I run them on both the original log and an anonymized log with the following privacy settings: $\epsilon = 29.99$, query frequency threshold $K = 500$, and noise scale $b = 10$.

Table 3.5: Utility by retrieval effectiveness with random walk. Two-tailed t-tests ($p < 0.01$) show no significant difference of utility scores before and after anonymization.

Query Log	nDCG@10	P@5	P@10	MAP
Original	0.6658	0.1484	0.0779	0.6395
Anonymized	0.6675	0.1486	0.0777	0.6424

Table 3.6: Utility by retrieval effectiveness with implicit feedback. Two-tailed t-tests ($p < 0.01$) show no significant difference of utility scores before and after anonymization.

Query Log	nDCG@10	P@5	P@10	MAP
Original	0.6919	0.1535	0.0796	0.6725
Anonymized	0.6897	0.1527	0.0790	0.6711

Results in Figure 4.3 indicate that my anonymized query log can produce comparable query effectiveness results to those of the un-anonymized version. Table 3.5 and Table 3.6 present more detailed utility results for the random walk run and the implicit feedback run. Two-tailed t-tests ($p < 0.01$) show that there is no significant difference of utility scores before and after anonymization. In other words, under certain circumstances, Q' can perform as well as the original non-private query log. This occurs when the noise scale b is much smaller than the query count threshold K. This means that the statistics in the released query logs are not influenced significantly by the added noise. These results confirm the utility level of the anonymized log generated by my framework can be comparable to the utility level of the original log, which is the foundation of further experiments.

Table 3.7: ϵ-DP and (ϵ, δ)-DP achieve similar utility scores with different privacy guarantees.

Anonymization Algorithm	ϵ	δ	nDCG@10	MAP
ϵ-DP	1.40	0.0000	0.5239	0.5127
(ϵ,δ)-DP [66]	1.20	0.0001	0.5259	0.5146
ϵ-DP	1.00	0.0000	0.5217	0.5105
(ϵ,δ)-DP [66]	0.86	0.0011	0.5241	0.5130
ϵ-DP	0.70	0.0000	0.5234	0.5123
(ϵ,δ)-DP [66]	0.60	0.0082	0.5212	0.5102

In addition, I also reimplement the (ϵ, δ)-differentially private query log anonymization algorithm from [66] in order to examine whether my utility evaluation methodology by retrieval effectiveness can be used to support other baseline query log anonymization. Practically, I use the same major parameters (noise scales b and cut-off threshold K) in my ϵ-differentially private anonymization algorithm and the (ϵ, δ)-differentially private anonymization algorithm. Table 3.7 shows that the ϵ-DP and (ϵ, δ)-DP achieve similar utility scores with different privacy guarantees, while using my utility evaluation system based on retrieval effectiveness. The data releaser may choose to apply ϵ-DP or (ϵ, δ)-DP according to the detailed privacy guarantee that is expected.

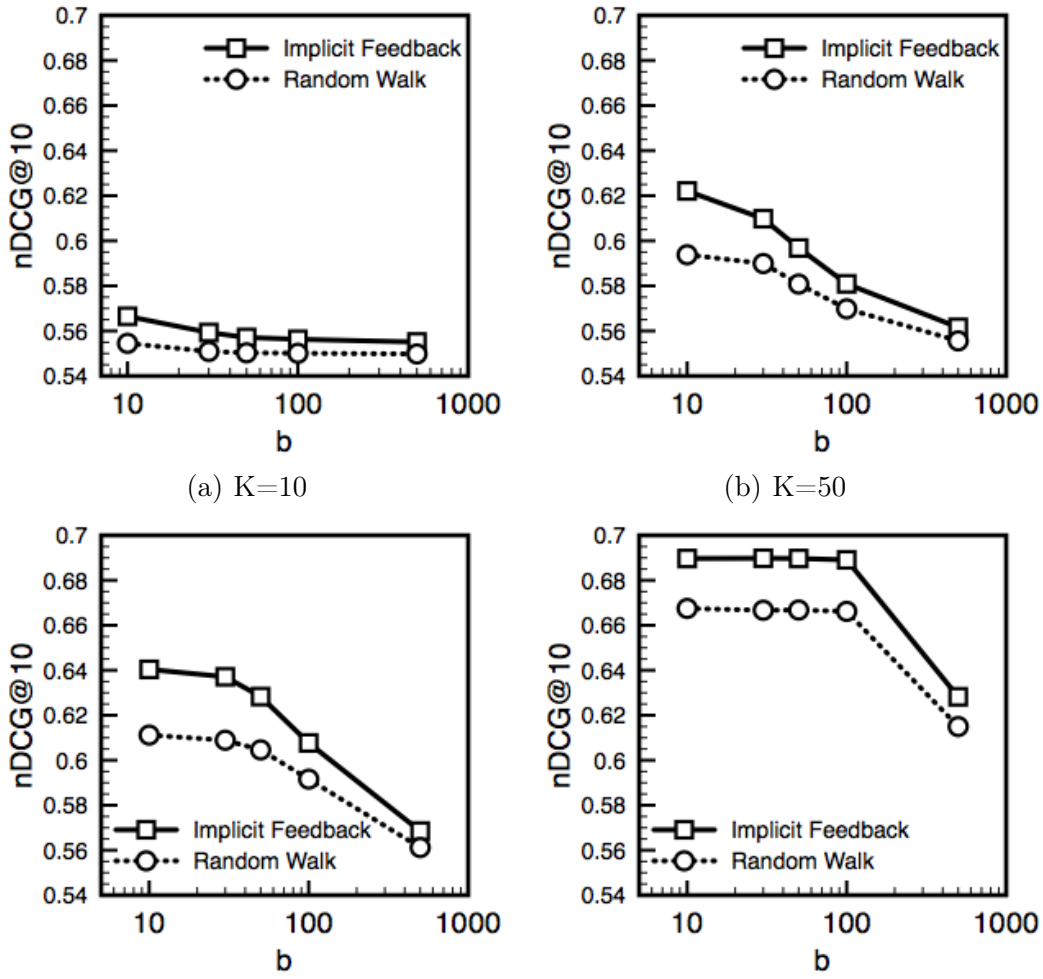
3.5.3 Impact of K and b on privacy and utility

In this section, I describe experiments that investigate the impact of the frequency cut-off threshold K and the noise scale parameter b on privacy and utility. Specifically,

I adjust different parameter settings to see when the privacy and utility values change significantly.

Figure 3.5 shows the impact of K and b on utility score $nDCG$@10. The results are based on document retrieval evaluation using Q' with varying K and b values while fixing other parameters. Among the three retrieval algorithms used, two of them are based on random walk models and are similar to each other. In this part, I focus on results based on the regular random walk algorithm and the implicit feedback algorithm. Each subgraph shows experiments using a different K value with K ranging from 10 to 500. Each data point on the subgraph represents the average of a set of 5-fold cross-validation results from the two retrieval algorithms (Implicit Feedback and Random Walk algorithm). Within each subgraph, all the data points share the same q_f, c_f and K values. They also use the same Q' size. Therefore, the results within each subgraph highlight the effect of different values of b. In general, as b increases, the utility $nDCG$@10 decreases. This matches intuition since I expect larger noise scales to reduce retrieval performance and cause decreased utility.

Figure 3.6 presents the impact of K and b on privacy. It shows the ϵ values for 25 different released query logs with varying K and b values while other parameters are fixed. They are organized in 5 subgraphs that show the b-ϵ relationships, each with different K values. In each subgraph, I label each point with the evaluated utility score ($nDCG$@10 from the implicit feedback algorithm). The figures show that the ϵ value is not always monotonically related to b. In graphs 3.6(c) and 3.6(d), I can observe



(a) K=10 (b) K=50

(c) K=100 (d) K=500

Figure 3.5: Impact of K and b on utility score $nDCG$@10.

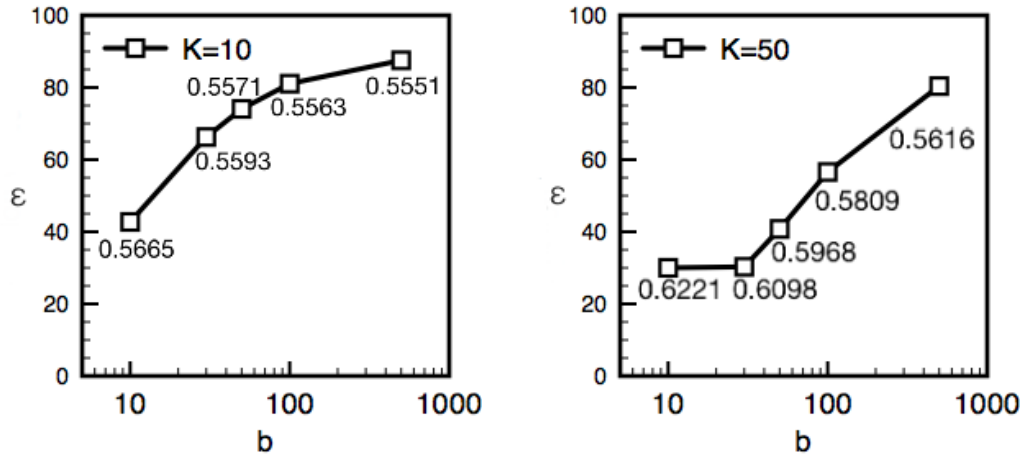
turning points with minimum ϵ values. These data points represent the smallest ϵ value I can achieve, i.e. the strongest privacy. I also notice that the utility at these points remains high. As b increases after the turning point, the performance decreases both regarding privacy (greater ϵ) and utility (smaller $nDCG$@10). Such turning points

can be mathematically calculated from Equation 5.1. I get such turning points when

$$\frac{e^{1/b}}{p_g} = 1 + \frac{1}{2e^{(K-1)/b} - 1} \tag{3.11}$$

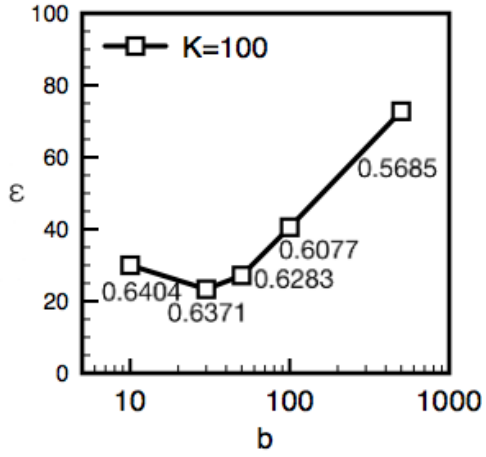
Actually, I recommend that the data releaser use the parameter combinations at those turning points since they are taking minimal ϵ values (strong privacy) and good utility values (before dropping significantly according to figure 3.5).

An additional observation is that the utility score is less sensitive to the noise scale b when b is much smaller than K. Finally, I can see that the implicit feedback algorithm performs better than the random walk algorithm, but they share similar patterns of retrieval performance as b changes. Note, because different values of K lead to different sizes of data, comparing across subgraphs does not make sense.

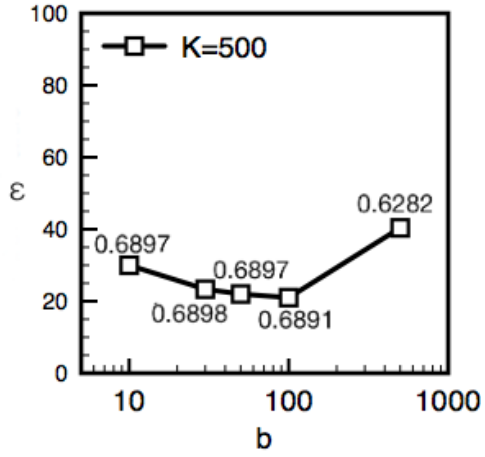


(a) K=10

(b) K=50



(c) K=100



(d) K=500

Figure 3.6: Impact of K and b on privacy level ϵ. The data points are marked with their corresponding $nDCG$@10 scores using the Implicit Feedback algorithm.

3.5.4 Impact of q_f and c_f on privacy and utility

Another set of experiments revealing the privacy-utility trade-off is necessary to investigate the performance differences caused by q_f (query limitations from each user in Q) and c_f (clicking limitations from each user in Q). It is worth noting from Equation

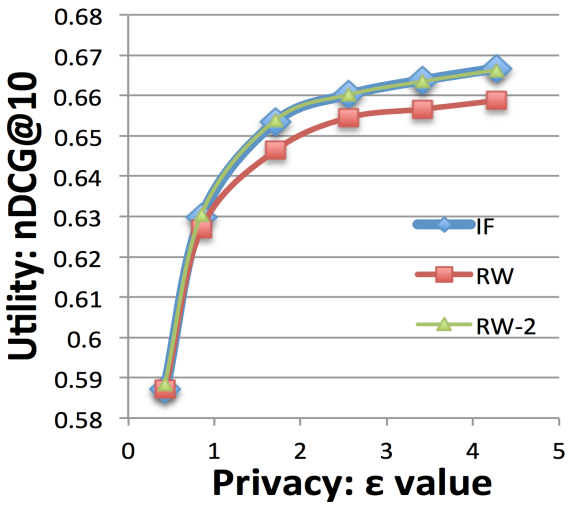


Figure 3.7: The tradeoff between privacy (ϵ value) and utility ($nDCG$@10).

5.1 that ϵ is very sensitive to q_f and c_f because the sensitivity of the algorithm is

$$\Delta f = Max(q_f, c_f) \tag{3.12}$$

while the ϵ value is linearly related to Δf. Since q_f and c_f have obvious influence on the value of ϵ, we can observe the direct privacy-utility trade-off in this experiment. Figure 3.7 presents the relationship between privacy (ϵ value) and utility (nDCG@10 value). All runs are evaluated on the same test set. This experiment takes fixed values for $K = 10$ and $b = 10$ while leaving ϵ changes according to varying q_f and c_f values. In this experiment, there are no limitations to the scale of the query log Q, which is the case in real data release processes. On one hand, greater values of q_f and c_f make use of the larger scale of input data which may increase the data accuracy

Table 3.8: General relationship between q_f,c_f and ϵ, when $K = 10$ and $b = 10$.

$q_f = c_f$	1	2	4	6	8	10	20	40	80
ϵ	0.43	0.85	1.71	2.56	3.42	4.27	8.54	17.08	34.16

Table 3.9: Detailed results for the tradeoff between Privacy (ϵ value) and Utility (nDCG@10).

$q_f = c_f$	ϵ	IF	RW	RW-2
1	0.43	0.5872	0.5871	0.5884
2	0.85	0.6298	0.6270	0.6305
4	1.71	0.6535	0.6464	0.6539
6	2.56	0.6602	0.6545	0.6602
8	3.42	0.6638	0.6566	0.6635
10	4.27	0.6667	0.6587	0.6663

and retrieval utility. On the other hand, greater values of q_f and c_f naturally lead to greater value of ϵ, which weakens the privacy.

According to the results based on all three retrieval algorithms, we do observe the obvious privacy-utility trade-off from the positive correlation between nDCG@10 and ϵ, while a smaller ϵ value means stronger privacy. Table 3.9 presents the detailed results of this experiment. If the data releaser prefers stronger privacy with smaller ϵ value, the retrieval utility (nDCG@10) may drop significantly from 0.6663 (when $\epsilon = 4.27$) to 0.5884 (when $\epsilon = 0.43$, taking only one research record from each individual). Alternatively, if the data releaser prefers better retrieval utility, the data limitations q_f and c_f should be relaxed in order to include more raw data from the original dataset, which hurts some privacy by having a greater value of ϵ. I recommend

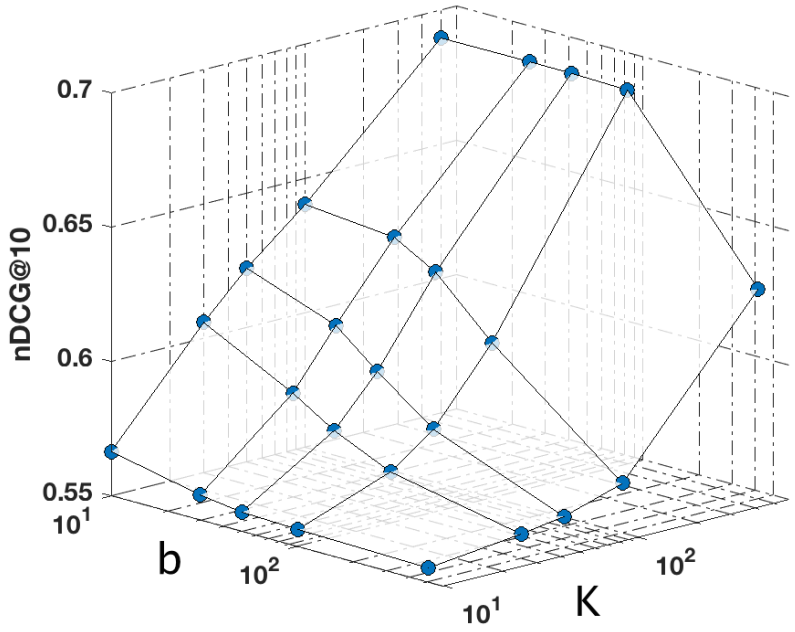


Figure 3.8: Parameter recommendations for noise b, query cutoff K and their relationship with utility score $nDCG$@10.

that the data releaser carefully consider this trade-off between privacy and utility in order to do the most appropriate query log anonymization.

3.5.5 Overall analysis and parameter recommendations

In this part, I analyze the privacy parameters and give recommendations to query log owners who want to use my framework to release and evaluate query logs.

Figure 3.8 provides an intuitive presentation by plotting the trends between the utility scores and the parameter values in a 3-D graph. In this graph, I observe that as the noise level b increases, the utility scores $nDCG$@10 decreases. I also observe that the utility score is less sensitive to b when b is much smaller than K.

This observation matches the intuition that larger noise (compared with K) will reduce retrieval performance and cause decreased utility. Hence, I recommend using smaller b values (around $b = 10$) when K values are small (e.g., $K = 10$, 30, or 50). As K gets larger (e.g., $K = 100$ or 500), it is better to set b to be the same scale as K and close to the turning point. These settings should achieve the best combination of privacy and utility.

Table 3.10 shows the optimal parameter combinations for query log anonymization given fixed privacy budget $\epsilon = 0.5, 1.0, 2.0$ and 4.0. The utility scores are all from the RW-2 retrieval algorithm. The parameter calculations are based on Theorem 1 in Chapter 3 and the "turning points" as I discussed in section 3.5.3. By using such combinations of parameters, we can make the best use of the privacy budget to pursue better utility. According to the results, while fixing the ϵ value, maximal utility values may be achieved when taking relatively small combinations of the parameters (bold font in Table 3.10). The utility appears to decrease when the parameter combinations shift away from the optimal combinations. Under certain fixed ϵ values, I recommend using those parameter combinations with optimal utility values. In addition, when we compare such bold font runs with different ϵ values, it is easy to observe that runs with smaller ϵ values (stronger privacy) may achieve lower utility values (worse utility), while runs with larger ϵ values (weaker privacy) may achieve higher utility values (better utility). That is a direct trade-off between privacy and utility.

Table 3.10: Optimal parameter combinations for query log anonymization given fixed privacy value ϵ.

ϵ	$q_f = c_f$	b	K	Utility
0.5	**1**	**6.0**	**8.08**	**0.6114**
0.5	2	14.0	28.74	0.5757
0.5	3	22.0	54.25	0.5475
0.5	4	30.0	82.74	0.5253
0.5	5	38.0	113.39	0.5090
1.0	1	3.0	2.70	0.6359
1.0	**2**	**7.0**	**10.26**	**0.6391**
1.0	3	11.0	20.25	0.6335
1.0	4	15.0	31.72	0.6245
1.0	5	19.0	44.27	0.6131
1.0	6	23.0	57.67	0.6051
1.0	7	27.0	71.77	0.5949
1.0	8	31.0	86.46	0.5863
1.0	9	35.0	101.68	0.5766
1.0	20	75.0	273.33	0.5260
2.0	2	3.5	3.45	0.6534
2.0	**4**	**7.5**	**11.41**	**0.6575**
2.0	6	11.5	21.61	0.6555
2.0	8	15.5	33.24	0.6541
2.0	10	19.5	45.90	0.6500
2.0	12	23.5	59.40	0.6446
2.0	14	27.5	73.58	0.6400
2.0	16	31.5	88.34	0.6345
2.0	18	35.5	103.61	0.6281
2.0	20	39.5	119.33	0.6213
4.0	5	4.75	5.60	0.6651
4.0	**10**	**9.75**	**16.94**	**0.6660**
4.0	15	14.75	30.97	0.6651
4.0	20	19.75	46.73	0.6635
4.0	25	24.75	63.76	0.6612

In summary, by carefully analyzing the expectation for privacy and utility, the data releaser could find proper combinations of the detailed parameters in the query log anonymization process according to the insights of these experiments. A good balance between privacy and utility in query log anonymization can be found.

3.6 Chapter Summary

In this chapter, I introduce a framework for anonymizing and evaluating the utility and privacy of the anonymized log. To the best of my knowledge, this work is the first to generate anonymized query logs that have been measured for utility on actual web search tasks. The framework provides effective query log anonymization algorithms that place adequate privacy guards on those logs while simultaneously maintaining high retrieval utility. The experiments demonstrate that the proposed framework is very effective – a statistical significance test (two-tailed t-test, $p < 0.01$) shows that popular web search algorithms using the anonymized logs perform comparably with those using logs before anonymization. In addition, my comparative experiments illustrate the privacy-utility trade-off in query log release. In particular, the stricter the privacy standard required, the lower the utility or usefulness of the released query log regarding web search. The work presented in this chapter shows that the differentially private query log is able to well support typical web search tasks. I hope that it encourages web search engine companies to release logs for research purposes.

Chapter 4

Query Log Anonymization for Sessions

In chapter 3, I have shown that the web search query logs can be properly anonymized with differential privacy in order to support typical web search research. It is like single record privacy protection [135] in the context of query log anonymization. In such a basic form, each web search query and its associated user actions are treated as one block for anonymization; each block is independent from each other. It might be sufficient to support ad-hoc retrieval that handles queries independently but will be not adequate for more complex IR tasks that require knowledge of query sequences.

In this chapter, I continue my research to support more complex IR applications. Query session data, as a special form of sequential data in IR, contains important information about the original web search retrieval process. It is a helpful complement to the click-through data in query logs. A properly anonymized query log containing both click-through data and session data can be used for many other IR applications such as query suggestion and session search. The challenge of how to anonymize session logs in order to support complex IR tasks remains.

In this chapter, we tackle this challenge by keeping session information in differentially privately anonymized logs so that an anonymized log can support IR tasks that

need query sequence information. In particular, I demonstrate how anonymized query logs can be used for the tasks of query suggestion and session search. I also provide analysis of how to achieve a proper balance between privacy and search utility.

This chapter involves contents from my publication [128].

4.1 Background

This research work of query log anonymization for session data is an important extension of the query log anonymization research I presented in chapter 3. As I have introduced earlier, recent research has made progress in the task of applying Differential Privacy in query log anonymization [40, 66, 135]. However, there are two main limitations of the current approaches.

First, the utility of anonymized query logs should be properly measured and should be task-based. Most existing approaches simply measure the utility of anonymized logs by the percentage of what is remaining instead of evaluating through actual IR tasks and IR effectiveness measures [66]. However, utility should be task dependent. We argue that we should use the latter. In this study, I find that percentage of the kept data and real IR utility is different, and sometimes they could even contradict each other.

Second and more importantly, sequential data and search session data was not involved in the existing query log anonymization research. Existing work only takes care of single record privacy protection [135]. This simplification is not sufficient for

Table 4.1: Search session examples from TREC 2012 session track.

session 6
1.pocono mountains pennsylvania
2.pocono mountains pennsylvania hotels
3.pocono mountains pennsylvania things to do
4.pocono mountains pennsylvania hotels
5.pocono mountains camelbeach
6.pocono mountains camelbeach hotel
7.pocono mountains chateau resort
8.pocono mountains chateau resort attractions
9.pocono mountains chateau resort getting to
10.chateau resort getting to
11.pocono mountains chateau resort directions
session 28
1.france world cup 98 reaction stock market
2.france world cup 98 reaction
3.france world cup 98
session 32
1.bollywood legislation
2.bollywood law
session 37
1.Merck lobbists
2.Merck lobbying US policy
session 85
1.glass blowing
2.glass blowing science
3.scientific glass blowing

complex IR tasks that require the use of sessions, a particular form of sequential data, from a query log.

Table 4.1 shows search session examples from the TREC 2012 Session track [62, 63]. In each session, the user keeps modifying the queries several times until the retrieved result satisfies its information need or leads to the frustration of the user.

From Table 4.1, we can observe how queries change constantly in a session, which is an important part of information in the query log. For instance, the patterns of query changes include general to specific (pocono mountains → pocono mountains park), specific to general (france world cup 98 reaction → france world cup 98), drifting from one to another (pocono mountains park → pocono mountains shopping), or slightly different expressions for the same information need (glass blowing science → scientific glass blowing). These changes vary and sometimes even look random (gun homicides australia → martin bryant port arthur massacre), which increases the difficulty of understanding user intention. However, since query changes are made after the user examines search results, we believe that such query change is an important form of feedback. We hence propose to study and utilize query changes to facilitate better session search. My previous work on session search have shown that such query change information hidden in the session data can be well used to support session search [43, 130]. Figure 4.1 presents the interactive process between the user and the search engine during a search session. Such a dynamic process can only be utilized if the anonymized query log contains the session data. In fact, session data in the query log can be used in a variety of IR applications such as query suggestion, dynamic search and session search. Furthermore, such query sequence data can also be beneficial to research in data mining, NLP, user study and other data-driven research involving natural language corpora. In summary, the session log data could be much better distributed and used by our IR community if we can come out with a proper

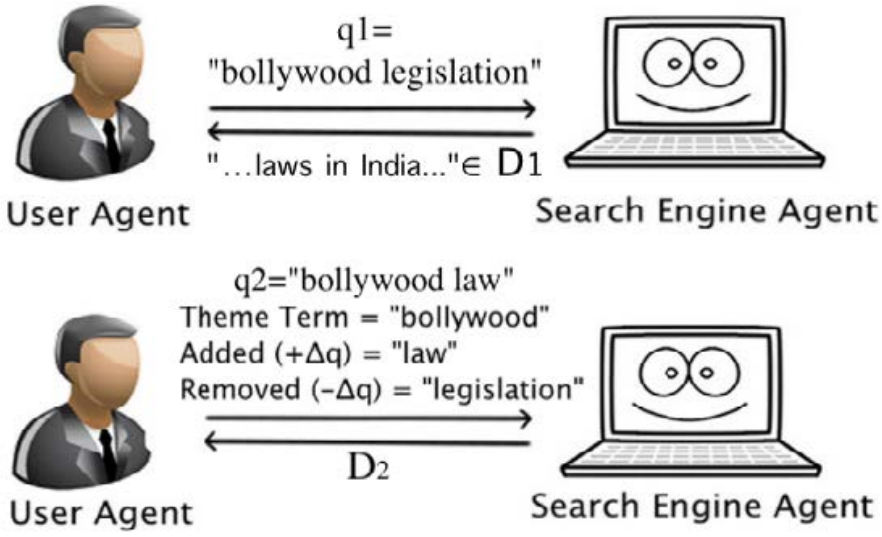


Figure 4.1: Search session example: the user modifies queries in the interactive process.

mechanism to anonymize session log. This is my major motivation to do session log anonymization.

A major challenge of session log anonymization is the sparsity of re-occurring sessions. The sparsity comes from the large number of possible query sequences. From the perspective of privacy research, the sparsity of sessions would require a greater scale of noise to be added and would make privacy protection harder to keep under control. From the perspective of IR utility, the sparsity of sessions also makes it more complicated to distinguish higher frequency sessions, which may be more valuable to support IR applications, from the other sessions.

In this research, I am developing query log anonymization algorithms that can release sequential queries as search sessions after anonymization [128]. Then, I use the

anonymized query log containing session data to support the typical IR application of query suggestion which depends on query sequences in a search session. To the best of our knowledge, this work is the first to study how session data can be released anonymously and be useful to support complex IR tasks.

To be specific, this research aims to anonymize a query log with (ϵ, δ)-differential privacy and release a differential privacy protected query log consisting of click-through data as well as session data. I will show that the anonymized query log can be used to support complex IR applications such as query suggestion and session search and report the remaining IR utility of the anonymized log.

In summary, I mostly study the following research questions in this chapter:

(1)How can a query log containing session data be released by differential privacy?

(2)How well can anonymized query logs containing frequent search sessions be used to support query suggestion?

4.2 Anonymization Algorithm for Sessions

I develop a new query log anonymization algorithm $A_{Session}$ that would release two parts of data as output. The first part contains query sequences in sessions. The second part contains the frequent query click-through data. Accordingly, my session log anonymization algorithm consists of two parts. Both parts of the algorithm take the original query log Q as input, and the mechanisms should satisfy differential privacy. Part 1 of $A_{Session}$ releases frequent search sessions as output, while part 2

of $A_{Session}$ releases frequent query click-through data as output. The major challenge falls in how to design the session data release part.

Because of the property of natural language, the total number of unique search queries could be infinite. Exact matches of search sessions are thus rare because sessions are ordered sequences of those natural language queries. As we know that low frequency data would always be difficult to protect by anonymization algorithms. This poses a challenge to session-based query log anonymization. To address this challenge, the following are potential solutions to develop the session log anonymization algorithm:

- A straightforward solution may be to consider each search session as an individual record in the raw data and apply a differentially private mechanism to filter the records. However, the major issue of this approach is that the output scale may be largely limited according to the low frequency of each individual distinct search session.

- A potential solution is to consider all non-trivial subsequences of a query sequence in the raw data as individual records of search session. The idea for this approach is to reduce the sparsity of session distribution so that the search sessions could get more occurrence frequency, which may make the session anonymization algorithm achieve better balance between privacy and utility.

- I may also be able to develop session log anonymization mechanisms based on the frequent sequence mining approaches. However, query log anonymization

works on an infinite domain of attribute value (search queries), which is different from the frequent sequence mining scenario in data mining where the attribute value range usually has a much smaller size. It will be my future research work to investigate which may be the most proper way to do the session log anonymization task.

4.2.1 $A_{Session}$: query log anonymization algorithm for sessions

Our query log anonymization algorithm releases two types of data: query sequences in sessions and click-throughs. The anonymization algorithm also consists of two parts to produce the two types of data. Both parts take the original query log Q as the input and satisfy the (ϵ, δ)-differential privacy. Part 1 of $A_{Session}$ releases frequent search sessions as the output, while part 2 of $A_{Session}$ releases frequent query click-through data as the output. We merge their outputs and release both in algorithm $A_{Session}$.

4.2.1.1 Part 1 of $A_{Session}$: Releasing session data

Because of the property of nature of natural language, the total number of unique search queries could be infinite. Exact match of the two search sessions is thus rare because sessions are ordered sequences of natural language queries. As we know, it is difficult to protect low frequency data from being re-identified. To address this challenge, we propose to increase the session and query counts by adding all non-trivial subsequences of sessions as individual sessions in the query log.

We propose part 1 of the $A_{Session}$ algorithm to release search sessions with (ϵ, δ)-differential privacy. Each non-trivial subsequence of the original sessions would contribute a count to reduce the sparsity of sessions. We add noise to the session frequency by applying the Laplace mechanism [40, 66].

Part 1 of the algorithm $A_{Session}$ takes in the following inputs: the original query log Q, the session timeout threshold T_{Gap}, the max number of sessions per user l_s , the max number of queries per session l_q, Laplace noise scale b, and the session frequency cut-off threshold K. The algorithm is described as the following:

1) Session segmentation. We define each session S as an ordered sequence of search queries:

$$S = [q_1, q_2, ..., q_{|S|}] \tag{4.1}$$

where $|S|$ is the number of queries in S.

We segment the sessions in Q based on either i) if they are from two different users, or ii) if the timestamp difference between these two queries is greater than the session timeout threshold T_{Gap}. Practically, we take 30 minutes as the timeout threshold according to previous work in session search [43]. Then we transfer Q into Q_{seg}, which is a set of sessions:

$$Q_{seg} = \{S_1, S_2, ..., S_{|Q_{seg}|}\} \tag{4.2}$$

Without loss of generality, we only keep non-trivial sessions that consist of at least two queries. "Sessions" with only a single query are excluded from Q_{seg}.

2) Subsequence mining. To increase the number of repeated sessions, we take into consideration all non-trivial subsequences within a session and treat them as sessions too. The relative orders from the original session are maintained with the permission to have some original queries missing in the new subsequences. That is to say, each subsequence of a session $S = [q_1, q_2, ..., q_{|S|}]$ is a sequence $S' = [q'_1, ...q'_{|S'|}]$ defined by $q'_t = q_{n_t}$, where the indices $n_1 < n_2 < ... < n_{|S'|}$ are monotonically increasing and $|S'| >= 2$. For example, if the original session is (q_1, q_2, q_3, q_4), our algorithm adds one count to each of the following 11 sessions:

$$\begin{gathered}(q_1, q_2, q_3, q_4), (q_1, q_2, q_3), (q_1, q_2, q_4), (q_1, q_3, q_4), (q_2, q_3, q_4) \\ (q_1, q_2), (q_1, q_3), (q_1, q_4), (q_2, q_3), (q_2, q_4), (q_3, q_4)\end{gathered} \tag{4.3}$$

Hence, the search sessions, or query sequences, may get greater frequency of occurrence from the same raw dataset.

3) Sensitivity control. In order to make sure the presence or absence of each individual user would not make too much of an impact on the statistics that we release, the sensitivity Δf of $A_{session}$ is controlled by adjusting the max number of sessions per user l_s and the max number of queries per session l_q. Sessions longer than l_q queries are trimmed down to only the first l_q queries. As a result, the sensitivity is kept as

$$\Delta f = l_s \times (2^{l_q} - 1 - l_q) \tag{4.4}$$

4) Session release decision. We denote the session counts for a session S after subsequence mining as $C(S)$. Then we apply the Laplace mechanism on the session

counts by adding i.i.d. noise to each count. We release session S iff. $C(S)+noise > K$, where $noise \sim Lap(0, b)$, and K is the session frequency cut-off threshold. The decision of whether to release session S is made based on:

$$\begin{cases} \text{if } C(S) + Lap(0, b) > K, & \text{Release session } S \\ \text{otherwise,} & \text{Do not release } S \end{cases}$$

5) Session count release. For all sessions S that we have decided to release, their counts generated from the previous step form a biased sample since they are all selected because their values are greater than K. However, we need to make sure the actual released counts still follow an unbiased Laplace distribution $Lap(C(S), b)$. We therefore need to do the sampling again and release the sessions together with their perturbed count $C(S) + Lap(0, b)$. The released perturbed frequency count $C(S) + Lap(0, b)$ follows the distribution of $Lap(C(S), b)$, which becomes the Laplace Mechanism [30] for Differential Privacy.

6) Output: A set of frequent sessions along with their corresponding counts, $\{(S, C(S))\}$, while each session S is in the form of an ordered sequence of free text queries. Table 4.3 shows an example output by part 1 of $A_{Session}$.

I state here that this session log anonymization part of the algorithm $A_{Session}$ achieves (ϵ, δ)-differential privacy while ϵ and δ are:

$$\begin{aligned} \epsilon &= l_s(2^{l_q} - 1 - l_q) \cdot (ln(Max\{e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b} - 1}\}) + 1/b) \\ \delta &= 0.5 l_s(2^{l_q} - 1 - l_q) \cdot exp(\frac{l_s(2^{l_q} - 1 - l_q) - K}{b}) \end{aligned} \tag{4.5}$$

I leave the proof for this in chapter 5.

4.2.1.2 Part 2 of $A_{Session}$: Releasing query click-through data.

We also release click-through data, i.e. a query and the URLs that a user clicked for the query. Part 2 of the algorithm $A_{Session}$ releases the click-through data as a tuple of query, clicked document and the count for the pair, $[q, d, c(q, d)]$. The released data satisfies the (ϵ, δ)-differential privacy. The count $c(q, d)$ is the frequency after adding Laplacian noise for a query-URL pair (q, d).

Part 2 of $A_{Session}$ is similar to algorithm A_{Click} in Chapter 3. However, I design part 2 of $A_{Session}$ to satisfy (ϵ, δ)-differential privacy rather than ϵ-differential privacy in A_{Click}. This makes the click-through part of the algorithm more general and without the reliance of any external dataset such as the query pool concept I proposed in Chapter 3.

This part of the $A_{Session}$ algorithm takes in the following inputs: the query log Q, query click records limit per user l, the Laplace noise scale b, and the frequency threshold K. Note that both parts of the $A_{session}$ algorithm need to share the same privacy parameter settings b and K. It is because the overall privacy guarantee is bottlenecked by the algorithm that has the weaker differential privacy guarantee.

This part of the $A_{Session}$ algorithm releases the click-through data to assist session data release using the following steps:

1) Sensitivity control for clicks. For each user in Q, we only keep the first l click records for each user and ignore the rest from the same user to make sure that data from any user will not contribute too much to the overall frequency statistics.

2) Query-clickthrough release decision. This is similar to step 4 in part 1 of the algorithm. We first count the total number of occurrences for each query-clickthrough pair as $C(q,d)$. We then decide to release a pair (q,d) iff. $C(q,d)+noise > K$, where $noise \sim Lap(0,b)$ and K is the frequency cut-off threshold.

3) Release query-clickthrough tuples. This is similar to step 5 of the part 1 of the algorithm. If it has been decided to release a (q,d) pair at the previous step, we perform a sampling again from the Laplacian distribution for added noise and release a 3-tuple $[q,d,C(q,d)+noise]$ where the new i.i.d. $noise \sim Lap(0,b)$; otherwise, we won't release anything for the (q,d) pair.

4) Output: A set of query-clickthrough tuples in the form of (Query q, Document d, Fuzzed Count $C(q,d)+noise$). Table 4.2 shows an example output by part 2 of $A_{Session}$.

This part of the algorithm $A_{Session}$ achieves (ϵ,δ)-differential privacy while ϵ and δ are:

$$\begin{aligned} \epsilon &= l \cdot ln(Max\{e^{1/b}, 1+\frac{1}{2e^{(K-1)/b}-1}\}) + l/b \\ \delta &= 0.5l \cdot exp(\frac{l-K}{b}) \end{aligned} \tag{4.6}$$

This part of the algorithm is similar to a simpler version of the anonymization algorithm proposed in [66], and I omit the proof in this dissertation.

4.2.2 Composition analysis

Table 4.2 and 4.3 present examples of the outputs from the two parts of the algorithm.

Table 4.2: $A_{Session}$ output example part 1: Click-through data.

Query	Clicked URL	Counts
weather	http://www.weather.com	4190
weather	http://weather.yahoo.com	1035
aol weather	http://weather.aol.com	30
aol weather	http://aolsvc.weather.aol.com	16
blue book	http://www.kbb.com	33
blue book	http://www.nadaguides.com	1
hairstyles	http://www.hairfinder.com	5
hairstyles	http://www.1001-hairstyles.com	19
hairstyles	http://www.hair-styles.org	21
hairstyles	http://hairstyles.free-beauty-tips.com	16
...	...	...

Table 4.3: $A_{Session}$ output example part 2: Session data.

Session 1	Session 2
q_1=daily record morristown nj	q_1=ny lottery
q_2=star ledger newark nj	q_2=pa lottery
q_3=google	q_3=nj lottery
	q_4=ny lottery
Counts: 11	Counts: 16

It is worth noting here that the two parts of the algorithm anonymize and release data independently. They can be applied separately. Moreover, both parts of the algorithm may be replaced with other query log anonymization approaches that generate the same output format. For instance, the second part of the algorithm may be replaced by the algorithm I proposed in Chapter 3 which achieves ϵ-differential privacy.

According to Theorem 3.16 from the *The Algorithmic Foundations of Differential Privacy* [32], the composition is "automatic" when we are combining multiple building blocks designing differentially private algorithms. If Part 1 of the algorithm achieves (ϵ_1, δ_1) differential privacy, while Part 2 of the algorithm achieves (ϵ_2, δ_2) differential privacy, then the overall algorithm achieves $(\epsilon = \epsilon_1 + \epsilon_2, \delta = \delta_1 + \delta_2)$ differential privacy.

For instance, if we perform query log anonymization using both parts of the $A_{Session}$ algorithm for the same input query log, then the overall algorithm will achieve (ϵ, δ)-differential privacy, where

$$\begin{aligned}
\epsilon &= l \cdot ln(Max\{e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b} - 1}\}) + l/b \\
&\quad + l_s(2^{l_q} - 1 - l_q) \cdot (ln(Max\{e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b} - 1}\}) + 1/b) \\
\delta &= 0.5l \cdot exp(\frac{l - K}{b}) + 0.5l_s(2^{l_q} - 1 - l_q) \cdot exp(\frac{l_s(2^{l_q} - 1 - l_q) - K}{b})
\end{aligned} \tag{4.7}$$

4.3 Utility Measurement with Query Suggestion and Session Search

Data utility should be evaluated task-dependently. However, most prior work simply measures the utility by how much data is kept. In our experiments, we reveal that amount of kept data and the actual task-dependent utility do not agree. In this research, we use real IR tasks – query suggestion and session search – and classic IR evaluation metrics to measure the utility of a query log after anonymization.

4.3.1 Query suggestion

Query Suggestion is a typical IR application that is using web search query log as well as session log data. I use query suggestion as the principle application to present how the anonymized query log can be used to support session-based tasks. In this section, I introduce the query suggestion task as well as the corresponding utility measurement.

4.3.1.1 The task of query suggestion

Query suggestion is a popular IR task. The goal of the task is to predict the next search query that a user is going to write. Given a session S with $n+t$ queries, $S = [q_1, q_2, ..., q_{n-1}, q_n, q_{n+1}, ..., q_{n+t}]$ the task of query suggestion is to generate a ranked list of suggested queries $\{q'_1, ..., q'_m\}$ as the candidates of the next query after q_n. For evaluation purposes, we use the queries that are after q_n in the same session and are generated by the same user as the ground truth. That is, $Truth(q_n) = \{q_{n+1}, ..., q_{n+t}\}$. The results can then be evaluated by comparing between the generated ranked list $\{q'_1, ..., q'_m\}$ and the ground truth set $Truth(q_n)$.

In this work, we use classic IR metrics Precision and Recall to evaluate the utility for query suggestion. In particular, we report Precision@5 and Recall@5:

$$Precision@5 = \frac{1}{5}\sum_{i=1}^{5} Hit(i);\ Recall@5 = \frac{1}{t}\sum_{i=1}^{5} Hit(i) \tag{4.8}$$

$$Hit(i) = \begin{cases} 1, & \text{if } q'_i \in Truth(q_n). \\ 0, & otherwise. \end{cases}$$

where $t = |Truth(q_n)|$, $Hit(i)$ shows whether the i^{th} predicted query q'_i hits the ground truth, $1 \leq i \leq 5$.

4.3.1.2 Query suggestion using anonymized logs

We build two graphs G_s and G from the anonymized log Q' to support query suggestion. We use the first graph, a query-flow graph $G_s = (V_s, E_s)$, to organize queries in sessions. We use the second graph, a query-URL bipartite graph $G = (V_q, V_d, E)$, to organize relations between queries and URLs in anonymized click-through data.

With the help of the anonymized session information, we are able to create a query-flow graph as in Boldi et al. [9]. The query-flow graph G_s organizes the ordered query transitions from the query sequences in Q'. In particular, $G_s = (V_s, E_s)$. V_s contains the set of query vertex in the graph and E_s is the set of edges connecting queries that have occurred adjacently. In G_s, we denote $e(q_i, q_j)$ as the edge weight between the transition from q_i to q_j, which is number of co-occurrences of q_i and q_j in the anonymized session log. Note that q_j is any query that appears after q_i and is not restricted to be the query immediately after q_i in the session. We also denote $d(q_i)$ as the out-degree of q_i. Then the probability of q_j following q_i in the same session can be calculated as $e(q_i, q_j)/d(q_i)$, if only based on the session data.

We also use the query click-through data. We organize queries and their corresponding click-through URLs into a query-URL bipartite graph $G = (V_q, V_d, E)$. V_q is the set of query nodes, V_d is the set of document (URL) nodes, and E is the set of weighted edges in G. According to this bipartite graph, we represent each query $q \in V_q$ as a vector of weighted documents $\overrightarrow{q}$. Then we calculate the similarities between any two queries q_i and q_j by their normalized dot product $\overrightarrow{q_i} \cdot \overrightarrow{q_j} / (|\overrightarrow{q_i}| \cdot |\overrightarrow{q_j}|)$. We use a variation of a state-of-the-art query suggestion approach [13] to quantify the similarities between the queries. The difference is that we generate a ranked list of relevant queries for each query q, rather than allocating queries into clusters [13]. The ranked lists of the relevant queries is equivalent to the results generated based on the Euclidean distance between the normalized feature vectors as in [13] according to the geometric properties of the vector space.

Finally, we combine the two scores from both G_s and G. The overall probability of having the candidate query q_j follow q_i is calculated as:

$$P(q_i, q_j) = \lambda \frac{\overrightarrow{q_i} \cdot \overrightarrow{q_j}}{|\overrightarrow{q_i}| \cdot |\overrightarrow{q_j}|} + (1 - \lambda) \frac{e(q_i, q_j)}{d(q_i)} \tag{4.9}$$

where λ is a parameter to control the value contributed between G and G_s.

4.3.2 Session search

Session search is also a major session-based IR application. I use session search as another example to present how the anonymized query log can be used to support the

task. In this section, I introduce the session search task as well as the corresponding utility measurement.

4.3.2.1 The task of session search

Session search is a complex search process involving multiple search iterations triggered by continuous query reformulations. It gets the name of session search because it is a document retrieval task for the entire session, rather than for individual queries. With the help of the anonymized query logs, the goal of the task is to generate a ranked list of documents $[d_1, d_2, ...d_m]$ relevant for the entire session. We use the official TREC [63] evaluation metrics for session search: nDCG@10 [53] (Equa. 4.10) and MAP (Mean Average Precision) (Equa. 4.11), as the utility metrics of the query log used for session search.

$$nDCG@10(S, D) = \{\sum_{r=1}^{10} \frac{rel_r}{log_2(r+1)}\}/\{\sum_{r=1}^{N_{Rel}} \frac{1}{log_2(r+1)}\} \tag{4.10}$$

$$MAP(S, D) = \{\sum_{r=1}^{10} (P(r) \cdot rel_r)\}/N_{Rel} \tag{4.11}$$

where D is the ranked list of documents retrieved for session S by our ranking algorithm. rel_r takes the value of 1 when the r^{th} ranked retrieved document $d_r \in D$ is relevant. Otherwise, $rel_r = 0$. $P(r)$ is the precision at cut-off r in the list. N_{Rel} is calculated by $Min\{$total number of relevant documents for S,10$\}$.

4.3.2.2 Session search using anonymized logs

State-of-the-art work [43, 78] in session search considers the procedure of session search as a Markov Decision Process. In this paper, we first calculate the relevance score $score(q_i, d)$ for each query q_i in session S using the anonymized query log Q' and generate a ranked list of relevant documents for each of them. Then, we combine the results from each query of S by an infinite horizon formula as proposed in Guan et al. [43].

Generate ranked lists of documents for each query. Different from the approach we used for query suggestion, here we apply a graph-based random walk framework [24] to support session search. We organize both session data and query click-through data into a graph G' and apply random walk on the graph. We define the graph as $G' = (V_q, V_d, E, E_q)$, where V_q is the set of query vertices, V_d is the set of document (URL) vertices, E is the set of weighted edges between a query vertex in V_q and a URL vertex in V_d, and E_q is the set of edges between query vertices in V_q.

Using the anonymized query click-through data, we calculate the edge weights $w(q, d)$ between a query vertex $q \in V_q$ and a document vertex $d \in V_d$ in a way similar to the click model proposed by Craswell and Szummer [24]. Using the anonymized session data, we calculate the directed edge weight $w(q_i, q_j)$ as the overall frequency of the ordered co-occurrence between q_i and q_j in the released sessions, while $q_j \in V_q$ appears later than $q_i \in V_q$ in the same session.

Now we can run the random walk model [24] on graph G'. The transition probabilities is calculated by:

$$P(v_i, v_j) = \begin{cases} (1-\lambda)\frac{w(v_i,v_j)}{\Sigma_x w(v_i,v_x)} & , \forall i \neq j \\ \lambda & , i = j \end{cases} \tag{4.12}$$

where λ is the self-transition rate in random walk, v_i, v_j, $v_x \in V_q \cup V_d$ are vertices of G', and $w(v_i, v_j)$ is the transition weight between vertex v_i and v_j as defined earlier.

For each given query q in the test set, we rank the URLs according to the descending order of the probabilities of staying at the corresponding URL after the random walk. It generates the relevance score: $score(q, d) = 1/$(rank of d in the ranked list) for each pair of (q, d).

Aggregate scores for the entire session. Given a session $S = [q_1, q_2, ..., q_n]$ from the test set, we first generate document relevant scores $score(q, d)$ for each query q_i as described above. Then, according to Guan et al. [43], we aggregate the scores for the entire session S as

$$score(S, d) = \sum_{i=1}^{n} \gamma^{n-i} score(q_i, d) \tag{4.13}$$

, where γ close to 1 is the discount factor which makes earlier queries in the session contribute less ($\gamma < 1$) or more ($\gamma > 1$) to the aggregated session relevance score.

4.4 Experiments

We evaluate our algorithms on the 2006 AOL dataset. The entire query log contains 36,389,567 search records. In total, there are 10,154,742 unique queries and 19,442,629

click-through records from 657,426 unique users over three months. We use nine-tenths of the query log as the original log Q to be anonymized, and reserve one-tenth of the data as the test set Q_{Test} for evaluating the IR applications. In the experiments, we compare a few query log anonymization schemes and use the anonymized logs Q' generated from each of them to test on the query suggestion task.

4.4.1 Anonymized methods to compare

The logs used in our experiments include the Original, KA (logs anonymized by k-anonymity), DP_C (logs anonymized by differential privacy, click-through data only), and DP_S (logs anonymized by differential privacy, containing both session and click-through data). The details of them are described as follows:

- Original: The original query log Q without anonymization.
- KA(K): The query log anonymized by the k-anonymity [105]. This log contains frequent click-through data from the original log while preserving certain privacy with k-anonymity. Major steps of the k-anonymity query log anonymization algorithm are as follows:
    1. Input: a query log Q, query click-through frequency threshold K.
    2. Count the number of users who formulate query q and click document d as $c(q, d)$.
    3. Release all tuples $[q, d, c(q, d)]$ iff. $c(q, d) > K$, where K is the frequency cut-off threshold.

4. Output: A set of tuples in the form of [Query q, Document d, User Count c(q,d)].

- DP$_C(\epsilon, \delta$;l,b,K): The second part (click-through data) of the query log anonymized by the (ϵ, δ)-differentially private algorithm $A_{Session}$ as presented in section 4.2.1.2, where l is the query click limits per user, b is the Laplacian noise scale, and K is the frequency threshold. The output format of the log is the same as KA(K).

- DP$_S(\epsilon, \delta; T_{Gap}, l_s, l_q$,b,K): The entire output of the query log anonymized by the (ϵ, δ)-differentially private algorithm $A_{Session}$ as presented in section 4.2.1. T_{Gap} is the session timeout threshold in minutes, l_s is the session limits per user, l_q is the query limits per session, b is the Laplacian noise scale, and K is the frequency threshold. The anonymized log contains both session and query click-through data.

In Table 4.4, I compare the level of differential privacy guarantees in DP$_S$ with different parameter settings. By comparing the typical runs with different parameter settings, I observe the ϵ value is very sensitive to session limit per user l_s and query limit per session l_q, while the δ value is also very sensitive to the value of noise scale b and frequency threshold k.

There are no hard rules for selection of parameter values. Generally, smaller ϵ and δ values lead to stronger privacy guarantees but the δ value cannot be too large. This gives the query log owner flexibility to pick proper privacy parameter values

Table 4.4: Privacy levels ϵ and δ for typical $A_{Session}$ runs.

Detail Parameters in DP	ϵ	δ
b=1, K=10, T_{Gap}=30, l_s=1, l_q=3	ϵ=8.00	$\delta = 4.95 * 10^{-3}$
b=1, K=20, T_{Gap}=30, l_s=1, l_q=3	ϵ=8.00	$\delta = 2.25 * 10^{-7}$
b=1, K=30, T_{Gap}=30, l_s=1, l_q=3	ϵ=8.00	$\delta = 1.02 * 10^{-11}$
b=3, K=10, T_{Gap}=30, l_s=1, l_q=3	ϵ=2.67	$\delta = 2.70 * 10^{-1}$
b=3, K=20, T_{Gap}=30, l_s=1, l_q=3	ϵ=2.67	$\delta = 9.66 * 10^{-3}$
b=3, K=30, T_{Gap}=30, l_s=1, l_q=3	ϵ=2.67	$\delta = 3.44 * 10^{-4}$
b=1, K=20, T_{Gap}=30, l_s=1, l_q=4	ϵ=22.00	$\delta = 6.79 * 10^{-4}$
b=2, K=30, T_{Gap}=30, l_s=1, l_q=4	ϵ=11.00	$\delta = 4.12 * 10^{-4}$
b=1, K=20, T_{Gap}=30, l_s=2, l_q=3	ϵ=16.00	$\delta = 2.46 * 10^{-5}$
b=2, K=30, T_{Gap}=30, l_s=2, l_q=3	ϵ=8.00	$\delta = 6.68 * 10^{-5}$

Table 4.5: Query suggestion results using different query logs.

Run	Precision@5	Recall@5	# of Evaluated Sessions
Original	0.0421	0.1402	**18,475**
KA	0.0693	0.2312	9,494
DP_C	0.1133	0.3891	4,144
DP_S	**0.1139**	**0.3911**	4,119

in order to achieve a good balance between privacy and utility. Many of the listed runs are acceptable to use. For instance, $DP_S(\epsilon = 8, \delta = 2.25 \times 10^{-7}$; T_{Gap}=30, l_s=1,l_q=3,b=1,k=20) is one of the good runs that I am using to support the following IR tasks.

4.4.2 Query suggestion

The query suggestion approach we proposed earlier is based on the calculation of similarities between query pairs which can be generated from Q'. In this section, we

use four different types of query logs as presented in section 4.4.1 to support query suggestion.

The effectiveness of a query suggestion approach is evaluated by comparison between the predicted ranked list of candidate queries and the ground truth. For each session in the test set, we perform query suggestion for each of the prefix query sequences and use the remaining queries of the session as the ground truth.

Table 4.5 presents Precision and Recall at the ranking position 5 for query suggestion, the number of sessions in the test set that can still be used by query suggestion, the number of the evaluated or remaining sessions. The major parameters for the anonymized query logs we used are:

- KA(K=20)
- DP_C(ϵ=8,δ=2.25*10^{-7};l=4,b=1,K=20)
- DP_S(ϵ=8,δ=2.25*10^{-7};T_{Gap}=30, l_s=1,l_q=3,b=1,K=20)

where all three anonymized runs share the same frequency threshold value $k = 20$ for a fair comparison.

As we can see, the number of test sessions that our algorithm successfully evaluated in DP_C and DP_S (4,144 and 4,119) are much fewer than in KA (9,494) and Original (18,475). Such information loss is inevitable for getting strong privacy protection. The two runs based on differential privacy successfully suggest queries for a similar amount of sessions. The KA run based on k-anonymity suggests queries for twice as

many sessions as the runs based on differential privacy would do. It is because k-anonymity doesn't limit the number of records from each individual while differential privacy does. Therefore, in terms of the quantity of the test sessions that could be evaluated, k-anonymity wins differential privacy, if both are constrained by the same cut-off threshold k.

Table 4.5 also presents the IR utility measures: Precision and Recall. DP_C and DP_S outperform the other runs. Especially, DP_S achieves the best utility results in both Precision and Recall. The KA run works less effectively in terms of IR utilities than DP_C and DP_S.

An interesting finding is that the number of evaluated sessions contradicts task-specific IR utility measures. That is, the runs releasing less sessions yield better IR utility scores. We think the underlying reason is rooted in the nature of IR. It is because (a) the records with the higher frequency (the more common ones) have a greater chance to be released by differential privacy; and (b) they are also records that are positively correlated to producing relevant results for an IR task because they reflect similar behaviors from many different users and are better and more effective data records. That is to say, although DP_C and DP_S release less data and suggest fewer sessions/queries, their released content happen to be more useful to IR. This result is very encouraging for us to advocate the use of actual IR utility metrics over data percentage.

Table 4.6: Session search results using different query logs.

Run	nDCG@10	MAP	# of Evaluated Sessions
Original	0.24792	0.19073	1,007,170
KA	0.19740	0.15363	782,873
DP_C	0.21875	0.17779	433,375
$\mathbf{DP}_S$	0.21879	0.17783	433,375

4.4.3 Session search

Although I use query suggestion as the major IR application to examine how the anonymized query log can be used to support session based applications, I implement additional experiments by applying the anonymized logs to support session search. I use the same three anonymized logs and the one original log as described earlier to apply to experiments on session search. The effectiveness of a session search approach is evaluated by comparing the generated ranked list of URLs for the session and the set of ground truth relevant URLs. In the AOL query log, we use the actual clicked URLs in the test session as the relevant document (ground truth) of the session.

Table 4.6 shows evaluation results for session search. The parameters for the anonymized query logs we used are the same with the 4 runs we used for query suggestion in Section 4.4.2. According to Table 4.6, the baseline run *Original* protects no privacy while achieving the best utility and # of evaluated session. The DP runs achieve nDCG10 and MAP scores better than the k-anonymity run. This means that the added noise in DP_C runs are acceptable and not hurting the IR utility.

However, the DP runs successfully evaluated fewer sessions compared with both the k-anonymity run and the un-anonymized run, which matches the tradeoff between privacy and data loss (not necessarily utility loss) as we observed in the query suggestion task.

We also observe that the DP_S run with anonymized session data outperforms the DP run without session data, which confirms the usefulness of the anonymized session data. In addition, by comparing all three anonymized runs with the baseline run using un-anonymized query log, we see a smaller amount of test sessions that could be evaluated while privacy level gets stronger from no privacy to k-anonymity then differential privacy.

4.4.4 Parameter settings

In Table 4.4, we compare the privacy level in DP_S with different parameter settings. By showing the typical runs and their parameter settings, we observe that ϵ is very sensitive to the max number of sessions per user l_s and the max number of queries per session l_q. Moreover, δ is also very sensitive to the noise scale b and the frequency threshold K.

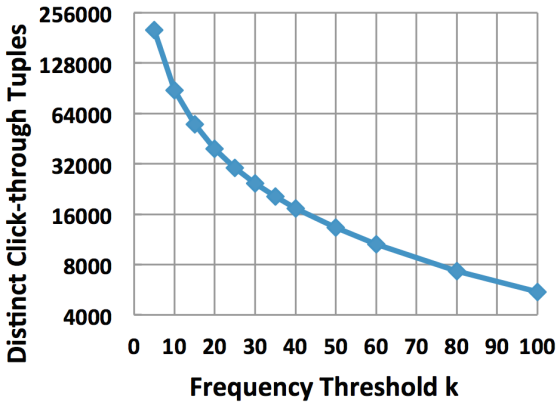
There are no hard rules for setting the parameters. Generally, smaller ϵ and δ values lead to stronger privacy guarantees, while δ value cannot be too large. This gives the query log owner, usually the commercial search companies, more flexibility to pick proper privacy parameter values in order to achieve a good balance between

privacy and utility. In Table 4.4, many listed runs are acceptable to use. For instance, $\mathrm{DP}_S(\epsilon = 8, \delta = 2.25 \times 10^{-7}$; T_{Gap}=30, l_s=1,l_q=3,b=1,k=20) is one of the good runs that we use in the experiments.

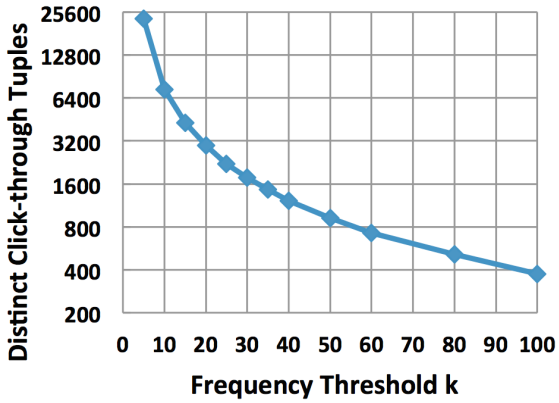
4.4.5 Privacy-utility tradeoff

In this section, we run further experiments to analyze the privacy-utility tradeoff during query log anonymization. It is important to show the consequences of using varying anonymization algorithms and using different parameter settings. The comparisons and suggestions we provide in this section should be able to help data owners make decisions when they need to anonymize a query log.

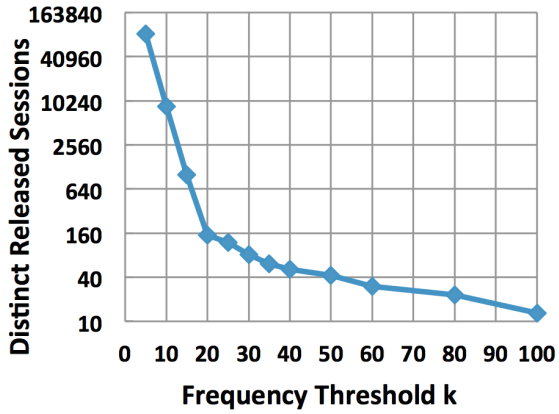
Figure 4.2 shows the scale of the anonymized data with varying frequency threshold K. Each data point in the figure corresponds to an anonymized query log. While changing the K values, we fix the other parameters in Figure 4.2 as T_{Gap}=30,l_s=1,l_q=3,$l = 4$, b=1.



(a) K-Anonymity (Click-throughs)



(b) Differential Privacy (Click-throughs)

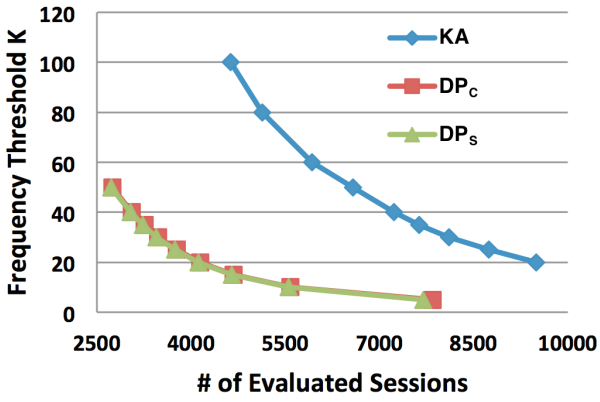


(c) Differential Privacy (Sessions)

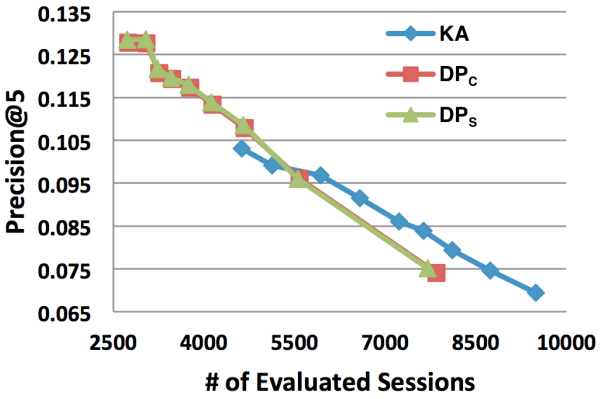
Figure 4.2: Data lost: Distinct click-through tuples and sessions after anonymization with varying frequency threshold k values.

Figure 4.2 (a) presents the change of distinct click-through tuples in query logs anonymized by KA; Figure 4.2 (b) presents the change of distinct click-through tuples in query logs anonymized by DP_C, while Figure 4.2 (c) presents the variation of different released sessions by DP_S. According to Figure 4.2, the scale of the anonymized query log is very sensitive to the frequency threshold parameter K. The anonymized log suffers a significant amount of data loss as K increases. Based on Eq. 5.17, the differential privacy parameters ϵ and δ (especially δ) decrease as K increases, which leads to even stronger privacy. Hence, the stronger privacy we require for differential privacy, the more data we lose in the anonymized query logs.

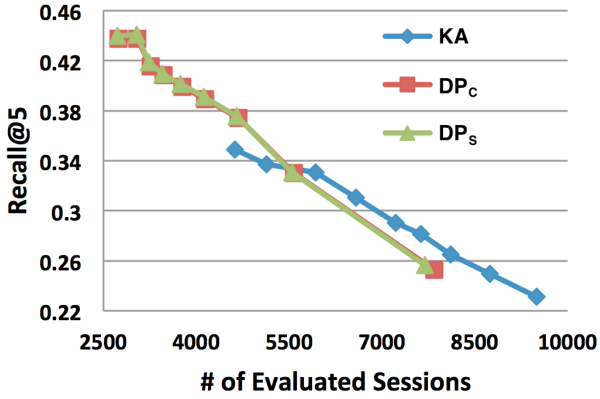
Figure 4.3 shows the relationships between the number of sessions being evaluated and (a) the frequency threshold K, (b) Precision@5 and (c) Recall@5 for the query suggestion task. Figure 4.3(a) reveals that if we want to evaluate a certain amount of sessions, the K used in differential privacy could be much less than the K used in k-anonymity. Figures 4.3(b) and (c) present the relationship between the application-based utility score and the amount of evaluated sessions. We observe that the differential privacy runs have better utility than the k-anonymity runs when the sessions being evaluated are no more than a certain value, around 5,500 in our case, while the k-anonymity runs may achieve even better utility than the differential privacy runs if there are more sessions evaluated.



(a) Frequency Threshold K



(b) Precision@5



(c) Recall@5

Figure 4.3: Query suggestion: Utility versus the number of evaluated sessions

Moreover, we observe from the analysis and experimental results that fewer input records from each user lead to stronger privacy. According to the mathematical char-

acteristics of (ϵ, δ)-differential privacy, the δ value is linearly related to the sensitivity value in our scenario. In other words, the fewer records we take from each user, the smaller δ value we can guarantee and thus achieve stronger privacy, and vice versa. For instance, if we double the number of input records per user l (or l_s), the anonymization mechanism will be with a doubled δ value. We therefore suggest including raw data from more users while limiting the number of sessions and click-throughs accepted from each user. The experiments based on different anonymization algorithms reveal a privacy-utility tradeoff. It seems that the balance between privacy and utility should be considered at the very beginning when we select the parameters for the anonymization methods.

4.5 Chapter Summary

Query log anonymization is challenging. It becomes even more challenging when search sessions are involved. In this chapter, I research how to release session data from query logs with differential privacy. I propose methods to evaluate the utility of the anonymized session data and generate experiments to support session-based IR application. The results show that our session-based query log anonymization algorithm not only satisfies differential privacy but also is sufficiently capable of supporting complex session-based IR applications such as query suggestion and session search.

To answer the research question of *how to release a query log containing session data*, I think there may not be a gold standard for this. Absolute privacy requires absolute removal of the data, while absolute utility requires absolute maintenance of the data. The best way to protect data depends on the goal of the protection, or over which metric we are optimizing. Practically, the most appropriate way to anonymize query logs depends on how we want to use them and what application are we aiming to support. We believe our differentially private approach is good enough to protect privacy during query log anonymization. Furthermore, experiments have shown that the query logs anonymized by our algorithm are very effective to support query suggestion, which answers the other research question of *how well can anonymized query logs containing search sessions be used to support query suggestion*.

Moreover, we observe from the analysis and experimental results that accepting fewer input records from each user lead to stronger privacy. According to the mathematical characteristics of (ϵ, δ)-differential privacy, the δ value is linearly related to the sensitivity value in our scenario. In other words, the fewer records we take from each user, the smaller δ value we can guarantee and thus achieves stronger privacy, and vice versa. For instance, if we double the number of input records per user l (or l_s), the anonymization mechanism will be with a doubled δ value. We hereby suggest including raw data from more users while limiting fewer sessions and click-throughs accepted from each individual user. Therefore, the tradeoff and balance between pri-

vacy and utility should be considered at the very beginning when we decide the amount of data as input.

Chapter 5

Proofs of Differential Privacy

This chapter provides proofs of differential privacy for the query log anonymization algorithms I proposed in the previous chapters. Chapter 3.2 has presented the preliminaries in differential privacy. According to the definition of differential privacy, a randomized query log anonymization algorithm satisfies (ϵ, δ)-differential privacy if and only if it can achieve Equa. 3.1. Therefore, the following proofs I provide for both algorithms are focusing on how to prove Equa. 3.1 in the corresponding scenario.

Chapter 5.1 presents proof of differential privacy for my query log anonymization algorithm A_{Click} for single queries presented in Chapter 3. Chapter 5.2 presents proof of differential privacy for my query log anonymization algorithm $A_{Session}$ for sessions presented in Chapter 4. Chapter 5.3 summarizes this chapter.

5.1 Proof of A_{Click}

I now present a general privacy proof sketch that analyzes the privacy guarantees of my approach in Algorithm A_{Click} (Chapter 3.4, query log anonymization for single queries). I prove that the algorithm satisfies $(\epsilon, 0)$-differential privacy as I presented in Chapter 3.4. Recall that K, q_f, c_f, b, b_q, b_c and b_t are parameters in our algorithm

as defined previously. Q is the original query log as input to the algorithm, Q_{clean} is the set of queries from Q that are possible options for release because they occur often enough while keeping at most q_f queries and c_f clicks from each user. Q_p is an externally generated stochastic query pool containing a large set of queries. Suppose each possible query q in the infinite domain has a probability of $p_g \in [0,1]$ to be included in the pool Q_p. In practice, the value of p_g depends on the source that is used to generate the query pool. Practically, major commercial search engines may create a large pool Q_p satisfying a p_g value close to 1.

Here is Theorem 1 as I presented it in Chapter 3:

Theorem 1: The query log anonymization algorithm presented in Algorithm A_{Click} satisfies ϵ-differential privacy, where ϵ is defined as:

$$\begin{aligned} \alpha &= Max\{\frac{e^{1/b}}{p_g}, 1 + \frac{1}{2e^{(K-1)/b} - 1}\} \\ \epsilon &= q_f \cdot ln(\alpha) + q_f/b_q + c_f/b_c \end{aligned} \tag{5.1}$$

In order to prove Theorem 1, first consider the following theorem:

Theorem 2: The generation of $q_{reduced}$ in Algorithm A_{Click} satisfies $(q_f \cdot ln(\alpha))$-differential privacy.

Theorem 2 is necessary for our algorithm to achieve ϵ-differential privacy rather than (ϵ,δ)-differential privacy. It makes our algorithm different from previous work [66] and helps us achieve stronger privacy guarantees. Being more specific, our Theorem 2 achieves stronger privacy guarantees than the *Select-Queries* procedure in [66], while the remainder of our algorithm has a similar structure to theirs in terms of

adding Laplacian noises. Theorem 1 is now a straightforward proof if we combine our Theorem 2 and Lemmas 2,3, and 4 as presented in [66].

I now prove Theorem 2, thereby showing why I achieve such a stronger privacy notion.

5.1.1 Proof of theorem 2

Proof. 1) We first consider the case in which q_f =1, K≥1. Q_1, Q_2 are two neighboring search logs, and Q_2 has one more user than Q_1, since $q_f = 1$. This means that Q_2 has one more query q^*. Also, we partition any set of query sets $\hat{Q}$ into two subsets: $\hat{Q}^+$, the query sets in $\hat{Q}$ that contain q^*, and $\hat{Q}^-$, the query sets in $\hat{Q}$ that do not contain q^*. The proof structure is similar to the Lemma 5 proof in Korolova et al. [66]. Here we only show why our algorithm has a stronger privacy guarantee than theirs. When $q^* \in Q_1$, we can prove that the algorithm satisfies $(1/b, 0)$-differential privacy using a similar idea as in Korolova et al. [66]. Here we give the derivatives in the case of $q^* \notin Q_1, q^* \in Q_2$. For all $q^* \in Q_p, q^* \notin Q$, we have $M(q, Q) = 0$, which is different from Korolova et al. [66]. Differential privacy requires the following two inequalities.

$$P[A(Q_1) \in \hat{Q}] \leq \alpha P[A(Q_2) \in \hat{Q}] + \delta \tag{5.2}$$

$$P[A(Q_2) \in \hat{Q}] \leq \alpha P[A(Q_1) \in \hat{Q}] + \delta \tag{5.3}$$

i). For inequality 5.2: First, we consider the case when q^* is not included in the output. Then $P[A(Q_2) \in \hat{Q}^-] = \mathrm{P}[q^* \text{ not released by } A(Q_2)] \cdot P[A(Q_1) \in \hat{Q}^-]$. Therefore,

$$\frac{P[A(Q_1) \in \hat{Q}^-]}{P[A(Q_2) \in \hat{Q}^-]} = \frac{1}{P[q^* \notin A(Q_2)]} = \frac{1}{1 - 0.5exp(\frac{1-K}{b})} \tag{5.4}$$

Next, we consider the other case when q^* is included in the output. Since $q^* \notin Q_1$, $A(Q_1) \in \hat{Q}^+$ will only be possible when q^* is generated from the query pool Q_p. Therefore,

$$\begin{aligned} & P[A(Q_1) \in \hat{Q}^+] \\ & = P[q^* \in Q_p] \cdot P[0 + Lap(b) \geq K] \cdot P[A(Q_1) \in (\hat{Q}^+ \setminus q^*)] \\ & = p_g \cdot 0.5exp(-\frac{K}{b}) \cdot P[A(Q_1) \in (\hat{Q}^+ \setminus q^*)] \end{aligned} \tag{5.5}$$

On the other hand, $q^* \in Q_2$, which means $A(Q_2) \in \hat{Q}^+$ requires q^* to be output from Q_2:

$$\begin{aligned} & P[A(Q_2) \in \hat{Q}^+] \\ & = P[1 + Lap(b) \geq K] \cdot P[A(Q_1) \in (\hat{Q}^+ \setminus q^*)] \\ & = 0.5exp(\frac{1-K}{b}) \cdot P[A(Q_1) \in (\hat{Q}^+ \setminus q^*)] \end{aligned} \tag{5.6}$$

Therefore, we achieve an upper bound such that:

$$
\begin{aligned}
&\frac{P[A(Q_1) \in \hat{Q}]}{P[A(Q_2) \in \hat{Q}]} = \frac{P[A(Q_1) \in \hat{Q}^+] + P[A(Q_1) \in \hat{Q}^-]}{P[A(Q_2) \in \hat{Q}^+] + P[A(Q_2) \in \hat{Q}^-]} \\
&\leq Max\{\frac{P[A(Q_1) \in \hat{Q}^+]}{P[A(Q_2) \in \hat{Q}^+]}, \frac{P[A(Q_1) \in \hat{Q}^-]}{P[A(Q_2) \in \hat{Q}^-]}\} \\
&= Max\{p_g \cdot \frac{exp(-\frac{K}{b})}{exp\frac{1-K}{b}}, \frac{1}{1 - 0.5exp(\frac{1-K}{b})}\} \\
&= \frac{1}{1 - 0.5exp(\frac{1-K}{b})}
\end{aligned}
\tag{5.7}
$$

The last step is because $p_g \in [0, 1)$, $exp(-\frac{1}{b}) \in (0, 1)$. Hence we get $p_g \cdot exp(-\frac{1}{b}) < 1 < \frac{1}{1-0.5exp(\frac{1-K}{b})}$. Therefore, inequality 5.2 holds for $\alpha = \frac{1}{1-0.5exp(\frac{1-K}{b})}$, $\delta = 0$

ii). For inequality 5.3:

Here I give a stronger upper bound in this case with the help of the query pool Q_p.

$$
\begin{aligned}
&\frac{P[A(Q_2) \in \hat{Q}]}{P[A(Q_1) \in \hat{Q}]} = \frac{P[A(Q_2) \in \hat{Q}^+] + P[A(Q_2) \in \hat{Q}^-]}{P[A(Q_1) \in \hat{Q}^+] + P[A(Q_1) \in \hat{Q}^-]} \\
&= \{0.5exp(\frac{1-K}{b}) \cdot P[A(Q_2) \in \{\hat{Q}^+ \setminus q^*\}] \\
&+ (1 - 0.5exp(\frac{1-K}{b})) \cdot P[A(Q_2) \in \hat{Q}^-]\} / \{0.5p_g \cdot \\
&exp(-\frac{K}{b}) \cdot P[A(Q_1) \in \{\hat{Q}^+ \setminus q^*\}] + P[A(Q_1) \in \hat{Q}^-]\} \\
&\leq Max\{\frac{0.5exp(\frac{1-K}{b})}{0.5p_g \cdot exp(-\frac{K}{b})}, 1 - 0.5exp(\frac{1-K}{b})\} \\
&= Max\{\frac{exp(1/b)}{p_g}, 1 - 0.5exp(\frac{1-K}{b})\} = \frac{exp(1/b)}{p_g}
\end{aligned}
\tag{5.8}
$$

Therefore, inequality 5.3 holds for $\alpha = \frac{e^{1/b}}{p_g}$, $\delta = 0$.

Combining the 2 cases, I conclude that our algorithm satisfies the $(ln(\alpha), 0)$-differential privacy, where:

$$\begin{aligned}\alpha &= Max\{e^{1/b}, \frac{1}{1-0.5exp(\frac{1-K}{b})}, \frac{e^{1/b}}{p_g}\} \\ &= Max\{\frac{e^{1/b}}{p_g}, 1+\frac{1}{2exp(\frac{K-1}{b})-1}\}\end{aligned} \tag{5.9}$$

which concludes the proof when $q_f = 1$.

2) Now we generalize the proof for cases when $q_f > 1$, which leads the approach from record level differential privacy to user level differential privacy. While our algorithm achieves differential privacy with $\delta = 0$, the generalization to situations with arbitrary q_f values becomes straightforward. Since Q_1 and Q_2 differ by one user, without loss of generality, suppose Q_2 contains one more user, i.e. it contains q_f additional queries at most, namely q_1, q_2, ...q_{q_f}. Then we have the following:

$$\begin{aligned}P[A(Q_1) \in \hat{Q}] &\leq \alpha \cdot P[A(Q_1 + q_1) \in \hat{Q}] \\ &\leq ... \leq \alpha^{q_f} \cdot P[A(Q_2) \in \hat{Q}]\end{aligned} \tag{5.10}$$

This concludes the proof of Theorem 2 that the generation of $Q_{reduced}$ is user level $(q_f \cdot ln(\alpha))$-differentially private.

5.2 Proof of $A_{Session}$

Now I present a general privacy proof sketch that analyzes the privacy guarantees for the session log anonymization part of my approach Algorithm $A_{Session}$ (Chapter 4.2, query log anonymization for sessions). I prove that the algorithm satisfies (ϵ, δ)-differential privacy as I presented in Chapter 4.2.

According to the definition of (ϵ, δ)-differential privacy, a query log anonymization algorithm A should satisfy the following two inequalities for all neighboring query logs Q_1 and Q_2 in order to achieve DP.

$$P[A(Q_1) \in \hat{Q}] \leq \alpha P[A(Q_2) \in \hat{Q}] + \delta \tag{5.11}$$

$$P[A(Q_2) \in \hat{Q}] \leq \alpha P[A(Q_1) \in \hat{Q}] + \delta \tag{5.12}$$

A major difference between our algorithm $A_{Session}$ and previous DP algorithms [40, 66, 135] is that we consider the query sequence of search sessions rather than individual queries or click-throughs as the unit to be counted and to be protected. However, a common mechanism is shared by all the work, which is achieving the requirements of (ϵ, δ)-differential privacy (Equa. 5.11 and 5.12) by adding i.i.d noise from the Laplace distribution on the data. A proof of the mechanism can be found in the Lemma 1 of Section 5.1 of Korolova et al. [66]. It shows that adding Laplacian noise would be able to achieve $(d \cdot ln(\alpha), \delta_{alg})$-differential privacy, where

$$\alpha = Max\{e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b} - 1}\} \tag{5.13}$$

$$\delta_{alg} = 0.5d \cdot exp(\frac{d - K}{b}) \tag{5.14}$$

Now let's evaluate the ϵ and δ values in our algorithm. In [66], d is the max number of queries per user and equals the sensitivity in their algorithm. That is to say, $d = \Delta f$ [66]. In our algorithm, the sensitivity Δf can be greater than the maximum of sessions per user l_s because the subsequences of original sessions also contribute to the sensitivity value. If we consider each of the Δf counts caused by

a user as an individual record, the privacy guarantee of Step 4 in $A_{Session}$ would be equivalent to $d = \Delta f$ as in Lemma 1. Hence Step 1 to Step 4 (to decide which session may be released) of $A_{Session}$ satisfies $(\Delta f \cdot ln(\alpha), \delta_{alg})$-differential privacy, where $\alpha = Max\{e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b}-1}\}$, $\delta_{alg} = 0.5\Delta f \cdot exp(\frac{\Delta f - K}{b})$. Step 5 of $A_{Session}$ is a standard procedure in differential privacy [30]. Such released session counts could be used to weigh query transitions in a session, which is helpful for the IR algorithms. This step itself achieves $(\Delta f/b, 0)$-differential privacy. Therefore, the overall $A_{Session}$ algorithm achieves $(\Delta f \cdot (ln(\alpha) + 1/b), \delta_{alg})$-differential privacy, while α and δ_{alg} are defined as earlier.

Next, we need to calculate the exact value of Δf in order to finalize the privacy level ϵ and δ. Sensitivity Δf is defined as the maximum difference of the statistics that could be made by the data generated from one user. In the worst case, the particular user differing between two neighboring datasets may issue l_s sessions, and each session may contain at most l_q queries. Given a session containing l_q queries, the amount of subsequences containing at least two queries is 2^{l_q} (total subsequences) -1 (the empty sequence) - l_q (subsequences containing only 1 query). Hence each input session with l_q queries contributes to a count by at most $2^{l_q} - 1 - l_q$ sessions. Therefore, the overall sensitivity of our algorithm is:

$$\Delta f = l_s \times (2^{l_q} - 1 - l_q) \tag{5.15}$$

By plugging in the sensitivity value Δf into

$$\begin{aligned} \epsilon &= \Delta f \cdot (ln(\alpha) + 1/b) \\ \delta &= 0.5\Delta f \cdot exp(\frac{\Delta f - K}{b}) \end{aligned} \tag{5.16}$$

, our session release approach can achieve (ϵ, δ)-differential privacy while ϵ and δ are:

$$\begin{aligned} \epsilon &= l_s(2^{l_q} - 1 - l_q) \cdot (ln(Max\{e^{1/b}, 1 + \frac{1}{2e^{(K-1)/b} - 1}\}) + 1/b) \\ \delta &= 0.5 l_s(2^{l_q} - 1 - l_q) \cdot exp(\frac{l_s(2^{l_q} - 1 - l_q) - K}{b}) \end{aligned} \tag{5.17}$$

5.3 Summary

In this chapter, I present proofs of differential privacy for the algorithms A_{Click} as I presented in Chapter 3 (query log anonymization for single queries) and $A_{Session}$ as I presented in Chapter 4 (query log anonymization for sessions). The A_{Click} algorithm achieves $(\epsilon, 0)$-differential privacy while the algorithm $A_{Session}$ achieves (ϵ, δ)-differential privacy.

Chapter 6

Conclusion

This chapter concludes my dissertation. Chapter 6.1 summarizzes my research in this thesis. Chapter 6.2 offers discussions of some interesting insights of my research. Chapter 6.3 presents the potential future work of this thesis. Chapter 6.4 concludes the chapter with the impact of this work.

6.1 Research Summary

Query log anonymization is an important and challenging task which gets even more difficult when session data is involved. In general, the ultimate goal of the research work I present in this dissertation is about the tradeoff and balance between privacy and utility of web search query logs.

In this dissertation, I first introduce a framework for anonymizing query logs and evaluating their web search utility. I make use of differential privacy, which is a strong privacy notation, to anonymize the logs. More importantly, I use real IR application to measure the utility of the anonymized query log. The experiments show that the anonymized query log can be well used to support ad-hoc search. The web search algorithms using anonymized logs do not perform significantly differently from those

using the original logs. Since high-level privacy has been guaranteed by the mechanism of differential privacy, I suggest that search engine companies use less strict parameter settings to maintain the high utility of the anonymized logs.

Furthermore, I expand the work to involve search session data. I propose a query log anonymization mechanism with differential privacy that can maintain both click-through data and session data in the anonymized query log. Then, I use typical session-based IR applications, which are query suggestion and session search, to evaluate the utility of the anonymized session log. Experiments show that the session log anonymized with differential privacy can be well used to support these IR applications.

In summary, by developing a proper framework for query log anonymization with the systematic analysis of both privacy and utility, this work makes an important step towards the final solution of web query log anonymization. All these research work contribute to my dissertation by proposing an analysis of the tradeoff between adequate privacy and sufficient utility of online user data in different scenarios.

6.2 Discussions

In this section, I would like to discuss about some major concerns in my work.

Weakness of Differential Privacy. Theoretically, differential privacy is defined as a strong standard [28, 33] which requires "unconditional privacy guarantees against computationally unbounded adversaries" [41]. Although differential privacy is very

powerful, the theory still requires the parameters to be carefully treated. Here I would like to make a brief discussion about the weakness of differential privacy and how it is inherited in my work.

Differential privacy is best for low-sensitivity scenarios [28]. This is a natural inference from the principle of differential privacy. Specifically, the noise scale b used to achieve differential privacy is usually proportional to $\frac{\Delta f}{\epsilon}$ [28, 39, 66, 135], which is the ratio between the sensitivity value Δf and the privacy scale ϵ. Under certain privacy requirements of fixed ϵ value, the corresponding noise scale will be proportional to the sensitivity value Δf. In the case of high-sensitivity scenarios, we may have to add a lot of noise to the results to achieve the same level of differential privacy, while hurting the data accuracy and utility. Therefore, it is better to keep the sensitivity value relatively low. Practically, I implement a variety of experiments to analyze the relationship between privacy and utility by adjusting parameters dominating sensitivity (c_f and q_f) and the noise scale. I managed to find a good balance between privacy and utility as well as keeping the sensitivity low.

Interactivity and non-interactivity. Another weakness of differential privacy may be exposed in the interactive scenario with multiple release of the same data. This is equivalent to the case in a database when we allow adaptive querying to the same database. If we want to achieve differential privacy in such an interactive setting allowing multiple releases (or database queries), we must inject the noise multiple times [28]. And most importantly, we need to know the value of x up front, which

may not be available in many cases. Therefore, my data release mechanism uses the non-interactive setting [69] which allows only one released version of the same dataset. Actually, all of the existing differential privacy work on query log anonymization uses the non-interactive version to control the noise scale and avoid such weakness in interactive differential privacy. However, even non-interactive differential privacy is not perfect. Kifer and Machanavajjhala [65] reveal that differential privacy actually really works when the individuals are truly independent from each other. Practically, this assumption can be true in the scenario of query log anonymization since every search engine user generates their own search log. In summary, differential privacy is very powerful. Although there are some weaknesses in differential privacy that need to be taken care of, I have addressed them in this work to minimize the influences of the weakness of the mechanism.

Privacy Parameters. The ϵ and δ values in differential privacy are essential parameters that quantifies the privacy level of the query log anonymization mechanisms. However, there are no fixed rules for how to determine the best values for ϵ or δ in an algorithm [28]. Smaller values of the parameters leading to stronger privacy. However, the selection of parameter values is task dependent. For instance, for an anonymization algorithm satisfying $\epsilon = 1.0$-differential privacy, the probability ratio of getting a certain output from two neighboring logs will be constrained in a $e^{1.0}$ range, which is an acceptable privacy level in the scenario of query log anonymization.

Session Log Anonymization. Session log anonymization by differential privacy is challenging. One key obstacle in session log anonymization is that there are too many potential combinations of search queries to form search sessions. Such sparsity in frequency distribution makes the session log anonymization mechanism hard to design. Actually, we may be able to propose new methods to organize and represent the search sessions in order to reduce the sparsity in session logs. For instance, if the anonymized session log will be used to support research about topic drifts during a search session, we may simplify the task with a query classifier. With a query classifier that allocates each search query into keywords or a topic in the knowledge graph, we can generate a mapping from an infinite domain (natural language queries) into a finite domain of topics. Hence, a search session can be represented as a search path and transitions among finite topics. If such topic paths (rather than exact contents of search queries) are enough to support the corresponding applications as anonymization output, more research and mechanisms from the frequent sequence mining domain may be able to apply to query/session log anonymization since they are also working on the finite domain data. I wish my work can inspire more follow-up research on session log anonymization.

6.3 Future Work

As I have mentioned in the introduction, the task of query log anonymization should contribute to the greater goal of better generate, distribute and use of data.

Query log anonymization is also valuable to support web mining tasks, which could be a direction of my future work. In this section, I give a discussion of web mining tasks that can be supported by the anonymized query logs and show some preliminary results for the task of website clustering.

Query logs are used for some different web mining tasks. While testing the effectiveness of using an anonymized query log for this different task is outside the scope of this chapter, I conduct a preliminary analysis on a simple web page clustering task. The goal of this task is to group n websites $W = \{w_1 ... w_n\}$ using query log click information from similar queries. I now sketch how to accomplish this and show that I get meaningful clusters using the anonymized query log Q'. I pause to mention that I purposefully do not conduct a complete analysis of this task. Instead, I demonstrate that web mining utility is still possible using Q' and that future work should explore more tasks from these anonymized query logs.

I consider a simple single-link hierarchical clustering algorithm [44] to cluster the websites (URLs) in Q' using the generated query-click graph. In the query-click graph, for each query q and website w, I define $C(q, w)$ as the number of clicks from query q to website w. I also define $C(w)$ to be the total number of clicks to website w. $QSet(w)$ is defined as the set of queries leading to clicks on website w. Then, for every website pair w_1 and w_2, I define their similarities as the ratio of clicks from common queries:

$$Sim(w_1, w_2) = \min\{\frac{\sum_{q \in QS(w_1,w_2)} C(q, w_1)}{C(w_1)}, \frac{\sum_{q \in QS(w_1,w_2)} C(q, w_2)}{C(w_2)}\} \tag{6.1}$$

where $QS(w_1, w_2) = QSet(w_1) \bigcap QSet(w_2)$. Once I compute the similarities, I use single-linkage clustering to cluster the websites. I empirically tested different hierarchical clustering methods and chose single link clustering for this task because of the sparsity of the query log click graph. The parameter settings for generating Q' were: query count threshold $K = 100$, query limit and click limit per user $q_f = c_f = 100$, all noise scales $b_x = 10$.

To evaluate the quality of the clusters, I clustered 1000 websites. The hierarchical clustering algorithm merges two clusters when the websites have a high similarity score: $Sim(w_1, w_2) \geq 0.5$. I hand labelled 10 clusters with class labels (Table 6.1 shows an example) and then measured *purity*. To compute purity, I used the class label that is most frequent in the cluster and assumed that to be the correct class label. The accuracy is the number of correctly assigned websites w_i divided by the total number of websites $|W|$. For the 10 clusters I hand labeled, there were a total of 59 websites in them. The purity was 0.76. This means that there were some websites that were put into the wrong clusters, but the majority were not. In other words, a meaningful structure for web mining can still be extracted from anonymized logs.

Table 6.1: Some clustering results based on the anonymized query log.

(Cluster of Lyrics)	(Cluster of Lottery)
lyrics.astraweb.com	lottery.yahoo.com
www.lyricsfreak.com	www.flottery.com
www.lyrics.com	www.lotteryusa.com
www.azlyrics.com	www.flalottery.com
www.sing365.com	(Cluster of Banks)
www.musicsonglyrics.com	www.bankone.com
www.lyricsdownload.com	www.chase.com

This work shows that the differentially private query log can well support typical mining tasks such as website clustering. As future work, I will research how query log anonymization could be better utilized to help research in more related domains.

6.4 Resources

In this section, I provide some additional resources related to this thesis.

- I gave a Tutorial "Differential Privacy for Information Retrieval" [123] with my advisor professor Grace Hui Yang in the 3rd ACM International Conference on the Theory of Information Retrieval (ICTIR 2017), Amsterdam, Netherlands. Oct 1, 2017. I will give an updated version of this Tutorial in the 11th ACM International Conference on Web Search and Data Mining (WSDM 2018), Los Angeles, USA. Feb 5, 2018.

- As a student organizer, I contributed to the first, second, and third "Privacy-Preserving IR: When Information Retrieval Meets Privacy and Security" Work-

shops PPIR'14 [96], PPIR'15 [122] and PPIR'16 [125] during the 37-39th International ACM SIGIR Conferences (SIGIR 2014 - 2016).

- Our website of detailed related resources in privacy-preserving Information Retrieval is located at *https://privacypreservingir.org/*. It includes our publications in privacy-preserving Information Retrieval and more information about the tutorial in ICTIR 2017 and workshops in SIGIR 2014 - 2016.

6.5 Impact

Concern with privacy is not a new issue. It has been developing since human being could see and hear (collect data), write and remember (store data), think and understand (analyze data), or talk and trade (distribute data). The fact is, the rapid growth of computer science and informatics technologies in recent years, especially the spread of online services such as search engines and social network services, has fundamentally changed how we collect data, store data, analyze data, and distribute data. That's the reason for the rising privacy concern in our society nowadays. It also shows the importance of my research and how much impact we may achieve if we can contribute to resolving the massive privacy concern in the era of big data.

After the inappropriate data release from AOL in 2006, search engine companies became much more serious about the privacy of their web search data. They have been hesitating to release web search query logs to third-party researchers to avoid privacy risks, which leads to a significant shortage of real search engine query logs in

the research community. In this dissertation, I focus on query log anonymization by differential privacy to address such privacy concerns with query logs. In my research, I propose privacy-preserving mechanisms to reduce the privacy risks by anonymizing query logs and help web users to understand the potential privacy risks in online services. I use typical real IR applications, including ad-hoc retrieval, query suggestion, and session search, to quantify the utility of the anonymized query log to evaluate how useful the query log anonymization mechanisms are.

I hope the research can encourage web search engine companies to release query logs for research purposes, inspire our IR community to develop better approaches to address the rising privacy concerns in multiple IR scenarios and applications and help web users in general to gain better understanding of the potential privacy risks of their online behavior.

Furthermore, I hope this work can inspire the IR community to use more advanced techniques such as differential privacy from other research fields. As I have mentioned in the introduction and related work, although there have been a few implementations of differential privacy in data mining research, the use of differential privacy in IR research has just started. Recent research including this work has shown that differential privacy can be well used to address privacy concerns in some IR research topics such as query log anonymization and location data privacy. I hope this work will inspire more researchers in the IR community to investigate what other research

problems may benefit from differential privacy or other advanced mechanisms from other research domains.

In summary, by resolving the privacy concerns from the view of both data owners (search engine companies) and data providers (web service users), my work aims to address the privacy concern and contribute to the improvement of how we manage information. I hope this work can not only benefit the research in this particular task of query log anonymization but also can inspire more research in privacy-preserving Information Retrieval and related data-driven domains.

Bibliography

[1] Eytan Adar. User 4xxxxx9: Anonymizing query logs. In *Proceeding of Query Log Analysis Workshop, International Conference on World Wide Web*, 2007.

[2] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 19–26, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148177. URL `http://doi.acm.org/10.1145/1148170.1148177`.

[3] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Query recommendation using query logs in search engines. In *International Conference on Extending Database Technology*, pages 588–596. Springer, 2004.

[4] Michael Barbaro and Tom Zeller. A face is exposed for AOL searcher no. 4417749. In *New York Times*, Aug 2006.

[5] John Bateman and Michael Zock. Natural language generation. In *The Oxford handbook of computational linguistics*. 2003.

[6] Doug Beeferman and Adam Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 407–416, New York, NY, USA, 2000. ACM. ISBN 1-58113-233-6. doi: 10.1145/347090.347176. URL http://doi.acm.org/10.1145/347090.347176.

[7] Raghav Bhaskar, Srivatsan Laxman, Adam Smith, and Abhradeep Thakurta. Discovering frequent patterns in sensitive data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 503–512, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835869. URL http://doi.acm.org/10.1145/1835804.1835869.

[8] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. Query suggestions in the absence of query logs. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 795–804, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010023. URL http://doi.acm.org/10.1145/2009916.2010023.

[9] Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, and Sebastiano Vigna. Query suggestions using query-flow graphs. In *Proceedings of the 2009 Workshop on Web Search Click Data*, WSCD '09, pages 56–63, New York, NY,

USA, 2009. ACM. ISBN 978-1-60558-434-8. doi: 10.1145/1507509.1507518. URL http://doi.acm.org/10.1145/1507509.1507518.

[10] Luca Bonomi and Li Xiong. A two-phase algorithm for mining sequential patterns with differential privacy. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 269–278, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505553. URL http://doi.acm.org/10.1145/2505515.2505553.

[11] Fei Cai, Shangsong Liang, and Maarten de Rijke. Personalized document re-ranking based on bayesian probabilistic matrix factorization. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 835–838, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2257-7. doi: 10.1145/2600428.2609453. URL http://doi.acm.org/10.1145/2600428.2609453.

[12] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 243–250, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390377. URL http://doi.acm.org/10.1145/1390334.1390377.

[13] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 875–883, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401995. URL `http://doi.acm.org/10.1145/1401890.1401995`.

[14] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong. Quantifying differential privacy under temporal correlations. In *2017 IEEE 33rd International Conference on Data Engineering, ICDE'17*, pages 821–832, April 2017. doi: 10.1109/ICDE.2017.132.

[15] Claudio Carpineto and Giovanni Romano. Semantic search log k-anonymization with generalized k-cores of query concept graph. In *Advances in Information Retrieval: 35th European Conference on IR Research*, ECIR'13, pages 110–121, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-36972-8. doi: 10.1007/978-3-642-36973-5_10. URL `http://dx.doi.org/10.1007/978-3-642-36973-5_10`.

[16] Ben Carterette, Evangelos Kanoulas, and Emine Yilmaz. Simulating simple user behavior for system effectiveness evaluation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 611–620, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0717-8.

doi: 10.1145/2063576.2063668. URL http://doi.acm.org/10.1145/2063576.2063668.

[17] Wendy W Chapman, Prakash M Nadkarni, Lynette Hirschman, Leonard W D'avolio, Guergana K Savova, and Ozlem Uzuner. Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions, 2011.

[18] Rui Chen, Gergely Acs, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 638–649, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1651-4. doi: 10.1145/2382196.2382263. URL http://doi.acm.org/10.1145/2382196.2382263.

[19] Xiang Cheng, Sen Su, Shengzhi Xu, Peng Tang, and Zhengyi Li. Differentially private maximal frequent sequence mining. *Comput. Secur.*, 55(C):175–192, November 2015. ISSN 0167-4048. doi: 10.1016/j.cose.2015.08.005. URL http://dx.doi.org/10.1016/j.cose.2015.08.005.

[20] C. Clifton and T. Tassa. On syntactic anonymity and differential privacy. In *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, pages 88–93, April 2013. doi: 10.1109/ICDEW.2013.6547433.

[21] Alissa Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. *ACM Trans. Web*, 2(4):19:1–19:27, October 2008. ISSN 1559-1131.

[22] Graham Cormode. Building blocks of privacy: Differentially private mechanisms tutorial talk. *Privacy Preserving Data Publication and Analysis (PrivDB) workshop*, 2013.

[23] Mark Cramer, Mike Wertheim, and David Hardtke. Demonstration of Improved Search Result Relevancy Using Real-Time Implicit Relevance Feedback. In *Understanding the user - workshop in conjuction with SIGIR'09*, 2009.

[24] Nick Craswell and Martin Szummer. Random walks on the click graph. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 239–246, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277784. URL `http://doi.acm.org/10.1145/1277741.1277784`.

[25] Mark Davies. N-grams data from the corpus of contemporary american english (coca). *Downloaded from http://www.ngrams.info*, 23:2012, 2011.

[26] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 202–210, New York, NY, USA, 2003. ACM. ISBN 1-58113-670-6. doi: 10.1145/773153.773173. URL `http://doi.acm.org/10.1145/773153.773173`.

[27] Abdigani Diriye, Ryen White, Georg Buscher, and Susan Dumais. Leaving so soon?: Understanding and predicting web search abandonment rationales. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1025–1034, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398399. URL `http://doi.acm.org/10.1145/2396761.2398399`.

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

[29] Cynthia Dwork. The differential privacy frontier. In *Theory of Cryptography Conference*, pages 496–502. Springer, 2009.

[30] Cynthia Dwork. *Differential Privacy*, pages 338–340. Springer US, Boston, MA, 2011. ISBN 978-1-4419-5906-5.

[31] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009.

[32] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. ISSN 1551-305X. doi: 10.1561/0400000042.

[33] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third*

Conference on Theory of Cryptography, TCC'06. ISBN 3-540-32731-2, 978-3-540-32731-8. doi: 10.1007/11681878_14. URL http://dx.doi.org/10.1007/11681878_14.

[34] Liyue Fan, Luca Bonomi, Li Xiong, and Vaidy Sunderam. Monitoring web browsing behavior with differential privacy. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 177–188, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. doi: 10.1145/2566486.2568038. URL http://doi.acm.org/10.1145/2566486.2568038.

[35] Henry Feild and James Allan. Task-aware query recommendation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 83–92, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484069. URL http://doi.acm.org/10.1145/2484028.2484069.

[36] Henry Allen Feild, James Allan, and Joshua Glatt. Crowdlogging: Distributed, private, and anonymous search logging. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 375–384, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2009969. URL http://doi.acm.org/10.1145/2009916.2009969.

[37] Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 493–502, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835868. URL http://doi.acm.org/10.1145/1835804.1835868.

[38] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. Exploiting innocuous activity for correlating users across sites. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 447–458, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488428. URL http://doi.acm.org/10.1145/2488388.2488428.

[39] Michaela Gotz, Ashwin Machanavajjhala, Guozhang Wang, Xiaokui Xiao, and Johannes Gehrke. Publishing search logs – a comparative study of privacy guarantees. *IEEE Trans. on Knowl. and Data Eng.*, 24(3), March 2012. ISSN 1041-4347.

[40] Michaela Gotz, Ashwin Machanavajjhala, Guozhang Wang, Xiaokui Xiao, and Johannes Gehrke. Publishing search logs:a comparative study of privacy guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 24(3), 2012.

[41] Adam Groce, Jonathan Katz, and Arkady Yerukhimovich. *Limits of Computational Differential Privacy in the Client/Server Setting*, pages 417–431.

Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-19571-6. doi: 10.1007/978-3-642-19571-6_25. URL http://dx.doi.org/10.1007/978-3-642-19571-6_25.

[42] Dongyi Guan, Hui Yang, and Nazli Goharian. Effective structured query formulation for session search. In *TREC '12*.

[43] Dongyi Guan, Sicong Zhang, and Hui Yang. Utilizing query change for session search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 453–462, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484055. URL http://doi.acm.org/10.1145/2484028.2484055.

[44] Gan Guojun, Ma Chaoqun, and Wu Jianhong. Data clustering: theory, algorithms, and applications. *ASA-SIAM Series on Statistics and Applied Probability*, 2007.

[45] Morgan Harvey, Claudia Hauff, and David Elsweiler. Learning by example: Training users with high-quality query suggestions. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 133–142, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767731. URL http://doi.acm.org/10.1145/2766462.2767731.

[46] Michael Hay, Kun Liu, Gerome Miklau, Jian Pei, and Evimaria Terzi. Privacy-aware data management in information networks. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 1201–1204, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0661-4. doi: 10.1145/1989323.1989453. URL http://doi.acm.org/10.1145/1989323.1989453.

[47] Q. He, D. Jiang, Z. Liao, S. C. H. Hoi, K. Chang, E. P. Lim, and H. Li. Web query recommendation via sequential query prediction. In *2009 IEEE 25th International Conference on Data Engineering*, pages 1443–1454, March 2009. doi: 10.1109/ICDE.2009.71.

[48] Yuan Hong, Xiaoyun He, Jaideep Vaidya, Nabil Adam, and Vijayalakshmi Atluri. Effective anonymization of query logs. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1465–1468, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646146. URL http://doi.acm.org/10.1145/1645953.1646146.

[49] Mathias Humbert, Théophile Studer, Matthias Grossglauser, and Jean-Pierre Hubaux. *Nowhere to Hide: Navigating around Privacy in Online Social Networks*, pages 682–699. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

ISBN 978-3-642-40203-6. doi: 10.1007/978-3-642-40203-6_38. URL http://dx.doi.org/10.1007/978-3-642-40203-6_38.

[50] Tereza Iofciu, Peter Fankhauser, Fabian Abel, and Kerstin Bischoff. Identifying users across social tagging systems. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, ICWSM'11*, 2011.

[51] D. Irani, S. Webb, K. Li, and C. Pu. Large online social footprints–an emerging threat. In *2009 International Conference on Computational Science and Engineering*, volume 3, pages 271–276, Aug 2009. doi: 10.1109/CSE.2009.459.

[52] Paridhi Jain, Ponnurangam Kumaraguru, and Anupam Joshi. @i seek 'fb.me': Identifying users across multiple online social networks. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, pages 1259–1268, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2038-2. doi: 10.1145/2487788.2488160. URL http://doi.acm.org/10.1145/2487788.2488160.

[53] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188.

[54] Jiepu Jiang and Daqing He. Different effects of click-through and past queries on whole-session search performance. In *TREC '13*, 2013.

[55] Jiepu Jiang, Shuguang Han, Jia Wu, and Daqing He. Pitt at trec 2011 session track. In *TREC '11*, 2011.

[56] Jiepu Jiang, Daqing He, and Shuguang Han. Pitt at trec 2012 session track. In *TREC '12*, 2012.

[57] Xiaoran Jin, Marc Sloan, and Jun Wang. Interactive exploratory search for multi page search results. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 655–666, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488446. URL `http://doi.acm.org/10.1145/2488388.2488446`.

[58] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceeding of the 14th International Conference on Machine Learning, ICML'97*, 1997.

[59] Rosie Jones and Kristina Lisa Klinkner. Beyond the session timeout: Automatic hierarchical segmentation of search topics in query logs. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 699–708, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458176. URL `http://doi.acm.org/10.1145/1458082.1458176`.

[60] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. "i know what you did last summer": Query logs and user privacy. In *Proceedings of the Sixteenth*

ACM Conference on Conference on Information and Knowledge Management, CIKM '07, pages 909–914, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-803-9. doi: 10.1145/1321440.1321573. URL http://doi.acm.org/10.1145/1321440.1321573.

[61] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. Vanity fair: Privacy in querylog bundles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 853–862, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458195. URL http://doi.acm.org/10.1145/1458082.1458195.

[62] Evangelos Kanoulas, Ben Carterette, Mark Hall, Paul Clough, and Mark Sanderson. Overview of the trec 2012 session track. In *TREC'12*, 2012.

[63] Evangelos Kanoulas, Ben Carterette, Mark Hall, Paul Clough, and Mark Sanderson. Overview of the trec 2013 session track. In *TREC'13*, 2013.

[64] Makoto P. Kato, Tetsuya Sakai, and Katsumi Tanaka. Structured query suggestion for specialization and parallel movement: Effect on search behaviors. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 389–398, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187890. URL http://doi.acm.org/10.1145/2187836.2187890.

[65] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 193–204, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0661-4. doi: 10.1145/1989323.1989345. URL `http://doi.acm.org/10.1145/1989323.1989345`.

[66] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 171–180, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526733. URL `http://doi.acm.org/10.1145/1526709.1526733`.

[67] Alexander Kotov, Paul N. Bennett, Ryen W. White, Susan T. Dumais, and Jaime Teevan. Modeling and analysis of cross-session search tasks. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 5–14, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2009922. URL `http://doi.acm.org/10.1145/2009916.2009922`.

[68] Jaewoo Lee and Christopher W. Clifton. Top-k frequent itemsets via differentially private fp-trees. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 931–940, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9. doi: 10.1145/

2623330.2623723. URL http://doi.acm.org/10.1145/2623330.2623723.

[69] David Leoni. Non-interactive differential privacy: A survey. In *Proceedings of the First International Workshop on Open Data*, WOD '12, pages 40–52, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1404-6. doi: 10.1145/2422604.2422611. URL http://doi.acm.org/10.1145/2422604.2422611.

[70] Haoran Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. In *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014.*, pages 475–486, 2014. doi: 10.5441/002/edbt.2014.43. URL http://dx.doi.org/10.5441/002/edbt.2014.43.

[71] Haoran Li, Li Xiong, Xiaoqian Jiang, and Jinfei Liu. Differentially private histogram publication for dynamic datasets: An adaptive sampling approach. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1001–1010, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806441. URL http://doi.acm.org/10.1145/2806416.2806441.

[72] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering, ICDE'07*, pages 106–115, April 2007. doi: 10.1109/ICDE.2007.367856.

[73] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. Overview of the trec-2014 microblog track. Technical report, DTIC Document, 2014.

[74] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. What's in a name?: An unsupervised approach to link users across communities. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 495–504, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1869-3. doi: 10.1145/2433396.2433457. URL `http://doi.acm.org/10.1145/2433396.2433457`.

[75] Jingjing Liu and Nicholas J. Belkin. Personalizing information retrieval for multi-session tasks: The roles of task stage and task type. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 26–33, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835457. URL `http://doi.acm.org/10.1145/1835449.1835457`.

[76] Kun Liu, Gerome Miklau, J Pei, and E Terzi. Privacy-aware data mining in information networks. *Tutorial in KDD'10*, 2010.

[77] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 51–62, New York, NY, USA,

2014. ACM. ISBN 978-1-4503-2376-5. doi: 10.1145/2588555.2588559. URL http://doi.acm.org/10.1145/2588555.2588559.

[78] Jiyun Luo, Sicong Zhang, and Hui Yang. Win-win search: Dual-agent stochastic game in session search. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 587–596, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2257-7. doi: 10.1145/2600428.2609629. URL http://doi.acm.org/10.1145/2600428.2609629.

[79] Jiyun Luo, Sicong Zhang, Xuchu Dong, and Hui Yang. *Designing States, Actions, and Rewards for Using POMDP in Session Search. Advances in Information Retrieval: 37th European Conference on IR Research, ECIR'15*, pages 526–537. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16354-3. doi: 10.1007/978-3-319-16354-3_58. URL http://dx.doi.org/10.1007/978-3-319-16354-3_58.

[80] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007. ISSN 1556-4681.

[81] Ashwin Machanavajjhala, Xi He, and Michael Hay. Differential privacy in the wild: A tutorial on current practices and open challenges. *Proc. VLDB Endow.*,

9(13):1611–1614, September 2016. ISSN 2150-8097. doi: 10.14778/3007263.3007322. URL http://dx.doi.org/10.14778/3007263.3007322.

[82] A. Malhotra, L. Totti, W. Meira Jr., P. Kumaraguru, and V. Almeida. Studying user footprints in different online social networks. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1065–1070, Aug 2012. doi: 10.1109/ASONAM.2012.184.

[83] Kobbi Nissim and Uri Stemmer. On the generalization properties of differential privacy. *CoRR, abs/1504.05800*, 2015.

[84] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International Conference on World Wide Web, WWW '98*, pages 161–172, 1998. URL citeseer.nj.nec.com/page98pagerank.html.

[85] Filip Radlinski and Thorsten Joachims. Query chains: Learning to rank from implicit feedback. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 239–248, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X. doi: 10.1145/1081870.1081899. URL http://doi.acm.org/10.1145/1081870.1081899.

[86] A. Ramachandran, L. Singh, E. Porter, and F. Nagle. Exploring re-identification risks in public domains. In *2012 Tenth Annual International Conference on*

Privacy, Security and Trust, pages 35–42, July 2012. doi: 10.1109/PST.2012.6297917.

[87] Karthik Raman, Paul N. Bennett, and Kevyn Collins-Thompson. Toward whole-session relevance: Exploring intrinsic diversity in web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 463–472, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484089. URL http://doi.acm.org/10.1145/2484028.2484089.

[88] S. Robertson. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. URL http://scholar.google.de/scholar.bib?q=info:U4l9kCVIssAJ:scholar.google.com/&output=citation&hl=de&as_sdt=2000&as_vis=1&ct=citation&cd=1.

[89] Stephen Robertson. A new interpretation of average precision. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 689–690, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390453. URL http://doi.acm.org/10.1145/1390334.1390453.

[90] Ian Ruthven. Interactive information retrieval. *Annual review of information science and technology, ARIST*, 42(1):43–91, 2008.

[91] Anand D Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, 30(5):86–94, 2013.

[92] Cyrus Shahabi, Liyue Fan, Luciano Nocera, Li Xiong, and Ming Li. Privacy-preserving inference of social relationships from location data: A vision paper. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15, pages 9:1–9:4, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3967-4. doi: 10.1145/2820783.2820880. URL `http://doi.acm.org/10.1145/2820783.2820880`.

[93] Entong Shen and Ting Yu. Mining frequent graph patterns with differential privacy. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 545–553, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487601. URL `http://doi.acm.org/10.1145/2487575.2487601`.

[94] Prajakta Shinde and Pranjali Joshi. Survey of various query suggestion system. *International Journal Of Engineering and Computer Science*, 3(12):9576–9580, 2014.

[95] Milad Shokouhi, Ryen W. White, Paul Bennett, and Filip Radlinski. Fighting search engine amnesia: Reranking repeated results. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development*

in Information Retrieval, SIGIR '13, pages 273–282, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484075. URL http://doi.acm.org/10.1145/2484028.2484075.

[96] 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, December 2014. ISSN 0163-5840. doi: 10.1145/2701583.2701593. URL http://doi.acm.org/10.1145/2701583.2701593.

[97] L. Singh, G. H. Yang, M. Sherr, A. Hian-Cheong, K. Tian, J. Zhu, and S. Zhang. Public information exposure detection: Helping users understand their web footprints. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM'15)*, pages 153–161, Aug 2015. doi: 10.1145/2808797.2809280.

[98] Lisa Singh, Hui Yang, Micah Sherr, Yifang Wei, Andrew Hian-Cheong, Kevin Tian, Janet Zhu, Sicong Zhang, Tavish Vaidya, and Elchin Asgarli. Helping users understand their web footprints. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 117–118, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3473-0. doi: 10.1145/2740908.2742763. URL http://doi.acm.org/10.1145/2740908.2742763.

[99] Yang Song and Li-wei He. Optimal rare query suggestion with implicit user feedback. In *Proceedings of the 19th International Conference on World Wide*

Web, WWW '10, pages 901–910, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772782. URL http://doi.acm.org/10.1145/1772690.1772782.

[100] Yang Song, Dengyong Zhou, and Li-wei He. Query suggestion by constructing term-transition graphs. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 353–362, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-0747-5. doi: 10.1145/2124295.2124339. URL http://doi.acm.org/10.1145/2124295.2124339.

[101] Brian H. Spitzberg and Gregory Hoobler. Cyberstalking and the technologies of interpersonal terrorism. *New Media & Society*, 4(1):71–92, 2002. doi: 10.1177/14614440222226271. URL http://dx.doi.org/10.1177/14614440222226271.

[102] S. Su, S. Xu, X. Cheng, Z. Li, and F. Yang. Differentially private frequent itemset mining via transaction splitting. *IEEE Transactions on Knowledge and Data Engineering*, 27(7), July 2015. ISSN 1041-4347.

[103] Wenhai Sun, Bing Wang, Ning Cao, Ming Li, Wenjing Lou, Y. Thomas Hou, and Hui Li. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ASIA CCS '13, pages 71–82, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1767-2. doi: 10.1145/2484313.2484322. URL http://doi.acm.org/10.1145/2484313.2484322.

[104] L. Sweeney. Protecting job seekers from identity theft. *IEEE Internet Computing*, 10(2):74–78, March 2006. ISSN 1089-7801. doi: 10.1109/MIC.2006.40.

[105] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[106] Christine Task and Chris Clifton. A guide to differential privacy theory in social network analysis. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '12, pages 411–417, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4799-2. doi: 10.1109/ASONAM.2012.73. URL `http://dx.doi.org/10.1109/ASONAM.2012.73`.

[107] Hien To, Gabriel Ghinita, and Cyrus Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *Proc. VLDB Endow.*, 7(10): 919–930, June 2014. ISSN 2150-8097. doi: 10.14778/2732951.2732966. URL `http://dx.doi.org/10.14778/2732951.2732966`.

[108] Hien To, Liyue Fan, and Cyrus Shahabi. Differentially private h-tree. In *Proceedings of the 2Nd Workshop on Privacy in Geographic Information Collection and Analysis*, GeoPrivacy'15, pages 3:1–3:8, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3969-8. doi: 10.1145/2830834.2830837. URL `http://doi.acm.org/10.1145/2830834.2830837`.

[109] Hien To, Kien Nguyen, and Cyrus Shahabi. Differentially private publication of location entropy. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '16, pages 35:1–35:10, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4589-7. doi: 10.1145/2996913.2996985. URL http://doi.acm.org/10.1145/2996913.2996985.

[110] Hien To, Gabriel Ghinita, Liyue Fan, and Cyrus Shahabi. Differentially private location protection for worker datasets in spatial crowdsourcing. *IEEE Transactions on Mobile Computing*, 16(4):934–949, April 2017. ISSN 1536-1233. doi: 10.1109/TMC.2016.2586058.

[111] Giang Tran, Ata Turk, B. Barla Cambazoglu, and Wolfgang Nejdl. A random walk model for optimization of search impact in web frontier ranking. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 153–162, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767737. URL http://doi.acm.org/10.1145/2766462.2767737.

[112] Ellen M Voorhees and William R Hersh. Overview of the trec 2012 medical records track. In *TREC'12*, 2012.

[113] Hongning Wang, Yang Song, Ming-Wei Chang, Xiaodong He, Ryen W. White, and Wei Chu. Learning to extract cross-session search tasks. In *Proceed-*

ings of the 22Nd International Conference on World Wide Web, WWW '13, pages 1353–1364, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488507. URL http://doi.acm.org/10.1145/2488388.2488507.

[114] Christina Warren. 10 people who lost jobs over social media mistakes. *Mashable. Mashable*, 16, 2011.

[115] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148204. URL http://doi.acm.org/10.1145/1148170.1148204.

[116] Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Query clustering using user logs. *ACM Trans. Inf. Syst.*, 20(1):59–81, January 2002. ISSN 1046-8188. doi: 10.1145/503104.503108. URL http://doi.acm.org/10.1145/503104.503108.

[117] Ryen W. White, Ian Ruthven, Joemon M. Jose, and C. J. Van Rijsbergen. Evaluating implicit feedback models using searcher simulations. *ACM Trans. Inf. Syst.*, 23(3), July 2005. ISSN 1046-8188.

[118] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1200–1214, Aug 2011. ISSN 1041-4347. doi: 10.1109/TKDE.2010.247.

[119] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *The VLDB Journal*, 22(6):797–822, 2013. ISSN 0949-877X. doi: 10.1007/s00778-013-0309-y. URL `https://doi.org/10.1007/s00778-013-0309-y`.

[120] L. Xu, C. Jiang, J. Wang, J. Yuan, and Y. Ren. Information security in big data: Privacy and data mining. *IEEE Access*, 2:1149–1176, 2014. ISSN 2169-3536. doi: 10.1109/ACCESS.2014.2362522.

[121] S. Xu, S. Su, X. Cheng, Z. Li, and L. Xiong. Differentially private frequent sequence mining via sampling-based candidate pruning. In *2015 IEEE 31st International Conference on Data Engineering, ICDE'15*, pages 1035–1046, April 2015. doi: 10.1109/ICDE.2015.7113354.

[122] Grace Hui Yang and Ian Soboroff. Privacy preserving ir 2015: A sigir 2015 workshop. In *SIGIR Forum*, volume 49, pages 98–101, 2015.

[123] Grace Hui Yang and Sicong Zhang. Tutorial: Differential privacy for information retrieval. In *the 3rd ACM International Conference on the Theory of Information Retrieval, ICTIR'17. Amsterdam, Netherlands.*, 2017.

[124] Hui Yang, Dongyi Guan, and Sicong Zhang. The query change model: Modeling session search as a markov decision process. *ACM Trans. Inf. Syst.*, 33(4): 20:1–20:33, May 2015. ISSN 1046-8188. doi: 10.1145/2747874. URL `http://doi.acm.org/10.1145/2747874`.

[125] Hui Yang, Ian Soboroff, Li Xiong, Charles L.A. Clarke, and Simson L. Garfinkel. Privacy-preserving ir 2016: Differential privacy, search, and social media. In *SIGIR '16*, 2016. ISBN 978-1-4503-4069-4.

[126] Yin Yang, Zhenjie Zhang, Gerome Miklau, Marianne Winslett, and Xiaokui Xiao. Differential privacy in data publication and analysis. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 601–606, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1247-9. doi: 10.1145/2213836.2213910. URL `http://doi.acm.org/10.1145/2213836.2213910`.

[127] Reza Zafarani and Huan Liu. Connecting users across social media sites: A behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 41–49, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487648. URL `http://doi.acm.org/10.1145/2487575.2487648`.

[128] Sicong Zhang and Grace Hui Yang. Deriving differentially private session logs for query suggestion. In *the 3rd ACM International Conference on the Theory of Information Retrieval, ICTIR'17. Amsterdam, Netherlands.*, 2017.

[129] Sicong Zhang and Hui Yang. Applying the query change retrieval model on session search – georgetown at trec 2013 session track. In *TREC'13*, 2013.

[130] Sicong Zhang, Dongyi Guan, and Hui Yang. Query change as relevance feedback in session search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 821–824, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484171. URL http://doi.acm.org/10.1145/2484028.2484171.

[131] Sicong Zhang, Jiyun Luo, and Hui Yang. A pomdp model for content-free document re-ranking. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 1139–1142, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2257-7. doi: 10.1145/2600428.2609529. URL http://doi.acm.org/10.1145/2600428.2609529.

[132] Sicong Zhang, Hui Yang, and Lisa Singh. Increased information leakage from text. In *PIR'14 Workshop of SIGIR*, pages 41–42, 2014.

[133] Sicong Zhang, Hui Yang, and Lisa Singh. Applying epsilon-differential private query log releasing scheme to document retrieval. In *PIR'15 Workshop of SIGIR 2015*, 2015.

[134] Sicong Zhang, Grace Hui Yang, Lisa Singh, and Li Xiong. Safelog: Supporting web search and mining by differentially-private query logs. In *2016 AAAI Fall Symposium Series*, 2016.

[135] Sicong Zhang, Hui Yang, and Lisa Singh. Anonymizing query logs by differential privacy. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 753–756, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. doi: 10.1145/2911451.2914732. URL http://doi.acm.org/10.1145/2911451.2914732.

[136] J. Zhu, S. Zhang, L. Singh, G. H. Yang, and M. Sherr. Generating risk reduction recommendations to decrease vulnerability of public online profiles. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM'16*, pages 411–416, Aug 2016. doi: 10.1109/ASONAM.2016.7752267.