Information Processing and Management 51 (2015) 74–88

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

K_θ-affinity privacy: Releasing infrequent query refinements safely

Claudio Carpineto *, Giovanni Romano

Fondazione Ugo Bordoni, Rome, Italy

ARTICLE INFO

Article history:
Received 3 April 2014
Received in revised form 24 October 2014
Accepted 31 October 2014

Keywords:
k_Θ-affinity privacy
Search log anonymization
Semantic k-anonymity
n-grams
Graph k-cores
Query expansion

ABSTRACT

Search log k-anonymization is based on the elimination of infrequent queries under exact matching conditions, usually at the cost of high data loss. We present a semantic approach to k-anonymity, termed k_Θ-affinity, in which a query can be protected by affine rather than identical queries. Based on the observation that many infrequent queries can be seen as refinements of a more general frequent query, we develop a three-step privacy model. We first represent query concepts as probabilistically weighted n-grams and extract them from the search log data. We then expand the original log queries with such concepts, defining the affinity between two queries as the similarity of their expanded representations. Finally, after building the graph of Θ-affine queries (for a given threshold Θ), we find the *generalized k-cores* of this graph, which coincide with the sets of queries satisfying k_Θ-affinity privacy. Experimenting with the AOL dataset, we compare k-anonymity under affinity to k-anonymity under equality and under WordNet generalization. We show that k_Θ-affinity achieves similar levels of privacy while at the same time reducing the data losses to a great extent. We also discuss its sensitivity to attacks.

1. Introduction

Search log data contain information about the interactions between users and search engines, usually including user IP, request time, query text, search results, and clickthrough data. The analysis of users' activity (as recorded in search logs) allows improvement and personalization of search services along several dimensions. Query logs are valuable for researchers and data analysers to develop and test better ranking and query refinement algorithms (Jiang, Pei, & Li, 2013), as well as to understand query intents (Jansen, Booth, & Spink, 2008) and query reformulation strategies (Carpineto, Romano, & Bernardini, 2012) and to detect shifting trends in longitudinal search behavior (Beitzel, Jensen, Chowdhury, Frieder, & Grossman, 2007). In addition, they have proved to be useful for many other applications, including targeted advertisement (Burghardt, Böhm, Guttman, & Clifton, 2010) and detection of epidemics (Ginsberg et al., 2009).

However, as query logs are subject to disclosure of personal information, storing and publishing these data pose several threats to their owners, discussed by Cooper (2008). As a matter of fact, since the AOL incident in 2006, in which a user was identified from a search log with randomized user identifiers (Barbaro & Zeller, 2006), the collectors of search log data have been wary of publishing them for new data users. One of the rare exceptions (to our knowledge) are the datasets made available at the Workshops on Web Search Click Data in recent WSDM conferences (see e.g., Serdyukov, Dupret, & Craswell

* Corresponding author.

http://dx.doi.org/10.1016/j.ipm.2014.10.009
0306-4573/

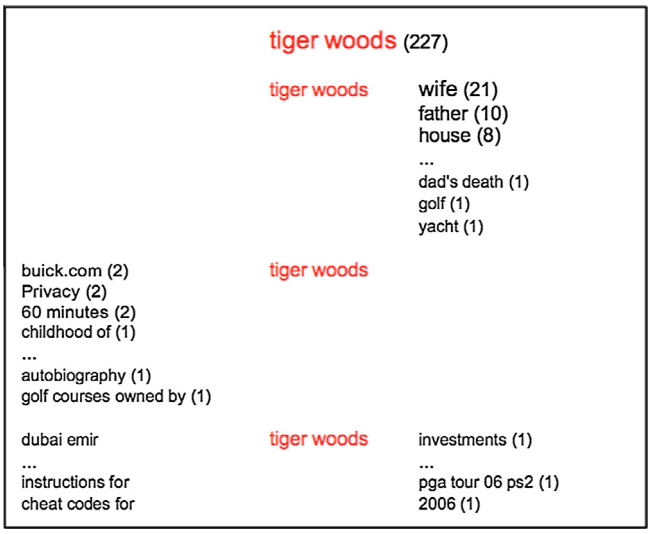

Fig. 1. A small sample of the 365 distinct AOL queries containing the string 'tiger woods', ordered by pattern and by frequency.

(2014)), which are meant for the evaluation of search log mining algorithms and are fully anonymized; i.e., all users, queries, query terms, urls, url domains, and clicks are numbers.[1]

The removal of specific entities such as names, age, and address through ad hoc techniques does not prevent user identification. Since 2006, several anonymization methods have been proposed that modify the content of the original query log in more principled ways; e.g., (Adar, 2007; Korolova, Kenthapadi, Mishra, & Ntoulas, 2009; Hong, He, Vaidya, Adam, & Atluri, 2009; Götz, Machanavajjhala, Wang, Xiao, & Gehrke, 2012). Such techniques lie along the spectrum of trade-offs that exist between privacy guarantees and data utility. Typically, increasing the limitations to information disclosure decreases the amount of useful data retained. As both the utilities gained from search logs and the discovery of private data from them rely, to a significant extent, on information retrieval techniques, the information retrieval community is in a good position to address this research problem.

One fundamental type of disclosure is represented by single queries that are unique or quasi-unique identifiers of some individual. This problem can be tackled using the well known model of k-anonymity for search query logs (Adar, 2007).

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

In this way, there is at most $1/k$ probability to link a query to a specific individual. However, this method leads to extreme data loss, with the deletion of a huge number of potentially useful and presumably harmless queries. For instance, about 90% of distinct queries in the AOL search log dataset (containing about 10 millions distinct queries submitted by about 650,000 users from March to May 2006) were issued by a single user.

This problem is not specific to k-anonymity because most anonymization methods rely on the removal of infrequent queries, explicitly or implicitly. Applying differential privacy to the AOL dataset, for instance, the queries whose frequency is below 110 are virtually guaranteed not to be released (Korolova et al., 2009), thus yielding a sanitized log containing a very tiny fraction (i.e., 0.14%) of the distinct queries present in the original query log.

The main drawback of k-anonymity can be tackled by relaxing the requirement that other users entered exactly the same query. In this way, a (released) infrequent query can be masked by a frequent, semantically similar item. To achieve this goal, we can exploit the fact that many infrequent queries can be seen as refinements of frequent queries, as illustrated in the following example.

We extracted and analyzed the AOL search queries about Tiger Woods. There are as many as 365 distinct queries containing the string 'tiger woods'. Most of these follow the pattern Q + R, where Q is the string 'tiger woods' and R is a sequence of words, but even the patterns R + Q and R + Q+R are well represented, as shown in Fig. 1. The query 'tiger woods' was entered by 227 distinct users, with the overwhelming majority of queries (i.e., 327) submitted by a single user. Such queries would be deleted if we simply require that there must be at least two distinct users per query, while they contain useful information to identify the natural subtopics of the query 'tiger woods'.

On the other hand, this example suggests that if we were able to recognize the affinity of a query to a frequent canonical concept of which it can be seen as a refinement, we could increase the amount of highly infrequent queries released by

[1] These data sets should be reasonably safe, although it is conceivable that certain types of privacy-relevant entities such as person names and company names might be disclosed by comparing the released distributions to those of an external data base with unmasked queries (Kumar, Novak, Pang, & Tomkins, 2007).

k-anonymization techniques by a great deal and in a presumably safe manner. Hu, Qian, Li, pei, and Zheng (2012) have estimated that about 40% of search log queries follow a Q + R query refinement pattern.

Based on these observations, we propose the following semantic definition of k-anonymity, termed k-anonymity under affinity.

A query log L satisfies k-anonymity under a Θ-affinity threshold, noted as k_Θ-affinity, if for every query in L there exist at least $k-1$ Θ-affine queries in L issued by distinct users.

There are two main issues involved in this definition, namely the computation of the affinity between two queries and the computation of the set of k_Θ-affine queries. As affinity relies on the refinement patterns noted above, we expand each query with the concepts contained in it, modeled as probabilistically weighted n-grams that are automatically extracted from the search log. Then we show that the computation of the set of k_Θ-affine (expanded) queries can be traced back to a well known notion of graph theory, namely *k-cores* (Seidman, 1983). The solution consists of two steps: (a) building the graph of Θ-affine queries and (b) computing a generalized version of the *k-cores* of this graph, in which vertices (e.g., queries) are enriched with class (e.g., users) identifiers. We ran a range of experiments using the AOL dataset under k_Θ-affinity, aimed to evaluate the trade off between privacy and utility as well as the ability to release non-sensitive queries, including a detailed comparison with the other main semantic approach to k-anonymity, i.e., based on WordNet generalization (He & Naughton, 2009).

The main contributions of the article are the following.

- We introduce a novel notion of semantic k-anonymity that leverages the query refinement patterns observable in search log data.
- We build a practical framework that integrates query concept mining, query expansion, and graph-theoretical analysis.
- As a byproduct of our research, we identify a novel notion of generalized k-cores and provide an efficient algorithm for their computation.
- We provide a focused evaluation of the ability to retain infrequent queries not containing sensitive information, including the use of an ad hoc test set.
- We compare our approach to k-anonymity under WordNet generalization, implemented at the query level.

The remaining of the article has the following organization. After reviewing related work in Section 2, in Section 3 we describe the main components of our method, i.e., extraction of n-grams, query expansion, construction of query graph, and computation of generalized k-cores. We then present in Section 4 the experiments with the AOL search log data set, showing that our method is able to release a much larger amount of queries without sacrificing privacy, compared to k-anonymization under equality and under generalization. In Section 5 we discuss how our approach can be used to improve the user's privacy and its sensitivity to attacks, and finally conclude the article in Section 6.

2. Related work

This article is an extended version of a paper that appeared at the 35th European Conference on Information Retrieval (Carpineto & Romano, 2013). The present version is significantly extended, including improved procedures for query concept mining (with an extensive experimentation on the full AOL dataset) and approximate computation of k_Θ-affine queries, a discussion of privacy guarantees and sensitivity to attacks (compared to strict k-anonymity), and also a detailed comparison with the other main semantic k-anonymization method for search log data (i.e., based on WordNet generalization), implemented at the single-querylevel. We now review related work on k-anonymity and search log anonymization.

The concept of k-anonymity has been extensively studied in the database field to prevent re-identification by multiple databases linking, following the early work of Sweeney (Sweeney, 2002). It is assumed that a subset of attributes are quasi-identifiers and a record is released only if there are at least other $k-1$ records that share the same values for those attributes, which is usually achieved through generalization and suppression of attribute values. A number of systems have been developed, mainly for data which have a fixed schema with a small number of dimensions, but also for more sparse, higher-dimensional data such as market basket data (Bayardo & Agrawal, 2008; Terrovitis, Mamoulis, & Kalnis, 2008; Ghinita, Tao, & Kalnis, 2008; LeFevre, DeWitt, & Ramakrishnan, 2006). Afterwords (Adar, 2007), k-anonymity has been applied to search query logs, which are fundamentally different from set-valued or relational data. As there is no explicit distinction between quasi-identifiers and other types of information, in Adar (2007) a query serves as the quasi-identifier. However, enforcing strict k-anonymity at the query level makes it hard to retain enough utility, due to high data sparseness.

One attempt at overcoming this difficulty is to create identical queries through generalization, e.g., by replacing two different queries with their common WordNet parents. This approach is described in He and Naughton (2009), building on an early pioneering approach developed for set-valued data (Terrovitis et al., 2008). The input to the algorithm consists of aggregated (or profile) queries, one for each user. All the profile queries are first generalized to the WordNet root concept, and then they are recursively partitioned top-down using more specific hierarchy concepts to form the subpartitions. A greedy strategy splits one cluster of the current partition at a time, choosing the cluster that maximizes the information gain due to concept specialization. The algorithm halts when no more partitions containing only clusters larger than k can be generated, and the queries describing the single clusters in the most specific partition are released.

One main disadvantage of this approach is that each released generalized query typically refers to multiple heterogeneous subjects (i.e., all the interests of the user) and contains many generic concepts such as 'event', 'thing', or just 'entity' (due to the limited coverage of WordNet). In this article, we apply *k*-anonymity under WordNet generalization at the single-query level and contrast it to our approach, as will be discussed in the experimental section.

Another approach that enforces *k*-anonymity at the user level is based on clustering. The idea is to form clusters of *k* users that are similar in terms of their data, expressed as queries (Hong et al., 2009; Navarro-Arribas, Torra, Erola, & Castella-Roca, 2012) or terms (Liu & Wang, 2013), using various similarity measures and clustering algorithms. These methods significantly reduce the risk of information disclosure when multiple relatively frequent queries are taken together.

However, clustering rearranges the query log destroying the query ordering, while the cluster representatives are fictitious users created by deleting original data and adding new artificial data. It is unclear how this affects the utility of the sanitized log. Furthermore, clustering algorithms only provide heuristic solution quality.

A further search log anonymization technique that is indirectly related to *k*-anonymity is differential privacy (Korolova et al., 2009), which has strong privacy guarantees. It ensures that the amount of knowledge that an attacker can learn about a user is roughly insensitive, according to some privacy parameters set by the data releaser, to omitting or changing the user's search history.[2] Like *k*-anonymity, differential privacy does not modify the content of the single queries, but it changes their frequency and remove the association between queries and users. The method to filter the queries and set their frequencies consists of three steps: (1) selecting a limited number of queries per user, (2) altering their frequencies by injecting noise, and (3) releasing only the queries with a frequency higher than a given threshold.

Differential privacy, for a typical choice of its parameters, results in the suppression of all rare as well as relatively frequent queries, up to frequencies of the order of hundreds (Korolova et al., 2009). Furthermore, the destruction of the association between users and queries prevents some of the most interesting applications of published search log data. In fact, the utility of differential privacy has been deeply questioned (Götz et al., 2012) due to the huge involved data loss. A recent proposal extends differential privacy to preserve associations between users and queries, but it requires the specification of a particular objective function to be optimized (Hong, Vaidya, Lu, & Wu, 2012).

Besides deleting infrequent queries (explicitly or implicitly), there are other popular approaches to protect the privacy of published logs, reviewed in Cooper (2008). One strategy is to alter the user identifiers without changing the query content; e.g., by removing all identifiers that make it possible to associate multiple queries with the same user, or by shortening the length of time that any identifier is associated with an individual. An alternative strategy consists of modifying the queries and keeping the user identifiers; e.g., by replacing queries or query tokens with a set of hash values, or by scrubbing identifying information from the remainder of the query content. Each technique has strengths and weaknesses but a direct comparison in terms of privacy protection and utility preservation is difficult unless we consider specific privacy models or specific usages of the query logs.

Deleting infrequent queries seems to achieve a good balance in preserving both the user identifiers and the original content of released queries, although it has its own limitations. The huge data loss and its reliance on high volume queries may undermine language-based applications of the query logs and query refinement. At the same time, it is prone to privacy breaches because an individual user can be re-identified through combination of multiple, relatively-frequent queries and other publicly available data.

Our framework addresses some of these limitations. We show that using k_Θ-affinity we are able to retain many useful infrequent query refinements, thus greatly increasing the utility of anonymized logs at the cost of limited privacy degradation. We also argue that k_Θ-affinity may be used to increase the level of privacy protection while keeping the utility constant, by replacing relatively-frequent unrefined queries that may be harmful to privacy with infrequent harmless query refinements.

3. Method description

Our full anonymization method is illustrated in Fig. 2. We describe its main steps, in turn, in the following sections.

3.1. Query concept mining

Our approach is based on word *n*-grams, because we aim to identify query portions corresponding to canonical concepts. However, *n*-grams do not always correspond to meaningful concepts or to concepts relevant to the meaning of the query. An example of the former is 'is a', an example of the latter is the sequence 'Saturday night' in the two queries 'Saturday night fever' and 'Saturday night live'.

The identification of key *n*-grams in search queries is usually carried out by combining statistical and grammatical evidence (Collins-Thompson & Callan, 2005; Kumaran & Allan, 2007; Hu et al., 2012), although supervised machine learning techniques are also used (Bendersky & Croft, 2008). We follow the former approach, combining multiple complementary criteria (e.g., user frequency, term dependency, part-of-speech tagging) in a simple and effective scheme tailored to the specific features of the anonymization task.

[2] The differential privacy model does not explicitly rely on the *k* anonymity parameter, although it can be modified in this direction (Feild, Allan, & Glatt, 2011).

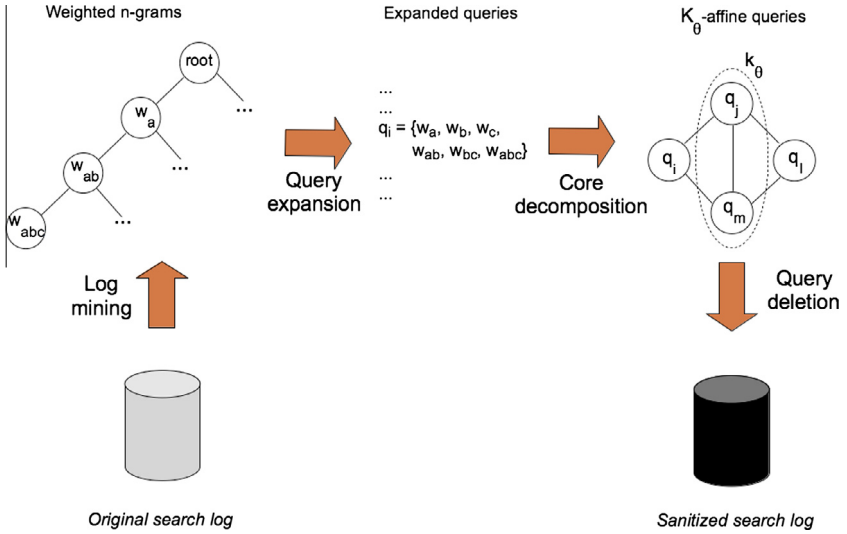

Fig. 2. General scheme of k_Θ-affinity privacy.

Table 1
Distribution of query length in the AOL search query log.

Query length	Number and % of unique queries		Number and % of queries (with repetitions)	
1	2,705,784	26.64%	11,909,026	41.21%
2	2,053,326	20.22%	7,089,861	24.53%
3	2,088,671	20.56%	4,536,990	15.69%
4	1,479,143	14.56%	2,568,638	8.88%
5+	1,827,818	17.99%	2,793,846	9.66%
Total	10,154,742	100.00%	28,898,361	100.00%

Table 2
N-grams statistics for the AOL search query log.

N-grams	Number of unique *n*-grams	Number of *n*-grams (with repetitions)
Unrestricted n-grams		
Unigrams	3,849,555	67,567,111
Bigrams	7,409,939	38,668,750
Trigrams	8,530,452	21,679,415
Frequent n-grams (user freq $\geqslant$ *0.01%)*		
Unigrams	30,716	49,346,473
Bigrams	24,298	11,761,821
Trigrams	5089	1,838,235

We first extract all word-level unigrams, bigrams, and trigrams contained in the search log data. Although our approach can be easily extended to larger *n*-grams, we at most consider trigrams because refinements of four-grams or five-grams query concepts are probably rare. The AOL query length statistics reported in Table 1 support this choice: only about 18% of distinct queries contain five or more terms.[3]

To efficiently compute, store, and retrieve the *n*-grams information, we use three tries, one for each type of *n*-grams. Bigrams and trigrams are seen as a sequence of characters (rather than words), and each *n*-gram entry stores the number of occurrences and the number of distinct users associated with that *n*-gram. In the top part of Table 2 we show the statistics about the number of *n*-grams (unique and with repetitions) contained in the whole AOL search log.

We then filter out infrequent *n*-grams, consistent with our focus on frequent query concepts. For the AOL search log, we required that each *n*-gram be supported by at least 65 users; i.e., one-ten-thousandth of the user population. Choosing this

[3] The total number of queries with repetitions (i.e., 28,898,361) has been computed assuming that two identical consecutive queries count for two, provided that their query time is different.

Table 3
Top ten bigrams and trigrams extracted from the AOL search log data set ordered by their mutual information score.

Top bigrams		Top trigrams	
estados unidos	19.28	priory of sion	18.64
myasthenia gravis	19.18	wisin y yandel	18.36
bryn mawr	19.14	buca di beppo	18.34
iwo jima	19.07	tristan and isolde	17.93
gwyneth paltrow	19.07	mitral valve prolapse	17.77
lynard skynard	19.01	preferencia de idioma	17.77
alanis morissette	18.83	osama bin laden	17.70
pina colada	18.74	bausch & lomb	17.70
smashing pumpkins	18.74	plow and hearth	17.55
shiba inu	18.74	ku klux klan	17.44

value seems a good compromise between removing noisy query concepts and matching a large majority of queries, as shown in the next section.[4]

We also remove those n-grams that are exclusively formed by words with little informative content; e.g., prepositions, articles, conjunctions, auxiliary verbs. In the bottom part of Table 2 we show the statistics about the number of n-grams (with and without repetitions) contained in the AOL search log, after pruning with the user frequency threshold.

The remaining n-grams are finally weighted. Unigrams are weighted using their frequency, i.e.,

$$W_x = \log_2(N+1) \tag{1}$$

where N is the number of queries in which term x occurs. In this way, we assign low weights to words associated with very few users. In the AOL data set for instance, the top three unigrams (along with their scores) are: free (18.16), google (18.09), http (17.84). For bigrams and trigrams we use mutual information, a well known measure for word association which compares the probability of a sequence of words to occur together to their probabilities of occurring independently. The bigram mutual information is defined as (Church & Hanks, 1990):

$$W_{x,y} = \log_2\left[\frac{P_{x,y}}{P_x \cdot P_y} + 1\right] \tag{2}$$

where $P_{x,y}$ is the joint probability that term y follows term x, and P_x and P_y are the probability of occurrence of x and y, respectively.[5] Such probabilities are estimated by relative frequency counts. The mutual information of a trigram is defined as (Su, Hsu, & Sailard, 1991):

$$W_{x,y,z} = \log_2\left[\frac{P_{x,y,z}}{P_x \cdot P_y \cdot P_z + P_x \cdot P_{y,z} + P_{x,y} \cdot P_z} + 1\right] \tag{3}$$

To give an impression of the query concepts generated using the procedure illustrated above, in Table 3 we show the top ten bigrams and trigrams extracted from the AOL data set along with their mutual information weights. All of the concepts are meaningful. They generally refer to well or relatively well known entities of various nature, with a predominance of musicians.

3.2. Query concept expansion and θ-affinity

Each query is represented as a weighted concept vector including all the unigrams, bigrams, and trigrams contained in the query. For instance, the query '*a b c*' will be represented by the following n-grams: $w_a,\ w_b,\ w_c,\ w_{ab},\ w_{bc},\ w_{abc}$. Although we do not add new terms to a query, the grouping of terms in weighted concepts and their explicit use in the query representation can be seen as a form of query expansion (Carpineto & Romano, 2012).

Using the set of 60,103 concepts extracted in the preceding step, we expanded all the queries in the AOL data set. We checked that 2,993,219 distinct queries were not affected at all, i.e., they contained no key concepts (n-grams). We then removed the expanded queries which contained, in addition to some key concept, one or more words associated with a unique user. This latter operation resulted in a large data loss, but it was useful to mitigate the risks of potential privacy breaches caused by query affinity, as will be discussed in Section 5. We were left with 5,037,881 distinct expanded queries (or 14,801,973 expanded queries with repetitions). As their n-grams weights were comparable (see Table 3), we did not perform any normalization.

The expanded representations are used to assess the query affinity. We say that two queries $p,\ q$ are Θ-affine if the cosine similarity of their expanded representations $p_E,\ q_E$, is greater than or equal to a threshold Θ:[6]

[4] More sophisticated strategies for setting the threshold are conceivable but they would require significant computational effort, such as automatically optimizing some global system performance measure or relating the numerical value to some user-specified preferences.

[5] This is an asymmetric version of the mutual information, where word order matters; e.g., compare 'book bar' to 'bar book'.

[6] Clearly, the Θ-affinity relation is not transitive.

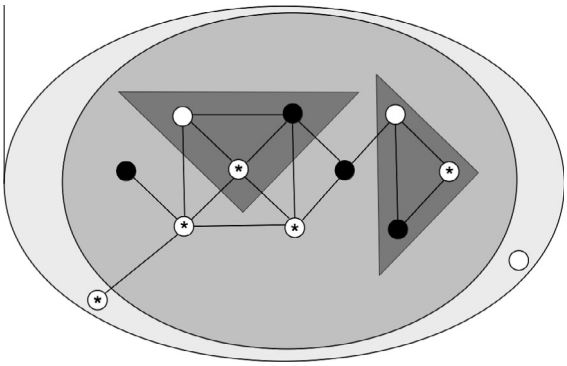

Fig. 3. Generalized cores of order 0, 1, 2 (for twelve vertices split in three classes).

$$Sim_{cosine}(p_E, q_E) = \frac{\overrightarrow{p_E} \cdot \overrightarrow{q_E}}{||\overrightarrow{p_E}||||\overrightarrow{q_E}||} \geqslant \Theta \tag{4}$$

As an illustration, consider the queries 'cell phone case' and 'nokia 2651cell phone case'. The list of concepts (i.e, the n-grams entered by at least 65 users) together with their weights is the following: cell (14.32), phone (15.33), case (13.24), nokia (11.35), cell phone (10.8), phone case (6.95). The similarity between the expanded representations of the two queries is equal to 0.92. We will return to this example in Section 3.4.

We talk about Θ-affinity instead of Θ-similarity to emphasize the fact that there is a structural resemblance indicating a common concept, while the queries may be superficially quite different. Note, however, that although our approach represents a departure from the traditional syntactic approach to search log k-anonymization, the functions used to weight n-grams do not necessarily reflect their semantic importance. The unigrams are weighted using their frequency, which may be useful to protect the user privacy but may have less value to utility; e.g., 'google'. Bigrams and trigrams are weighted with mutual information, which tells us how cohesive or related the terms are regardless of their relevance. In practice, two concepts may play a semantically similar role while receiving markedly different mutual information scores, depending on whether their terms tend to co-occur together (e.g., $W_{gwyneth.paltrow} = 19.07$) or not (e.g., $W_{james.bond} = 10.50$). An alternative, more semantic, approach to assess the affinity between queries is based on concepts extracted from user queries using a knowledge base (Wikipedia, YAGO, Freebase, etc.) or on clickthrough data, although for infrequent queries this types of information may be scarcely available. We did not investigate such approaches.

The next step is to find a maximum subset L' of the original query log L, such that each query in L' is Θ-affine to at least other k queries in L'. The computation of L' is not straightforward, because the deletion of a query that does not satisfy the k_Θ-affinity property can invalidate some queries that have been already evaluated. This problem can be solved by means of generalized graph k-cores. This is discussed in the next section.

3.3. Generalized k-cores of the graph of Θ-affine queries

A *k-core* (or core of degree k) of a graph is defined as the maximum subset of vertices such that every vertex has at least k neighbors in it, where k is some integer (Seidman, 1983). For our purpose, it is convenient to build a graph whose vertices are the queries and where there is an edge between two vertices if the corresponding queries are Θ-affine. If all Θ-affine queries of each vertex are made by distinct users, the k-core of this graph coincides with the set of queries satisfying $(k+1)_\Theta$-anonymity. For the general case when in a neighborhood there are multiple queries from the same user, caution must be taken to ensure that the queries of the same user only count for 1 in the computation of k.

We refer to this type of k-cores as generalized k-cores, because we assume that the vertices are labeled with class identifiers (which, in our case, are user identifiers) and that the degree is given by the number of distinct classes (rather than vertices) in the neighborhood.[7] Formally, let $G = (V, E)$ be a graph, where V is the set of vertices ($|V| = n$) and E is the set of edges ($|E| = m$). Let $f : V \to \mathbb{Z}$ be a function assigning integer values (e.g., user identifiers) to vertices, and denote by $N_G(v)$, $v \in V$, the neighborhood of v, consisting of all vertices adjacent to v (i.e., connected to v by an edge). A subgraph $C = (U, E|U)$ induced by the set $U \subseteq V$ (i.e., such that it contains any edges whose endpoints are both in U) is a generalized *k-core* iff:

(a) $\forall\ u \in U,\ \exists\ X \subseteq N_G(u)$, such that:

$\forall x \in X,\ f(x) \neq f(u)$,

[7] Note that this is different from p-cores (Batagelj & Zaversnik, 2002), where the goal is to find the set of vertices with a property value larger than a given threshold.

$\forall x,\ y \in X,\ f(x) \neq f(y)$,
and $|X| \geqslant k$;

(b) C is the maximum subgraph with property (a).

This definition holds for $k \geqslant 2$. The generalized core is of order 1 when X contains at least one vertex with a class different than $f(u)$, while disconnected vertices (or vertices connected only to vertices of the same class) form a generalized 0-core. Note that standard k-cores are a special kind of generalized k-cores; i.e., when all classes are distinct. Note also that this definition can be easily generalized to deal with multiple classes assigned to the same vertex (e.g., when a same query is issued by a set of users). The difference is that the classes of each vertex in X must be different from any of the classes associated with the other vertices in X, including the classes of u.

An illustration is given in Fig. 3 for twelve vertices labeled with three classes; i.e., white, black, and asterisk. The three generalized k-cores are nested and are depicted with different levels of gray. Note that the core of order 2 is formed by two unconnected subgraphs, and that the black node between these two subgraphs belongs to the core of order 1 despite being linked to vertices of the other two classes.

From the definition of k_Θ-affinity privacy and generalized k-core, the following statement holds.

The generalized k-core of the graph of Θ-affine queries coincides with the maximum subset of queries satisfying $(k+1)_\Theta$-affinity

To construct the graph of Θ-affine queries, we score the full set of queries against each expanded query, ordering the results by affinity. This operation is performed efficiently using an inverted index that associates each n-gram with the queries in which it appears, similar to document ranking. For a certain value of Θ, the graph is then formed by linking all the pairs of Θ-affine queries.

To construct the generalized k-cores of the graph of Θ-affine queries, we modify the algorithm described in Batagelj and Zasversnik (2003) to find k-cores. The algorithm in Batagelj and Zasversnik (2003) is based on the observation that if we recursively remove a vertex with degree smaller than k and all its adjacent edges from a given graph, the remaining graph is the k-core. In practice, it processes the vertices in increasing order of their degree, assigning to each vertex a core number equal to its current degree and decreasing the degree of the vertices that are linked to it. The algorithm returns for each vertex its core degree. Thanks to careful design of updating and re-ordering of vertices, its time complexity is $O(\max(m,n))$, where m and n are the number of vertices and edges, respectively.

The main difference between the basic and the generalized version of the algorithm is that neighbors are represented as query-user pairs rather than simple queries, and that the degree of a vertex is updated only when all the set of queries associated with a user in its neighbor becomes empty. Suitable data structures ensure that the generalized algorithm has the same complexity as the ungeneralized one.

In the last step of our method, the sanitized log for a certain value of k (say k_0) is generated by releasing all the queries with an associated degree of anonymity $k \geqslant k_0$. If a query could not be expanded, its anonymity degree is given by the number of distinct user who issued an identical query (as with plain k-anonymity), otherwise it is given by its generalized core degree plus 1.

3.4. Approximate computation of k_Θ-affinity privacy

The computation of k_Θ-affinity privacy can be performed by finding the graph of Θ-affine queries, given a certain value of Θ, and then processing the resulting graph for determining the generalized k-cores. However, depending on the value of Θ, the number of edges may grow very large because some queries will be Θ-affine to many other queries, thus slowing down the algorithm for finding generalized k-cores. In spite of the good theoretical properties of this algorithm, the construction of the sanitized log may thus become practically difficult for large search query logs and values of Θ of interest.

To overcome this problem, we develop an approximate procedure that does not require the construction of the full graph of Θ-affine queries and the subsequent computation of the generalized k-cores from this graph. The idea is to work with several small subgraphs, one for each query, as detailed below.

Given a query q and a specified value of Θ, we build a graph G_q of Θ-affine queries centered around q, by iteratively computing the set of neighbors (for the given value of Θ) and omitting the duplicates, until no new neighbors have been generated or a specified maximum number of vertices *Vmax* has been reached. This graph coincides with the subgraph (of the full database graph) formed by the corresponding queries.

We next compute the generalized k-cores for all vertices in G_q. The found degree of each vertex is a lower bound of the true degree of that vertex in the full database graph G, because the omitted vertices could but increase the degree of the vertices in G_q. For the case when the procedure halts before reaching the maximum allowed size (which means that G_q coincides with a disconnected component of the full graph G), the found degree is equal to the true degree.

To illustrate, in Fig. 4 we show the generalized k-cores of the graph obtained for the AOL query 'cell phone case', with $\Theta = 0.9$, *Vmax* = 2000.[8] The algorithm halted after generating 21 queries, listed in the caption of Fig. 4, which means that the shown graph is a true disconnected component of the whole graph of the AOL dataset (for $\Theta = 0.9$). All queries except

[8] The image was drawn by using the chart.ravenbrook.com server.

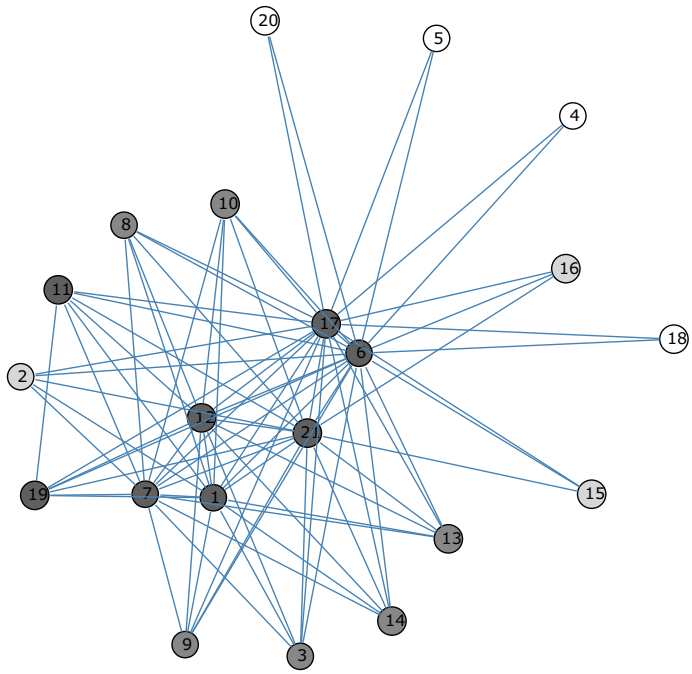

Fig. 4. Generalized *k*-cores of the graph originated from the AOL query 'cell phone case'. The complete list of queries is the following ('cell phone case' being abbreviated with cpc): (1) 3100 cpc, (2) sanyo 3100 cpc, (3) blackberry cpc, (4) playboy cpc, (5) coach cpc, (6) cpc 2, (7) cpc e815, (8) rutgers cpc, (9) flame cpc, (10) dolphin cpc, (11) cpc strap, (12) rugged cpc, (13) yorkie cpc, (14) jewel cpc, (15) nokia 2651 cpc, (16) la cg300 cpc, (17) waterproff cpc, (18) leather cpc, (19) titanium cpc, (20) cpc stars, (21) i530 cpc.

one were entered by only one user, with users overlapping across different queries. The query graph contained four associated generalized *k*-cores, with degree 2, 3, 5, 6. In Fig. 4, the vertices in a same generalized *k*-core are depicted with the same tone of gray (the darker the tone, the higher the degree). Note that in this example, using $k_{0.9}$-affinity with $k > 1$, all queries would be released, as opposed to suppressing all of them based on plain *k*-anonymity. Note also that there are many other AOL log queries containing the string 'cell phone' which were not selected due to lower affinity.

Using this approximate procedure, the key computational factor becomes the construction of the graph centered around each random query, which limits the number of random queries that can be analyzed. On the other hand, we will see in the experimental Section 4.1 that in practice a relatively small number of random queries can ensure reliable results.

4. Evaluation

4.1. Analysing the trade off between privacy and utility

In this section we study the trade-off between privacy and information loss when applying k_Θ-anonymization to the AOL dataset. Rather than considering all AOL queries, we used a set of random queries, following the approximate procedure described in Section 3.4. In order to decide how many queries to use, we experimented with increasing random samples until the results stabilized. We found that 5000 random queries ensured representative results.

In Fig. 5(a) we show how the number of released queries (in percentage) varies as a function of *k*, for three values of Θ: 1, 0.9, 0.8. The results were averaged over ten random samples to avoid results attributable to chance. Note that for $\Theta = 1$, we get exactly the number of queries released under plain *k*-anonymization. The three methods are denoted, respectively, as EQ (i.e., short for *k*-anonymization under equality), AFF (i.e., short for *k*-anonymization under affinity) ($\Theta = 0.9$), and AFF ($\Theta = 0.8$). For $\Theta = 0.9$, the computation of subgraphs always halted before reaching the maximum size (with an average size of 25 vertices), while with $\Theta = 0.8$ many subgraphs were approximated.

The figure clearly shows the trade-off between *k*-anonymity and data release associated with each privacy policy. Using the semantic method, the gain in terms of released queries is massive for all values of k. Furthermore, a comparison between the *k*-anonymity and k_Θ-affinity plots suggests that this gain grows monotonically with *k*. For instance, for $\Theta = 0.9$, the percentage improvement is about 100% for $k = 2$ and 1000% for $k = 10$. By inspecting the query subgraphs, we noticed that the queries with a much increased value of *k* were typically linked to one query entered by many distinct users, as in the Tiger Woods example.

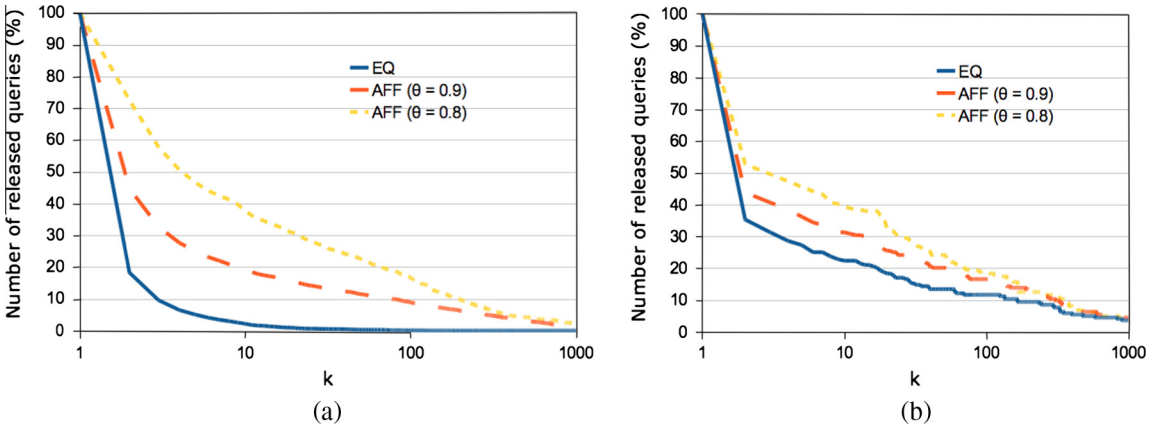

Fig. 5. Proportion of released queries as a function of k for a subset of the AOL data set (a) and for AOL user 4417749 (b), using plain k-anonymity (EQ) and k_Θ-affinity (AFF). The x axis is logarithmic.

An analogous experiment was performed on the AOL user 4417749, identified by the New York Times (Barbaro & Zeller, 2006). The 224 distinct queries entered by user 4417749 were analyzed using k-anonymity and k_Θ-affinity. In Fig. 5(b) we show the proportion of query released for user 4417749 under the different privacy policies. Similar to Fig. 5(a), there is a tangible growth of released queries as Θ becomes smaller. The main difference is that in Fig. 5(b) the plots are closer because user 4417749 entered fewer unique queries than the average AOL user. A comparison between Fig. 5(b) and the analogous plot shown in Feild et al. (2011) under alternative privacy policies suggests that our method released a larger percentage of queries, both for $\Theta = 0.8$ and $\Theta = 0.9$.

As the value of Θ decreases, the set of released queries becomes by definition larger, but there are of course more risks of privacy breach. This issue is addressed in the next two sections.

4.2. Privacy of AOL user 4417749

Search log privacy has two main sides, identification of user and disclosure of sensitive information. To evaluate these aspects, we consider again AOL user 4417749. She was identified by journalists from the New York Times by mainly combining queries related to user's relatives and to user's location. There are 30 identifying queries in her set of queries, nine with the same surname as the user and 21 with the location name.

For the nine surname queries the situation for EQ and AFF ($\Theta = 0.9$) was the same: all of the queries were removed for $k = 2$. Without these queries, it is virtually impossible to identify the user. In contrast, using AFF ($\Theta = 0.8$) the following four queries received a semantic anonymity degree greater than 2 (indicated in parenthesis): 'eugene oregon jarrett arnold' (18), 'eugene oregon jaylene arnold' (18), 'jack t. arnold' (3), 'jarrett arnold eugene oregon' (18).

We also considered the queries of AOL user 4417749 that are potentially sensitive; e.g., those about diseases, age, anxious feelings, civil status, etc. We identified 20 sensitive queries. Of these, eight were discarded using $k = 2$ under all privacy policies. In Table 4 we show the remaining twelve queries with their associated anonymity degree under each privacy policy, ordered by the degree of plain k-anonymization. As pointed out above, if a query does not contain any frequent concept, then the semantic anonymity degree is equal to the syntactic one, by default. For instance, the k value for the query 'paranoia' was

Table 4
k-anonymity degree of sensitive queries of AOL user 4417749, with $k \geqslant 2$ under at least one privacy policy.

Query	EQ	AFF ($\Theta = 0.9$)	AFF ($\Theta = 0.8$)
Helpful aids for arthritis	1	1	3
60 single men	1	9	9
Mature living	2	5	5
Hand tremors	3	3	14
Panic disorders	8	8	8
Vascular disease	9	19	39
Loneliness	19	21	21
Paranoia	21	21	21
Mini strokes	22	24	72
Osteopenia	61	61	61
Thyroid	329	471	491
Bipolar	332	373	374

set to 21 across all methods because the unigram 'paranoia' occurred in the queries of 50 distinct users; i.e., more than the 21 distinct users who entered the query 'paranoia' but still below the threshold value set to 65. The results reported in this section suggest that, for the specific user 4417749, AFF ($\Theta = 0.9$) offered a sufficient level of protection both for the disclosure of identifying and sensitive queries, while using AFF ($\Theta = 0.8$) the risk of privacy breaches was more tangible.

4.3. Discriminating between sensitive and non-sensitive queries

The ability to release as many as possible infrequent yet harmless queries was measured in a further experiment. As there is no standard method available, we developed our own procedure. We randomly extracted 5000 AOL queries and had them manually labeled as sensitive or non-sensitive by some colleagues of us, e.g., sensitive queries are those including facts about specific locations, times, people, or those about age, sexual preferences, religion, health concerns, etc. We next computed the anonymity degree k of these queries according to EQ, AFF ($\Theta = 0.9$), and AFF ($\Theta = 0.8$), and split the queries in two classes (i.e., released or non-released) for each value of k in the range from 1 to 1000, depending on whether their degree was $\geqslant k$ or $<k$.

We can now evaluate the performance of each k-anonymization method, seen as an information retrieval system that must retrieve (release) the relevant (non-sensitive) queries, under a certain value of k. We used the well known F_β measure (van Rijsbergen, 1979), defined as the weighted harmonic mean of precision (P) and recall (R):

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \tag{5}$$

The parameter β is a weighting factor for the importance of the recall (or precision). Because in the anonymization scenario it is probably more important to release *only* non-sensitive information than to release *all* non-sensitive information, we are interested in values of $\beta \leqslant 1$.

In Fig. 6 we show the performance of the three methods for $\beta = 1$ and $\beta = 0.1$. The EQ curve is incomplete because for $k > 172$ no queries were released under plain k-anonymity.

The main findings are the following. First, the AFF method clearly outperformed EQ for every value of k and for both Θ values. Second, AFF ($\Theta = 0.8$) markedly outperformed AFF ($\Theta = 0.9$) for $\beta = 1$, and achieved slightly better results for $\beta = 0.1$. We observed that AFF ($\Theta = 0.9$) becomes better than AFF ($\Theta = 0.8$) for further smaller values of β, i.e., when we attribute an even higher importance to precision than recall. Third, F_β decreases as k grows because recall is severely affected, unless β becomes very small. Overall, these experiments show that k_Θ-affinity can trade good levels of privacy for limited data losses in a much more effective manner than plain k-anonymization.

4.4. Comparison with k-anonymization by WordNet generalization

From a conceptual point of view, He and Naughton (2009) is the most relevant work because it describes a different semantic approach to k-anonymization of search log data in which the similarity between queries is driven by WordNet. However, the algorithm described in He and Naughton (2009) works at the user level, because its input consists of a set of profile queries, each formed by taking the union of the nouns searched by a single user.

4.4.1. Implementation of the algorithm at the single-query level

To enable fair comparison with the other k-anonymization models, we have adapted it to work at the level of single queries. The main difference is that we group two single queries together provided that there is at least one term per query that

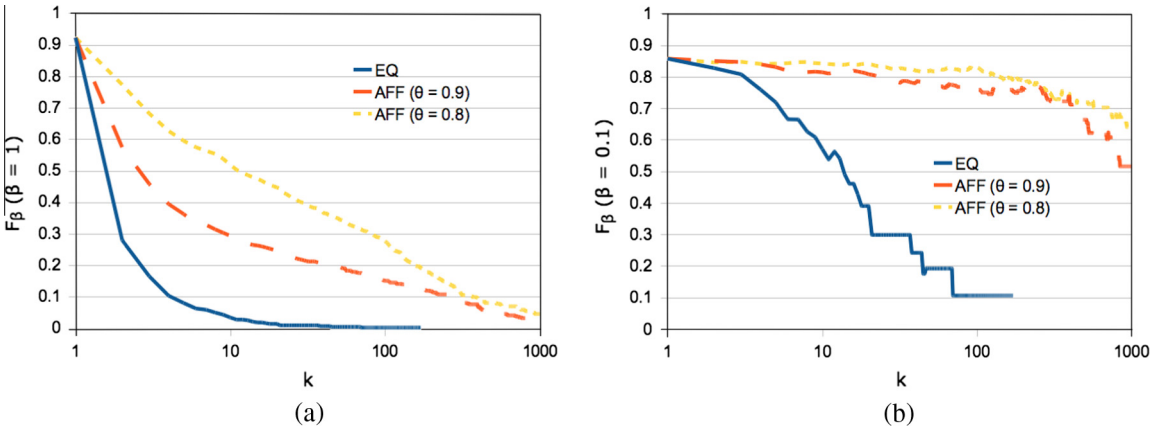

Fig. 6. F_β performance of AFF and EQ k-anonymization models on queries with sensitivity labels, for $\beta = 1$ (a) and $\beta = 0.1$ (b). The x axis is logarithmic.

Table 5
A high level partition of the AOL search log data set. The number of queries associated with each WordNet concept is shown in parenthesis.

```
entity (3,682,195)
  physical_entity (0)
    thing (158,027)
    object (1,969,348)
    causal_agent (692,302)
    matter (246,881)
    physical_process (188,357)
  abstraction (0)
    psychological_feature (1,019,832)
    attribute (590,749)
    grouping (608,554)
    relation (397,081)
    communication (601,416)
```

can be generalized to the same concept, rather than trying to form conjunctions of concepts that generalize multiple terms (as with profile queries). Each generated partition has an associated degree of anonymity corresponding to the smallest cardinality of its clusters. The more specific the partition, the lower the degree of anonymity.

We applied the implemented algorithm to the full AOL dataset. We found that 3,682,195 distinct queries (out of 10,154,742) could not be generalized because their terms did not match any WordNet concept (as of April 2013). They were thus automatically assigned to the root concept. The remaining 6,472,547 queries were recursively partitioned using the procedure illustrated above. As an illustration, in Table 5 we show the partition associated with the third recursion level. This partition ensures that the corresponding generalized queries satisfy at least 158,027-anonymity; i.e., the size of the smallest cluster in the partition.

4.4.2. Comparative evaluation of data distortion

In order to compare k-anonimity by equality and by affinity to k-anonimity by WordNet generalization (denoted by WNGEN), we cannot rely on the count of released queries because the queries released by WNGEN are distorted. We use the *Normalized Certainty Penalty* information loss measure, as introduced in Terrovitis et al. (2008) and employed in He and Naughton (2009) (our notation emphasizes that items are terms and transactions are queries).

The *Normalized Certainty Penalty* for a term t with respect to a generalization hierarchy is defined as:

$$NCP(t) = \begin{cases} 0, & |u_t| = 1 \\ |u_t|/|I|, & otherwise \end{cases} \tag{6}$$

where u_t is the node in the hierarchy tree corresponding to t, $|u_t|$ is the total number of leaves under that node, and $|I|$ is the total number of leaves in the hierarchy. $NCP(t)$ ranges from 0 (when the term is released as is) to 1 (when the term is generalized to the root). Then, the total information loss for a sanitized log L is the average of the information loss of all terms:

$$NCP(L) = \frac{\sum_{q \in L} \sum_{t \in q} NCP(t)}{T} \tag{7}$$

where T is the total number of terms, including repetitions. Although *NCP* was primarily devised to evaluate the information loss due to query generalization, it can also be applied to sanitized logs consisting of ungeneralized queries. In the latter case, *NCP(t)* takes only two values, namely 0 or 1 depending on whether t appears in a released or in a suppressed query.

In Fig. 7 we plot NCP as a function of k, using the same sets of random queries as in the experiments in Section 4.1 and under three k-anonymization privacy models, namely equality (EQ), affinity (AFF), with $\theta = 0.9$, and WordNet generalization (WNGEN). For WNGEN, the set of released queries was computed by extracting the generalized test queries from the global query tree.

The plots in Fig. 7 show that AFF and WNGEN were better than EQ for all values of k,[9] while the relative performance of AFF and WNGEN depended on k. For low values of k, AFF performed better than WNGEN, because AFF was able to release a great deal of infrequent queries that cannot be generalized by WordNet. As k grows, the number of queries released by AFF reduces and the corresponding NCP function steadily increases, while the performance of WNGEN remains nearly stable. In fact, the NCP loss of generalized queries tends to 1 only for very high values of k; e.g., when k becomes comparable to the size of the search log.

To illustrate this behavior, consider the situation in which the queries are single-term and we generalize all the terms using just two concepts that cover half of the terms each. We get $NCP = 0.5$ for $k = N/2$, and $NCP = 1$ for $k > N/2$, where N is the number of queries. Theoretically, we obtain a relatively good distortion value for a huge anonymity degree, but the NCP function probably underestimates the information loss due to query generalization, because a sanitized log consisting of a few very generic concepts seems useless.

[9] Note that a low NCP value (for a certain k) is better than a high value.

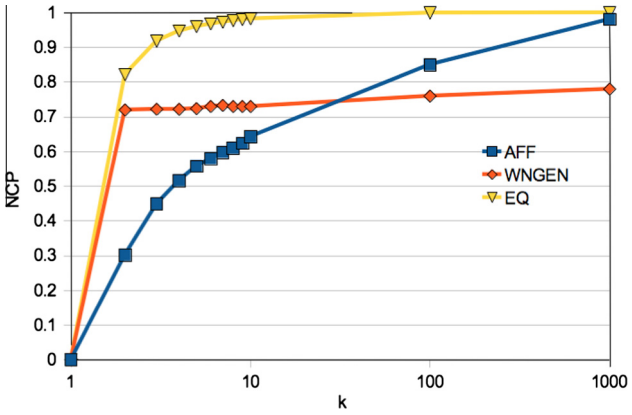

Fig. 7. Normalized Certainty Penalty as a function of k for three k-anonymization models: affinity (AFF), equality (EQ), WordNet generalization (WNGEN). The x axis is logarithmic.

At the same time, the NCP measure seems to overestimate the information loss due to query filtering. If we suppress half of the queries and release the other half unchanged, we still get $NCP = 0.5$, but the utility of the two sanitized logs is clearly different. Our results should thus be interpreted considering that the evaluation function is biased towards WNGEN.

It is also interesting to see if our findings are consistent with the analogous results for WNGEN reported in He and Naughton (2009). While the behavior of NCP as k varies is similar, the absolute values of NCP reported in He and Naughton (2009) were much smaller than ours; i.e. $NCP \cong 0.1$ (instead of $NCP \cong 0.7$), for $k = 2, 3, 4, 5, 6$. To gain more insights into this apparent discrepancy, we computed the statistics about the percentage of terms in the full AOL data set not matching any WordNet concept. There are 48,611,583 terms in all (including repetitions), with 23,502,849 matching concepts. Thus, 51.66% of terms cannot be generalized. Clearly, this is a lower bound for NCP with respect to the full AOL data set, because it is the minimum amount of distortion that any generalization strategy will incur. We hypothesize that the authors of He and Naughton (2009) in their experiments used a subset of queries that did not reflect the truly distribution of terms, or that they took into account only the nouns contained in the queries.

We finally study what happens to the queries of AOL user 4417749 if we apply k-anonymity privacy by WordNet generalization. First of all there are 48 queries (out of 224) whose terms do not match any WordNet concept. These queries are therefore removed (or, equivalently, generalized to the root 'entity'). For high values of k, the released queries correspond to very generic concepts. For low values of k, the released queries reflect the meaning of the original queries in a closer manner, although some queries are affected by the disambiguation issue; e.g., the preposition 'in' matches the concept 'indiana, hoosier_state, in'.

Turning to the ability of protecting the privacy of the user, we checked that all the queries containing the location of the user were either removed or generalized to harmless concepts, while the queries with the surname of the user were generalized to concept 'arnold, benedict_arnold' for low values of k, and then to concept 'bad_person' as k increased. As for the sensitive queries, in general the masking was effective even for very low values of k, although same generalizations were still revealing; e.g., through the WordNet concepts 'paranoia', 'aloneness, loneliness, lonesomeness, solitariness', and 'aneurysm, aneurism'.

5. Privacy protection and sensitivity to attacks

If we use k_θ-affinity with $\theta < 1$, then, by definition, the utility will be higher but the privacy will be lower than that ensured by k-anonymity using the same value of k. Because each query is treated in isolation, there is no theoretical guarantee of preventing identification of users based on the combination of multiple anonymized queries. However, we argue that although k_θ-affinity is primarily a means of achieving great utility increase with controlled privacy degradation, in practice it can also be used to improve the protection of the user's privacy beyond k-anonymity, keeping the utility constant. The clue is to observe that the same given number of queries can be released using a higher value of k. You may think of it as reverse engineering of the charts in Fig. 5. For instance, to release about 10% of queries we need to set $k = 3$ if we use k-anonymity and $k = 80$ if we use $k_{0.9}$-affinity.

Compared to k-anonymity, the search log sanitized using k_θ-affinity with a higher value of k will contain more infrequent query refinements and fewer relatively frequent unrefined queries. With the latter type of queries, the user privacy may be more exposed. For example, by entering in a search engine like Google the main research interest of the first author of this paper (i.e., web search) and his affiliation, one gets his name several times in the first results page. By contrast, if we only release queries with a high k-anonymity degree, it is likely that some queries that concur to the identification (such as the affiliation) will be removed. By replacing frequent unrefined queries with infrequent query refinements, we may thus reduce the risk of privacy breaches while releasing the same number of impressions.

One possible downside of using k_θ-affinity instead of k-anonymity is that there may be cases where infrequent identifying or sensitive queries may not be suppressed even if we choose a high value of k. Think of a query where private data co-occur with a key concept. For instance someone might enter a query formed by his/her credit card number and the type of card (e.g., '123XYZ visa'), just to see if such information can be found online. The query '123XYZ visa' will have an affinity to the query 'visa' close to 1, because it would be expanded using only the unigram 'visa' occurring 4166 times, and not the rare bigram '123XYZ visa'. The query '123XYZ visa' would thus be released under k_θ-affinity even for high values of θ and k (because the query 'visa' has a plain anonymity degree equal to 2168), as opposed to using strict k-anonymity. In order to avoid releasing unique user identifiers, our framework requires that any disclosed query must *not* contain terms linked to only one user, as mentioned in Section 3.2.

However, there is still the possibility of releasing queries containing key concepts together with some potentially harmful term entered by few (rather than unique) users, such as the name of a very specific location. To mitigate this risk, we could use more conservative policies such as requiring that each query term must be entered by a larger number of users. Of course, this would also limit the possibility of recovering harmless infrequent queries, but only to some extent because many infrequent queries consist of relatively frequent terms; i.e., it is their combination which is unfrequent. For instance, considering again the example in Fig. 4, if we required that each term be supported by at least 6 users (i. e., one hundredthousandth of the user population) then all the 21 queries in the example would still be released, because the least shared term is '2651' in query (15), entered by just six users.

Before concluding this section, it is interesting to briefly analyze whether the released query log is sensitive to attacks; e.g., if it is possible to force the anonymizer to release a private query such as someone's credit card number. The k_θ-affinity privacy model is at least as sensitive to attacks as plain k-anonymity, where it would suffice to create k identical queries issued from distinct users to cheat the anonymizer. A less obvious approach to bypassing k_θ-affinity privacy consists of simulating the situation described above when illustrating the credit card number example. It suffices to create two new queries, one formed by combining the private data that an attacker wants to be published and the content of some very frequent query, and the other consisting of the same private data but issued by a distinct user.

The k_θ-affinity privacy model may present other vulnerabilities to attacks, related to the release of rare queries. Considering again the example in Fig. 4, query (7) – 'cell phone case e815' – is in a generalized core of degree 6 and thus will be released under $5_{0.9}$-affinity, although it is associated with only one user. If the adversary saw a friend typing that query, he would be able to uniquely identify his friend's query set from the anonymized query log, a leak not possible under plain k-anonymity. How to combat such malicious activities is left for future work.

6. Conclusion

We presented a semantic approach to search log k-anonymization that leverages the affinity between frequent canonical concepts and their infrequent refinements. We showed that this approach is able to mask identifying queries while retaining a substantial amount of highly infrequent queries, to a much larger extent than allowed by plain k-anonymization or by k-anonymization under WordNet generalization.

Future work includes the use of different similarity measures and auxiliary information (external or log-based) to compute the affinity between queries, and an extension of the basic approach to deal with larger and more structured chunks of search log data than single queries, such as sessions and users.

Acknowledgments

We would like to thank two anonymous reviewers for their valuable comments and suggestions.

References

Adar, E. (2007). User 4xxxxx9: Anonymizing query logs. In *WWW workshop on query log analysis*.

Barbaro, M., & Zeller, T. (2006). *A face is exposed for AOL searcher no. 4417749*. New York Times.

Batagelj, V., & Zasversnik, M. (2003). *An O(m) algorithm for cores decomposition of networks*. CoRR cs.DS/0310049.

Batagelj, V., & Zaversnik, M. (2002). *Generalized cores*.CoRR cs.DS/0202039.

Bayardo, R. J., & Agrawal, R. (2008). Data privacy through optimal k-anonymization. In *Proceedings of the 2008 IEEE 21st international conference on data engineering (ICDE '05)* (pp. 217–228).

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Frieder, O., & Grossman, D. (2007). Temporal analysis of a very large topically categorized Web query log. *JASIST, 58*(2), 166–178.

Bendersky, M., & Croft, W. B. (2008). Discovering key concepts in verbose queries. In *SIGIR* (pp. 491–498).

Burghardt, T., Böhm, K., Guttman, A., & Clifton, C. (2010). Search-log anonymization and advertisement: are they mutually exclusive? In *CIKM* (pp. 1269–1272).

Carpineto, C., & Romano, G. (2013). Semantic search log k-anonymization with generalized k-cores of query concept graph. In *Proceedings of the 35th European conference on information retrieval (ECIR 2013), ECIR 2013 shared best paper award*.

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM CSUR, 44*(1), 1–50.

Carpineto, C., Romano, G., & Bernardini, A. (2012). Analyzing the behavior of professional video searchers using RAI query logs. In *Proceedings of the 10th international workshop on content-based multimedia indexing (CBMI)* (pp. 1–6). Annecy, France: IEEE.

Church, K., & Hanks, P. (1990). Word association norms, mutual information and lexicography. *Computational Linguistics, 16*(1), 22–29.

Collins-Thompson, K., & Callan, J. (2005). Query expansion using random walk models. In *Proceedings of the 14th ACM international conference on information and knowledge management (CIKM '05)* (pp. 704–711). New York, NY, USA: ACM Press.

Cooper, A. (2008). A survey of query log privacy-enhancing techniques from a policy perspective. *ACM TWEB, 2*(4), 1–26.
Feild, H. A., Allan, J., & Glatt, J. (2011). CrowdLogging: Distributed, private, and anonymous search logging. In *SIGIR* (pp. 375–384).
Ghinita, G., Tao, Y., & Kalnis, P. (2008). On the anonymization of sparse high-dimensional data. In *Proceedings of the 2008 IEEE 24th international conference on data engineering (ICDE '08)* (pp. 715–724).
Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature, 457*(3), 1012–1014.
Götz, M., Machanavajjhala, A., Wang, G., Xiao, X., & Gehrke, J. (2012). Publishing search logs: A comparative study of privacy guarantees. *TKDE, 24*(3), 520–532.
He, Y., & Naughton, J. F. (2009). Anonymization of setvalued data via topdown, local generalization. In *VLDB* (pp. 934–945).
Hong, Y., He, X., Vaidya, J., Adam, N., & Atluri, V. (2009). Effective anonymization of query logs. In *CIKM* (pp. 1465–1468).
Hong, Y., Vaidya, J., Lu, H., & Wu, M. (2012). Differentially private search log sanitization with optimal output utility. In *Proceedings of EDBT 2012* (pp. 50–61).
Hu, Y., Qian, Y., Li, H., pei, J., & Zheng, Q. (2012). Mining query subtopics from search log data. In *SIGIR* (pp. 305–314).
Jansen, B. J., Booth, D. L., & Spink, A. (2008). Determining the informational, navigational, and transactional intent of Web queries. *Information Processing and Management, 44*(3), 1251–1266.
Jiang, D., Pei, J., & Li, H. (2013). Mining search and browse logs for Web search: A survey. *ACM TIST, 4*(4), 1–57.
Korolova, A., Kenthapadi, K., Mishra, N., & Ntoulas, A. (2009). Releasing search queries and click privately. In *WWW* (pp. 171–180).
Kumaran, G., & Allan, J. (2007). A case for shorter queries, and helping users create them. In *NAACL-HLT* (pp. 220–227).
Kumar, R., Novak, J., Pang, B., & Tomkins, A. (2007). On anonymizing query logs via token-based hashing. In *WWW*.
LeFevre, K., DeWitt, D.J., & Ramakrishnan, R. (2006). Mondrian multidimensional *k*-anonymity. In *Proceedings of the 2008 IEEE 22nd international conference on data engineering (ICDE '06)*.
Liu, J., & Wang, K. (2013). Anonymizing bag-valued sparse data by semantic similarity-based clustering. *Knowledge and Information Systems, 35*(2), 435–461.
Navarro-Arribas, G., Torra, V., Erola, A., & Castella-Roca, J. (2012). User *k*-anonymity for privacy preserving data mining of query logs. *IPM, 48*, 476–487.
Seidman, S. (1983). Network structure and minimum degree. *Social Networks, 3*(5), 269–287.
Serdyukov, P., Dupret, G., & Craswell, N. (2014). Log-based personalization: The 4th web search click data (WSCD) workshop. In *Proceedings of the 7th ACM international conference on web search and data mining (WSDM 2014)* (pp. 685–686). New York, NY, USA: ACM Press.
Su, K. -Y., Hsu, Y. -L., & Sailard, C. (1991). Constructing a phrase structure grammar by incorporating linguistic knowledge and statistical log-likelihood ratio. In *ROCLING IV* (pp. 257–275).
Sweeney, L. (2002). *k*-Anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10*(5), 557–570.
Terrovitis, M., Mamoulis, N., & Kalnis, P. (2008). Privacy-preserving anonymization of set-valued data. In *Proceedings of VLDB 2008, Auckland, New Zeland* (pp. 115–125).
van Rijsbergen, K. (1979). *Information retrieval*. Butterworth-Heinemann.