# Semantic Search Log k-Anonymization with Generalized k-Cores of Query Concept Graph

Claudio Carpineto and Giovanni Romano

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

**Abstract.** Search log k-anonymization is based on the elimination of infrequent queries under exact (or nearly exact) matching conditions, which usually results in a big data loss and impaired utility. We present a more flexible, semantic approach to k-anonymity that consists of three steps: query concept mining, automatic query expansion, and affinity assessment of expanded queries. Based on the observation that many infrequent queries can be seen as refinements of a more general frequent query, we first model query concepts as probabilistically weighted n-grams and extract them from the search log data. Then, after expanding the original log queries with their weighted concepts, we find all the k-affine expanded queries under a given affinity threshold $\Theta$, modeled as a generalized *k-core* of the graph of $\Theta$-affine queries. Experimenting with the AOL data set, we show that this approach achieves levels of privacy comparable to those of plain k-anonymity while at the same time reducing the data losses to a great extent.

## 1 Introduction

Search log data are extremely valuable for a number of applications but they are subject to disclosure of personal and sensitive information of users. The infamous AOL search data release in 2006 has shown that replacing user-ids with random numbers does not prevent user identification [2], even when complemented with the removal of specific entities such as names, age, and address through ad-hoc techniques [14]). More principled anonymization methods have recently been developed that lie along the spectrum of trade-offs that exist between privacy guarantees and data utility; e.g., [1], [13], [11], [8]. Typically, increasing the limitations to information disclosure decreases the amount of useful data retained.

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 notion of k-anonymity [1], i.e., by requiring that for each query there are at least other k-1 equal queries entered by distinct users. In this way, there is at most 1/k probability to link a query to a specific individual. However, this method leads to extreme data loss (e.g., about 90% of distinct AOL search log queries were issued by a single user), with the deletion of a huge number of potentially useful and presumably harmless queries.

As an illustration, we give an example taken from the AOL search log data set, containing about 10 millions distinct queries submitted by about 650000 users from March to May 2006. We extracted and analyzed the 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. We noted that the query 'tiger woods' has been entered by 227 distinct users, while the overwhelming majority of queries (i.e., 327) are from a single user.

This example shows the main drawback of k-anonymization: simply requiring that there are at least two distinct users per query implies that 327 out of the 365 queries about Tiger Woods will be suppressed. The suppressed queries will in general contain useful information to identify the natural subtopics of the query 'tiger woods'. This phenomenon is illustrated in Figure 1. 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 k-anonymization techniques by a great deal and in a presumably safe manner. Hu et al. [12] have estimated that about 40% of search log queries follow a similar refinement pattern.

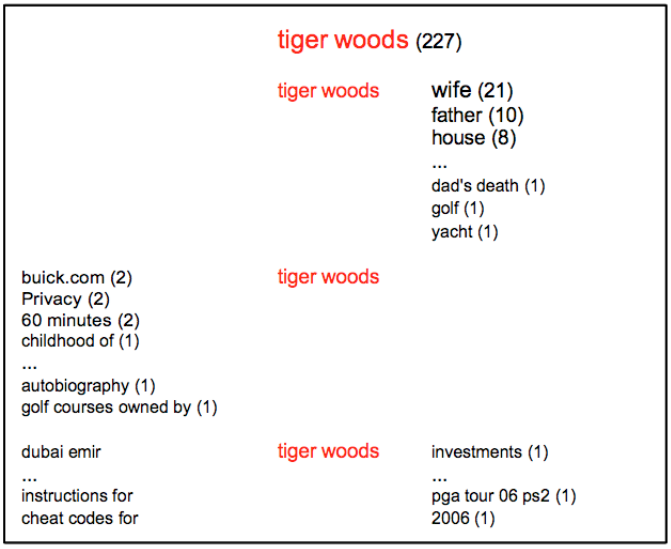


**Fig. 1.** A small sample of the 365 distinct AOL queries containing the string 'tiger woods', ordered by pattern and by frequency.

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

*A query log L satisfies k-anonymity under a $\Theta$-affinity threshold, noted as $k_\Theta$-anonymity, if for every query in L there exist at least k-1 $\Theta$-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 $k_\Theta$-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 AOL search log. Then we show that the computation of the set of $k_\Theta$-affine (expanded) queries can be traced back to a well known problem of graph theory. The solution consists of two steps: (a) building the graph of $\Theta$-affine queries, (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.

The main contributions of the paper 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.

The remaining of the paper has the following organization. After reviewing related work, 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 describe 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 plain k-anonymization. We finally conclude the paper.

## 2 Related Work

The concept of k-anonymity has been extensively studied in the database field, following the early work of Sweeney [18]. It is assumed that a subset of attributes are quasi-identifiers and a record is retained only if there are at least k-1 other records that share the value of each of those attributes. Afterwords, k-anonymity has been applied to search logs, where a query serves as the quasi-identifier [1].

Search log data, however, are fundamentally different from set-valued or relational data. Enforcing strict k-anonymity at the query level makes it hard to retain enough utility, due to data sparseness. One attempt at overcoming this difficulty is to build identical queries through generalization, e.g., by replacing two different queries with their common WordNet parent [10]. However, this

method is hampered by the limited coverage of WordNet, resulting in generic queries with little utility such as 'event', 'process', or just 'thing'.

The k-anonymity approach has been also applied at the user level, by clustering users and forming clusters of k users that are similar in terms of their data [11] . 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.

Another popular approach is differential privacy [13], providing a stronger privacy guarantee. It ensures that the amount of knowledge that an attacker can learn about a user is roughly insensitive to omitting or changing the user's search history. This is achieved by representing a search log with a query click graph and by injecting noise. However, differential privacy is penalized by a large data loss and distortion, with its utility being deeply questioned [8].

By contrast, our approach is based on retaining as much as possible of the original query log content and structure. One disadvantage of using k-anonymity at the query level is that there is no theoretical guarantee that this will prevent user identification through combination of multiple queries, although in this paper we show that semantic k-anonymity experimentally ensures good privacy levels.

## 3 Method Description

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

### 3.1 Query Concept Mining

Our approach is based on n-grams, because we aim to identify query portions corresponding to canonical concepts. The identification of key n-grams in search queries is usually carried out by combining statistical, probabilistic, and grammatical evidence ([15], [12]), although supervised machine learning techniques are also used [5]. We follow the former approach, adapting known techniques to the specific features of the anonymization task.

We first extract all unigrams, bigrams, and trigrams contained in the search log data. We do not consider larger n-grams because search queries are usually formed by few words. To efficiently 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. The AOL search log data set contains 3833549 distinct unigrams, 7365431 bigrams, and 8470381 trigrams.

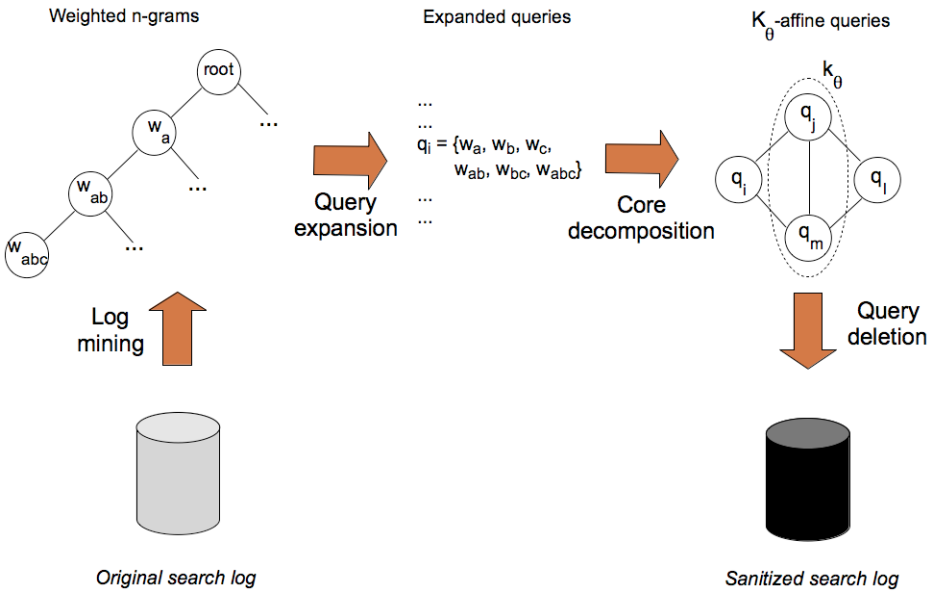


**Fig. 2.** Flowchart of semantic k-anonymization.

We then filter out rare n-grams, because they are often due to typos, or they may be user identifiers (e.g., social security number, credit card number, user name), or may correspond to overly specific concepts. We require that each n-gram be supported by at least ten users. We also rule out relatively frequent n-grams that are proper nouns of person or place, because they could be used to recover rare sensitive queries containing such n-grams. We finally remove those n-grams, such as 'is a', that are formed by words with little informative content; e.g., prepositions, articles, conjunctions, auxiliary verbs.

The remaining n-grams are finally weighted. Unigrams are weighted using their frequency, i.e., $W_x = \log_2(N+1)$, where N is the number of queries in which term $x$ occurs. In this way, we penalize (rather than reward) words associated with very few users. For bigrams and trigrams we use mutual information, a well known measure for word association. The bigram mutual information is defined as [7]:

$$W_{x,y} = \log_2 \left[ \frac{P(x,y)}{P(x) \cdot P(y)} + 1 \right] \quad (1)$$

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.[1] Such probabilities are estimated by relative frequency counts. The mutual information of a trigram is defined as [17]:

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

$$W_{x,y,z} \;=\; \log_2 \left[ \frac{P(x,y,z)}{P(x)\cdot P(y)\cdot P(z) + P(x)\cdot P(y,z) + P(x,y)\cdot P(z)} + 1 \right] \quad (2)$$

### 3.2 Query Concept Expansion and $\theta$-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 unigrams: $w_a$, $w_b$, $w_c$, $w_{ab}$, $w_{bc}$, $w_{abc}$. 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 [6].

We expanded all the queries in the AOL data set using the set of concepts extracted in the preceding step. Of the 10092308 distinct queries, 2695182 contains only one word, 2041455 two words, 2074258 three words, 1467863 four words, and 1813550 five or more words. After query expansion, we checked that 2683223 queries were not affected at all, i.e., they contained no valid concepts. The overwhelming majority of such queries consisted of one very rare unigram. We removed all these queries as well as those containing, in addition to some concept, one or more words associated with a unique user. This latter operation resulted in a large data loss, but it was necessary to ensure that queries with user identifiers would not be recovered due to the presence of key concepts. We were left with 5037881 queries. As their n-grams weights were comparable, we did not perform any normalization.

The expanded representations are used to assess the query affinity. We say that two queries $p$, $q$ are $\Theta$-affine if the cosine similarity of their expanded representations $p_E$, $q_E$, is greater than or equal to a threshold $\Theta$:[2]

$$Sim_{cosine}(p_E, q_E) \;=\; \frac{\overrightarrow{p_E} \;\cdot\; \overrightarrow{q_E}}{||\overrightarrow{p_E}||\;||\overrightarrow{q_E}||} \;\geq\; \Theta \quad (3)$$

We talk about $\Theta$-affinity instead of $\Theta$-similarity to emphasize the fact that there is a structural resemblance indicating a common concept, while the queries may be superficially quite different. Note also that in principle we could use different similarity functions. We could also use additional sources of information to assess the affinity between queries, such as clickthrough data or external taxonomic knowledge, although for infrequent queries this types of information are 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 $\Theta$-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 $k_\Theta$-affinity property can invalidate some queries that have been already evaluated. This problem can be solved by means of graph k-cores. This is discussed in the next section.

[2] Clearly, the $\Theta$-affinity relation is not transitive.

### 3.3 Generalized k-Cores of the Graph of $\Theta$-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 [16]. 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 $\Theta$-affine. If all $\Theta$-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 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, and that the degree is given by the number of distinct classes (rather than vertices) in the neighborhood.[3] An illustration is given in Figure 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 there are two nodes whose degree is equal to one despite being linked to vertices of the other two classes.

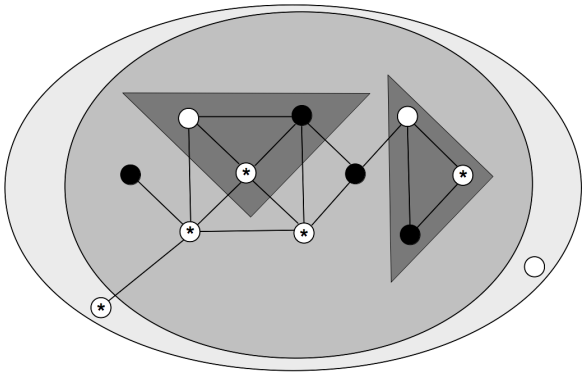

**Fig. 3.** Generalized cores of order 0, 1, 2 (for twelve vertices split in three classes).

To construct the graph of $\Theta$-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 $\Theta$, the graph is then formed by linking all the pairs of $\Theta$-affine queries.

To construct the generalized k-core of the graph of $\Theta$-affine queries, we modify the algorithm described in [4] to find k-cores. The algorithm in [4] is based on

[3] Note that this is different from p-cores [3], where the goal is to find the set of vertices with a property value larger than a given threshold.

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. While a detailed description of both the basic and the generalized algorithms are outside the scope of this paper, we highlight that the main difference is that in the generalized version we represented the neighbors as query-user pairs, updating the degree of a vertex only when all the set of queries associated with a user becomes empty. With suitable data structures, the generalized algorithm has the same complexity as the ungeneralized one.

## 4 Evaluation

In this section we study the trade-off between privacy and information loss when using semantic k-anonymization. Given our limited computational resources, it is not practical for us to experiment with the full AOL data set. This would require to compute one ranking for each query, given a certain value of $\Theta$, and then to process the resulting graph for finding the k-cores. Depending on the value of $\Theta$, the number of edges may grow large, thus slowing down the algorithm for finding k-cores.

To overcome this problem, we develop an approximate procedure. We first randomly select a subset of the 5 millions of AOL queries (after pruning). Then, for each random query $q$ and a specified value of $\Theta$, we build a graph $G_q$ of $\Theta$-similar queries centered around $q$, by iteratively computing the set of neighbors (for the given value of $\Theta$) 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 AOL graph) formed by the corresponding queries. We next compute the 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 AOL graph G, because the omitted vertices might only 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 Figure 4 we show the k-cores of the graph obtained for the query 'sanyo 3100 cell phone case', with $\Theta = 0.9$, $Vmax = 2000$.[4] We found 21 queries, listed in the caption of Figure 4. All but one queries 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 Figure 4, the vertices in a same k-core are depicted with the same tone of gray

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

(the darker the tone, the higher the degree). Note that in this example, using semantic k-anonymization 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', or even 'cell phone case', which were not selected due to lower affinity.

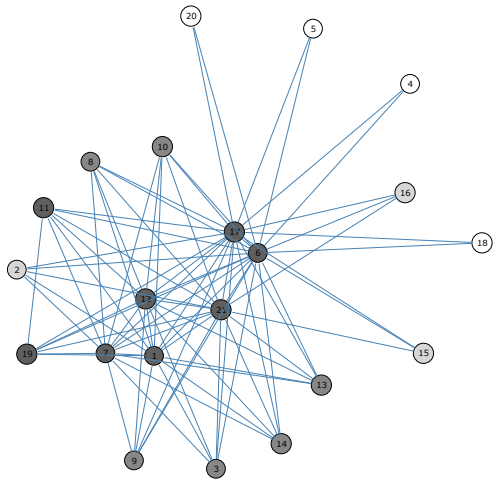


**Fig. 4.** Generalized k-cores of the graph originated from the query 'sanyo 3100 cell phone case'. The complete list of queries is the following ('cell phone case' being abbreviated with cpc): 1) 3100 cpc, 2) sanyo 3100 cpc, 3) blackberry cpc, 4) playboy cpc, 5) coach cpc, 6) cpc 2, 7) cpc e815, 8) 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.

The next step is to decide how many random queries to use. We experimented with increasing random samples until the results stabilized. We found that 5000 random queries ensured representative results. 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.

In Figure 5 (a) we show how the number of released queries (in percentage) varies as a function of k, for three values of $\Theta$: 1, 0.9 ,0.8. Note that for $\Theta = 1$, we get exactly the number of queries released under plain k-anonymization. The three methods are denoted, respectively, as KA, SKA ($\Theta = 0.9$), and SKA ($\Theta = 0.8$). The results were averaged over all the queries generated using the random sample.[5] The figure clearly shows the trade-off between k-anonymity and data release associated with each privacy policy. Using the semantic method, the

[5] The percentage of released queries under KA for k=2 is about about 20% instead of 10% because we used a pruned version of the full AOL data set.

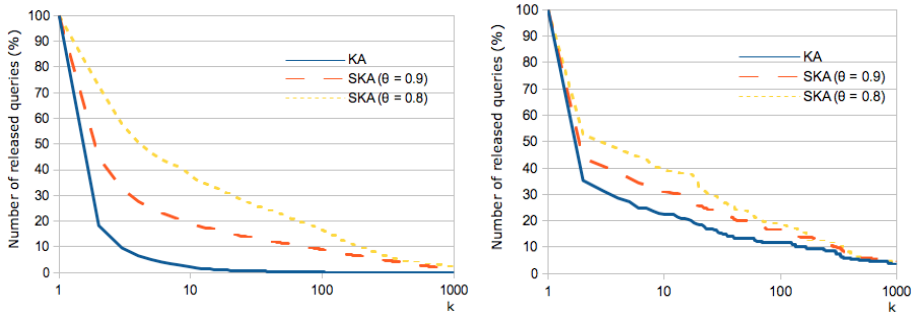


**Fig. 5.** Proportion of released queries as a function of k for a subset of the AOL data set (a) and for AOL user 4417749 (b), under plain k-anonymity (KA) and semantic k-anonymity (SKA). The x axis is logarithmic.

gain in terms of released queries is massive for all values of k. Furthermore, a comparison between the plain and semantic k-anonymization 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.

As the value of $\Theta$ increases, the set of released queries becomes by definition larger, but there are of course more risks of privacy breach. Of particular interest is the identification of an individual from multiple queries. To evaluate this aspect, we consider the AOL user 4417749, identified by the New York Times [2]. The 224 distinct queries entered by User 4417749 were analyzed using plain and semantic k-anonymity. In Figure 5 (b) we show the proportion of query released for User 4417749 under the different privacy policies. Similar to Figure 5 (a), there is a tangible growth of released queries as $\Theta$ becomes smaller. The main difference is that in Figure 5 (b) the plots are closer because User 4417749 entered fewer unique queries than the average AOL user. On closer inspection, we found that all highly identifying queries referencing the user's surname were discarded for $k > 1$ under all privacy policies, while some of the queries referencing the user location required slightly higher values of k to be discarded, especially using SKA ($\Theta = 0.8$); e.g., 'gwinnet animal shelter' or 'gwinnet humane society'. Without these queries, it is virtually impossible to identify the user. Other possibly sensitive queries such as 'mini strokes' or 'panic disorders' were discarded for higher values of k, comparable across all methods. A comparison between Figure 5 (b) and the analogous plot shown in [9] under alternative privacy policies suggests that our method released a larger percentage of queries, both for $\Theta = 0.8$ and $\Theta = 0.9$.

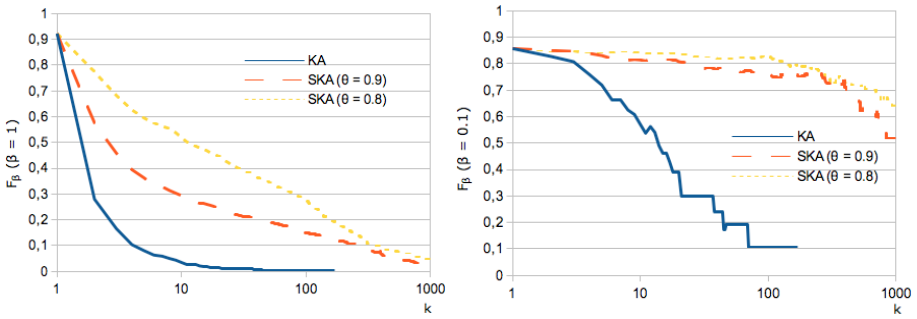


**Fig. 6.** $F_\beta$ performance of anonymization methods on queries with sensitivity labels, for $\beta = 1$ (a) and $\beta = 0.1$ (b). The x axis is logarithmic.

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 considered again the 5000 random queries used above 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 the 5000 queries according to KA, SKA ($\Theta = 0.9$), and SKA ($\Theta = 0.8$), 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 $\geq$ 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 $F_\beta$ measure, that combines precision and recall with a weighting factor $\beta$. 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 $\beta \leq 1$. In Figure 6 we show the performance of the three methods for $\beta = 1$ and $\beta = 0.1$. The KA curve is incomplete because for k $>$ 172 no queries were released under plain k-anonymity.

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

## 5 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. Future work includes the use of different similarity measures and auxiliary information (external or log-based) to compute the affinity between queries, a comparison with other semantic k-anonymization techniques (e.g., based on taxonomic generalizations), and an analysis of the sensitivity of our privacy scheme to attacks.

## References

1. E. Adar. User 4xxxxx9: Anonymizing query logs. In *WWW Workshop on Query Log Analysis*, 2007.
2. M. Barbaro and T. Zeller. A face is exposed for aol searcher no. 4417749. *New York Times*, 2006.
3. V. Batagelj and M. Zaversnik. Generalized Cores. *CoRR cs.DS/0202039*, 2002.
4. V. Batagelj and M. Zaversnik. An O(m) Algorithm for Cores Decomposition of Networks. *CoRR cs.DS/0310049*, 2003.
5. M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *SIGIR*, pages 491–498, 2008.
6. C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM CSUR*, 44(1):1–50, 2012.
7. K.W. Church and P. Hanks. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.
8. M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Publishing Search Logs: A Comparative Study of Privacy Guarantees. *TKDE*, 24(3):520–532, 2012.
9. Feild H, J. Allan, and J. Glatt. CrowdLogging: distributed, private, and anonymous search logging. In *SIGIR*, pages 375–384, 2011.
10. Y. He and J. F. Naughton. Anonymization of SetValued Data via TopDown, Local Generalization. In *VLDB*, pages 934–945, 2009.
11. Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri. Effective anonymization of query logs. In *CIKM*, pages 1465–1468, 2009.
12. Y. Hu, Y. Qian, H. Li, J. pei, and Q. Zheng. Mining Query Subtopics from Search Log Data. In *SIGIR*, pages 305–314, 2012.
13. A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and click privately. In *WWW*, pages 171–180, 2009.
14. Ravi Kumar, Jasmine Novak, Bo Pang, and Andrew Tomkins. On anonymizing query logs via token-based hashing. In *WWW*, 2007.
15. G. Kumaran and J. Allan. A Case for Shorter Queries, and Helping Users Create Them. In *NAACL-HLT*, pages 220–227, 2007.
16. S. Seidman. Network structure and minimum degree. *Social Networks*, 3(5):269–287, 1983.
17. K.-Y. Su, Y.-L. Hsu, and C. Sailard. Constructing a Phrase Structure Grammar by Incorporating Linguistic Knowledge and Statistical Log-Likelihood Ratio. In *ROCLING IV*, pages 257–275, 1991.
18. L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.