Using search results to microaggregate query logs semantically

Arnau Erola and Jordi Castellà-Roca

Departament d'Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Spain
{arnau.erola,jordi.castella}@urv.cat

Keywords: Privacy, web search, microaggregation, k-anonymity, query logs, semantics, semantic microaggregation.

Abstract. Query log anonymization has become an important challenge nowadays. A query log contains the search history of the users, as well as the selected results and their position in the ranking. These data are used to provide a personalized re-ranking of results and trend studies. However, query logs can disclose sensitive information of the users. Hence, query logs must be submitted to an anonymization process to guarantee that: a) no sensitive information can be linked to an identity; b) the analysis of the anonymized data produces similar results than the original data, i.e. minimize data distortion. Latest anonymization approaches utilize microagreggation, a statistical disclosure control technique that provides a privacy comparable with k-anonymity, attempting to minimize the data distortion. We propose a new method that uses search results to optimize microaggregation, providing more data reliability than the existing methods.

1 Introduction

The query logs stored by Web Search Engines (WSE) contain valuable information for researchers and marketing companies, as they allow the study of users' behavior or any changes in users' trends [1]. Furthermore, WSE can use query logs to improve users' experience by providing advanced functionalities such as personalizing search results (disambiguating terms and predicting user interests), autocompleting search terms, or correcting spelling mistakes [2].

However, the attractiveness of the advanced functionalities obtained from storing query logs can be counteracted by the privacy problems they present. Queries can contain sensitive information (e.g. religion, sexuality, politics etc.) and can disclose the identity of the user (e.g. a query contains

the passport number). Note that a query itself can contain information about several issues, for instance, a simple query such as *Drug Clinic Portland* is probably disclosing that the user lives in Portland and has some problems with drugs. Thus, queries, like microdata, can be classified into the following categories, which are not mutually exclusive, depending on their content:

- **Identifier queries**. The queries univocally identify the query owner. For example, the query contains the passport number, the credit card number or the full name. Such queries are usually removed because they are not valuable for advanced functionalities and constitute a high privacy threat.
- **Quasi-identifier queries**. These queries cannot identify a user. However, in combination or with the help of external information (for instance, the information of public databases), they can univocally identify a user. The zip code, age, first name, etc. are some examples.
- **Confidential queries**. The queries contain sensitive information of the users. For example, queries related with religion, illnesses, investments, etc.
- **Non-confidential queries**[1]. The queries do not contain sensitive information. Country names and favorite sports can be some examples. However, they can also constitute an identity disclosure risk because in combination they can form a quasi-identifier.

Accordingly, query logs must be submitted to an anonymization process prior to their publication. It involves some data modifications which reduce the probability of disclosing personal information, but at the same time they reduce the data utility. This is a trade-off between utility and privacy, which implies that the greater the privacy, the greater the information loss will be. However, although data have been anonymized, personal information can be inferred from the remaining data. The AOL's case [4], where some individuals were re-identified, is an example.

This study is aimed at exploring the preservation of users' privacy in the dissemination of query logs. To that end, we take the privacy criterion of offering a desired privacy protection. Thus, the main objective is to attain it while maximizing the data utility.

[1] Note that a user can consider some information private or not according to her beliefs, i.e. whereas a user can consider her religion a public issue, another user can consider this information private. Determining what information is private or not is out of the scope of this paper. For this reason, we consider that all the information has the same importance and is private, as is made in [3].

As some authors have stated, the utility in textual data, as query logs are, is related with the semantics [5, 6]. To that end, we propose an optimization of the microaggregation method for query logs that groups the users with more similar targets, with the help of the selected results.

The paper is organized in the following sections: Section 2 introduces the related work. The notation used and the similarity metric are presented in Section 3. In Section 4, we detail our proposal and in Section 5 we discuss the evaluation measures. Finally, we present the obtained results and the conclusions in Sections 6 and 7, respectively.

2 Related work

Microaggregation [7] is a Statistical Disclosure Control (SDC) technique used to prevent against from re-identification. Although it was initially proposed to protect microdata, it can be adapted to protect query logs [8–11]. It works by creating clusters of at least k users and by replacing their logs by a new log called centroid, which is representative of the group. The centroid is usually created with random queries of the users in each cluster. The privacy achieved is comparable to k-anonymity [12], for the fact that there are at least k users with the same log. Data utility (data homogeneity) depends on the clustering method, where the optimum solution is to cluster the users with more similar queries [13]. As k is an input parameter of the method, the only thing that can be done to optimize microaggregation is to maximize data homogeneity [14].

The first approaches to anonymize query logs, remove or delete queries in order to avoid identity disclosures [15–17]. However, they only achieve an acceptable level of privacy if the data are deeply perturbed [15].

Recently, microaggregation has been used to anonymize query logs [8–11] and to maximize data utility. Authors in [9] use the edit distance to compute queries' similarities. However, edit distance only takes into account the spelling similarities of the queries, and different queries can point to the same target. For instance, both *Freddie Mercury* and *Queen singer* refer to information about Freddie Mercury.

Authors in [3] use semantic taxonomies to anonymize query logs for the first time. They generalize query terms using WordNet [18]. However, Wordnet is a general-purpose dictionary that can only deal with simple concepts, and proper names are not included. For this reason, a lot of original queries are not published, thus losing a large amount of information. Erola et. al [10, 11] introduces the concept of semantic microaggregation, which consits in clustering users according to what they were looking for.

In order to interpret queries, it retrieves the semantics of query terms by using the ODP [19] as a knowledge source. Note that neither WordNet nor ODP can address sentences, only terms.

Retrieving the semantics of query terms can produce a change or loss in meaning. For example, the ambiguous term *mercury* acquires different meanings in the queries *mercury planet*, *mercury element* and *Mercury*, *Freddie*. Moreover, the other query terms, such as *Freddie*, can also be ambiguous by themselves. Consequently, clustering query logs according to the semantics of the query terms can increase the information loss. A criterion that can help in achieving better data homogeneity may be to cluster queries searching for the same information. By pursuing this idea, we propose to use the selected results instead of queries in order to retrieve the real interest of the user (query meaning).

3 Notation and background

For notation purposes, let $U = \{u_1, \ldots, u_n\}$ be a set of n users, being $|U| = n$. Their respective set of queries (logs) are $Q = \{Q(u_1), \ldots, Q(u_n)\}$, where $Q(u_i) = \{q(^1_{u_i}), \ldots, q(^{m_i}_{u_i})\}$ are the set of queries of user u_i and $|Q(u_i)|$, their number.

After the clustering process, the set of clusters $Z = \{z_1, \ldots, z_\gamma\}$ is obtained, being $z_i = \{u(^1_{z_i}), \ldots, u(^{k'}_{z_i})\}$ the set of users that belong to cluster z_i, where $2k > k' \geq k$. Let $Cent_{z_i}$ be the centroid of cluster z_i and $|Cent_{z_i}|$ its size, i.e. the number of queries that it contains.

3.1 ODP Similarity metric

ODP [19] is the most widely distributed database of web content classified by humans. ODP directory is hierarchically structured in themes, and web sites are hand-classified on them, thus forming the ODP tree. Figure 1 shows an example of this classification. Observe that categories can be divided into levels, being the root category the most generic one.

In order to measure the semantic similarity of two queries classified in the ODP hierarchy, we use the measure ODP_{sim} introduced in [11]. To that end, we define a working depth level $l \in \{1, \ldots, L\}$, being L the maximum depth considered. So in Figure 1, when $l = 1$, we are referring to the root level, i.e. *Recreation*; and the next level is *Recreation* : *Autos* :; when $l = 2$, and so on.

Let $C_l = \{c(^1_l), \ldots, c(^{m_l}_l)\}$ be the set of possible categories at depth level l in the ODP tree. Let u_i be a user and $Q(u_i) = \{q(^1_{u_i}), \ldots, q(^{p_i}_{u_i})\}$,

```
Open Directory Categories  (1-5 of 100)
   1. Recreation: Autos: Makes and Models: Audi (18)
   2. Recreation: Autos: Makes and Models: Audi: Clubs (8)
   3. Recreation: Autos: Makes and Models: Audi: A4 (5)
   4. Recreation: Autos: Makes and Models: Audi: TT (4)
   5. Recreation: Autos: Makes and Models: Audi: Clubs: United Kingdom (2)
```

Fig. 1: Example of ODP query result.

her set of queries. We denote $C_l(u_i) = \{c_l(^1_{u_i}), \ldots, c_l(^{p_i}_{u_i})\}$ as the set of categories of u_i at level l, being $c_l(^{j_i}_{u_i})$ the category corresponding to the classification of the query $q(^{j_i}_{u_i})$ in the ODP hierarchy at depth level l. $|C_l(u_i)| = p_i$ is the number of categories of user u_i, and $|c_l^{j_i}(u_i)|$ is the number of queries classified in the category $c_l^{j_i}$ that u_i has.

Thus, given two users u_i and u_j, ODP_{sim} is defined as:

$$OPD_{sim}(u_i, u_j) = \sum_{l=1}^{L} \{|c_l| : c_l \in \{C_l(u_i) \cap C_l(u_j)\}\} \tag{1}$$

where $|c_l|$ is the number of categories that users u_i and u_j have in common - the minimum between $|c_l(u_i)|$ and $|c_l(u_j)|$ at level l -. Note that ODP_{sim} returns a similarity value, whose weight is maximum when the matching (similarity) is maximum.

4 Our proposal in detail

Previous semantic microaggregation algorithms split queries into terms, thus changing the semantics of the queries and producing an information loss in the protected logs. We propose a semantic microaggregation method to solve this problem. In the first step, we find the semantics of all the queries by submitting them to a WSE (in our case, Google). Then, we classify the search results in the ODP hierarchy and we cluster the users with more common real interests, thus reducing the distortion of the centroid. To that end, we present our system divided in two parts:

1. Semantic interpretation: we obtain the semantics of the query by submitting it to the WSE.
2. Microaggregation using ODP: we cluster the logs according to the classification of the queries and we then aggregate them.

4.1 Semantic interpretation

Using search results of a WSE to retrieve meanings is not new, they were already used in [20, 21]. However, this is the first time that a WSE is used to anonymize query logs, to the best of our knowledge.

Roughly speaking, this part of the method is a pre-process of the queries for the microaggregation. For each query in the users' logs, we want to obtain the *url* that represents the interest of the user - the query semantics -. Obviously, if this process is made by a WSE, it can select the *url* clicked by the user. Analogously, if we have a query log that contains the *url*s selected by the users, we can also use them.

However, this situation does not always happen. AOL logs are anonymized (most of the *url*s are deleted) and a significant amount of the *url*s are obsolete (AOL logs date from 2006). Therefore, we must redo the search to obtain the *url*s that users had probably selected. To that end, we take into account the work presented in [22], which states that a 68% of the users of WSEs click a search result within the first page of results. Even more relevant is the fact that a 92% of the users click a result within the first three pages of search results. Note that WSEs attempt to rank the search results according to the users' interests, i.e. the links which could probably be more interesting for the users appear in the first positions of the rank.

Accordingly, we consider that the first *url*s returned by the WSE are those more reliable according to the interests of the users. Thus, for each query in the users' logs, the query is searched in the WSE, and the first ranked *url*, which is not a sponsored link or a wikipedia link, is stored. Hereafter, we consider that $URL = \{URL(u_1), \ldots, URL(u_n)\}$ are the set of results selected from the Google searches, being $URL(u_i) = \{url(^{1}_{u_i}), ..., url(^{p_i}_{u_i})\}$ the set of *url*s of user u_i, and $url(^{j_i}_{u_i})$ the selected result when query $q(^{j_i}_{u_i})$ was submitted.

The *url*s are cut at domain name because ODP does not contain all the *url*s of every domain. *Wikipedia* links are not stored because they can contain a large amount of topics and the *url wikipedia.org* is always classified in the same ODP category.

4.2 Microaggregation using ODP

The first step is to obtain the ODP categories for all the *url*s corresponding to the queries in the logs, i.e. the set of categories $C_l(U) = \{C_l(u_1), \ldots, C_l(u_n)\}$ can be obtained at level l. This process is described in Algorithm 1.

Next, we microaggregate the users trying to maintain the maximum data homogeneity. We use equation 1 (see section 3) to calculate the similarity between users, where the maximum weight means maximum matching. Thus, we take two steps:

1. Clustering: users are clustered into groups according to their semantics.
2. Aggregation: we calculate the centroid of each cluster and then replace the users' logs with the centroid.

Algorithm 1 Algorithm to classify the *url*s in the ODP

Require: the maximum depth l for the ODP categories
Require: the set of users $U = \{u_i, \ldots, u_n\}$
Require: the set of *url*s $URL(u_i) = \{url(^1_{u_i}), \ldots, url(^{p_i}_{u_i})\}$ of each user u_i
Ensure: the set of categories $C_l(u_i) = \{c_l(^1_{u_i}), \ldots, c_l(^{p_i}_{u_i})\}$ of each user u_i
  for $u_i \in \{u_1, \ldots, u_n\}$ **do**
    for $url(^j_{u_i}) \in URL(u_i) = \{url(^1_{u_i}), \ldots, url(^{p_i}_{u_i})\}$ **do**
      obtain the category $c_l(^j_{u_i})$ at depth l for $url(^j_{u_i})$ using ODP;
    end for
  end for
  return C_l

Clustering Algorithm 2 describes the clustering process. Briefly, it groups the k most similar users of U. Then, if there are enough users to form a new cluster ($k \leq |U|$), the selected users are deleted and a new cluster is formed with the remaining users of U. It continues until there are not enough users to form a new cluster. If there are users remaining, they are added to the last formed cluster.

Aggregation Z_i are sets of users and $Q(z_i)$ contains the queries of all users in Z_i. $C_l(z_i)$ is the set of categories of $Q(z_i)$. We use expression 2 to determine which queries of $Q(z_i)$ have to appear in the centroid. We obtain the number of queries classified in category c_l^t, which will appear in the centroid using equation 2.

$$|c_l(^t_{Cent_{z_i}})| = \frac{\sum_{i=1}^{k'} \frac{|c_l(^t_{u^i_{z_i}})|}{|Q(u^i_{z_i})|}}{k'} \times |Cent_{z_i}| \tag{2}$$

being

$$|Cent_{z_i}| = \frac{|Q(z_i)|}{k'} \tag{3}$$

Therefore, $Cent_{z_i}$ contains $|c_l(^t_{Cent_{z_i}})|$ queries of the category c_l^t. These queries are selected randomly among those that are in Q_{z_i} and are classified in c_l. As these queries belong to the same category, we can assume they have the same semantics.

Algorithm 2 Algorithm for computing the clusters $Z = \{z_1, \ldots, z_\gamma\}$ of users

Require: the maximum depth l for the ODP categories
Require: the set of users $U = \{u_1, \ldots, u_n\}$
Require: the set of categories $C_l(u_i) = \{c_l(^1_{u_i}), \ldots, c_l(^{p_i}_{u_i})\}$ for each user $u_i \in U$
Require: the clusters size k
Ensure: the clusters $Z = \{z_1, \ldots, z_\gamma\}$ of users for $\gamma = \lfloor n/k \rfloor$
  while $|U| \geq k$ **do**
    obtain the cluster z with $2k - 1 > |z| \geq k$ users using Algoritm 3 and $C_l(U)$;
    remove the users $u_i \in z$ from U;
    add z to the set Z
  end while
  return C_l

Algorithm 3 Algorithm for computing a cluster z

Require: the set of categories $C_l(U)$
Require: the clusters size k
Ensure: a cluster z with $2k > |z| \geq k$ users
  $z \leftarrow \emptyset$
  obtain $u_i, u_j | \max ODP_{SIM}(u_i, u_j)$
  add u_j, u_i to the set z;
  $C_l(u_i) = C_l(u_i) \cup C_l(u_j)$
  remove $C_l(u_j)$
  while $(|z| < k)$ **and** $|C_l(U)| > 0)$ **do**
    obtain $u_j | \max ODP_{SIM}(u_i, u_j)$
    add u_j to the set z;
    $C_l(u_i) = C_l(u_i) \cup C_l(u_j)$
    remove $C_l(u_j)$
  end while
  if $(|C_l(U)| < k)$ **then**
    add all $u_j \in C_l$ to z;
  end if
  return z

5 Evaluation

We have evaluated our proposal by using a random set of 840 users selected from the AOL files, where each one has submitted between 400 and 600 queries. In total, we have near $400,000$ queries. Simulations are performed for cluster sizes k between 2 and 30, and for depth level $l = 5$. We have not considered levels above 5 because only 35% of the queries can be classified beyond this level [11].

We have chosen Google to perform our tests because of its popularity, but we could have chosen any other web search engine to carry out our experiments.

Next, we present two measures to evaluate the utility and the privacy obtained with the microaggregation.

5.1 Utility and Privacy

The result of the microaggregation process is a log that contains queries of all the users clustered together. Its utility is conditioned by the data homogeneity, i.e. how similar the original log and the centroid are. This similarity is not conditioned by the spelling mistakes or alternative queries but by the meaning of the queries. Hence, we evaluate the utility of the queries according to their classification in the ODP tree.

We have used three classic similarity coefficients to evaluate the data utility (ODP classification homogeneity): the Jaccard coefficient, the Sokal and Sneath coefficient and the Dice coefficient. Table 1 shows their expressions.

Coefficient	**Formula**
Jaccard	$\frac{\lvert S_1 \cap S_2 \rvert}{\lvert S_1 \cup S_2 \rvert}$
Sokal and Sneath	$\frac{\lvert S_1 \cap S_2 \rvert}{2 \times (\lvert S_1 \rvert + \lvert S_2 \rvert) - 3 \times \lvert S_1 \cap S_2 \rvert}$
Dice	$\frac{2 \times \lvert S_1 \cap S_2 \rvert}{\lvert S_1 \rvert + \lvert S_2 \rvert}$

Table 1: Similarity coefficients between two sets S_1 and S_2

Thus, for cluster sizes (k) between 2 and 30 and maximum level l, we have calculated the three coefficients between the original logs and their corresponding centroids, which are generated with our proposal,

the method that also uses ODP to classify the query terms [11], and a random scheme (microaggregation with random clustering). The performance of our scheme can be compared with the performance of two other schemes by dividing the coefficients (see equation 4), where the utility of our proposal is the numerator and the utility of the other proposals, the denominator. [11] has been selected for comparison as it is the only anonymization method for query logs, which uses the semantics of the queries to cluster the users.

$$Utility_{Coef_y}(C_{prop}, C_x) = \frac{\sum_{i=1}^{n} Coef_y(C_{orig}(u_i), C_{prop}(u_i))}{\sum_{i=1}^{n} Coef_y(C_{orig}(u_i), C_x(u_i))} \quad (4)$$

where C_{orig} are the logs with the original categories, C_{prop} are the generated logs with our proposal, and C_x are the generated logs with the proposal [11] or the random proposal. $Coef_y$ is the coefficient chosen among the three proposed ($Jaccard, Sokal\ and\ Sneath, Dice$), which ranges from 0 to 1.

Regarding privacy, a log can be considered private if it does not allow to uniquely identify any user. Microaggregation [12] provides k-anonymity at user level, so the privacy of a user is guaranteed by $k-1$ other users.

Another thread of particular interest in query logs privacy is the disambiguation of a user. In a microaggregated query log, a user can be disambiguated if her queries can be isolated. Disambiguation means that the queries of a user can be distinguished from the queries of the other users. This is possible if the queries are about different topics, they are in different languages, etc. The isolated queries can contain enough information to identify the user, or they can help to do it.

We propose to evaluate the linkability between the original logs and the microaggregated logs by using the previous coefficients with the queries as input data, that is, how many queries of the original log also appear in the microaggregated log. The less common queries they have, the more ambiguity we will achieve, since the two logs are less similar syntactically. This measure is expressed by equation 5. Note that the linkability achived by our proposal is now the denominator, and the linkability of the other proposals is the numerator.

$$Linkability_{Coef_y}(Q_{prop}, Q_x) = \frac{\sum_{i=1}^{n} Coef_y(Q_{orig}(u_i), Q_x(u_i))}{\sum_{i=1}^{n} Coef_y(Q_{orig}(u_i), Q_{prop}(u_i))} \quad (5)$$

where Q_{orig} is the set of original query logs, $Q_{prop}(i)$ are the logs generated with our proposal and $Q_x(i)$ are the query logs generated with the proposal [11] or the random proposal.

6 Results

Figure 2 shows the utility outperformance obtained by using our method based on the method presented in [11] and the random microaggregation (microaggregation with a random clustering). The three similarity coefficients used provide close results, thus supporting the validity of the obtained results. When we compare our method with [11] (figure 2 (a)), the former achieves a maximum of 64% utility improvement when $k = 15$ (according to the Sokal and Sneath coefficient), althought it improves the utility in all cluster sizes. Thus, we can argue that it is easier to match categories using *urls* than query terms, as the latter are dificult to be interpreted or classified in the knowledge base correctly. Moreover, since categories can represent the conceptualizations of the *urls*, this approach enables a semantically-choerent clustering of logs.

We can also observe in figure 2 (a) that the outperformance is higher in cluster sizes between 10 and 20, increasing from 2 to 10 and decreasing above $k = 20$. We presume that working with larger sets of users could outperform the results, because we could presumably find more similar users when the number of users is greater.

Figure 2 (b) shows that the outperformance of our method is even better compared with a random microaggregation. However, the most interesting is to observe that the method [11] achieves better data utility than the random microaggregation. This is due to the fact that some queries are composed by few terms, and their interpretation in the knowledge base can be correct. In our dataset, 40% of the queries have 3 or less terms. Thus, we can determine that the interpretation of query terms produces less precise conceptualizations of the queries, but it still works better than a random microaggregation.

Figure 3 shows the linkability reduction of the queries using our proposal, based on the proposal [11] and a random microaggregation. We can

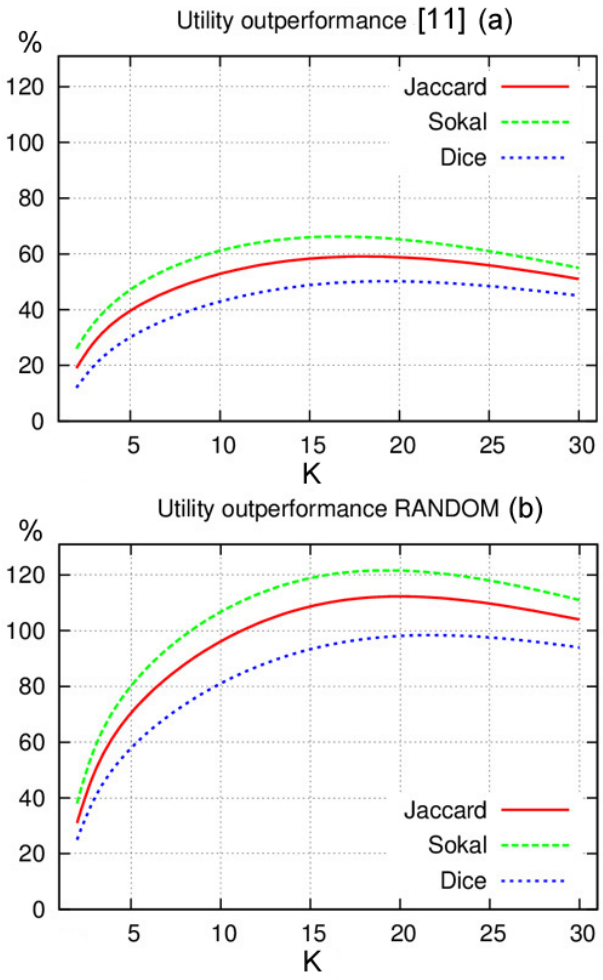


Fig. 2: Utility outperformance.

observe that our method achieves a significant reduction of the number of real queries that can be found in the original logs and in the centroid. In the proposal [11], the reduction is about 30% in median, and 40% in the random microaggregation. Thus, it seems clear that our method reduces the risks of privacy disclosures.

It is interesting to observe that [11] only obtains slightly better results than the random microaggregation. This is an effect of the clustering part, which has group users with few common interests, and the aggregation part, which has selected queries at random without taking into account their frequencies.

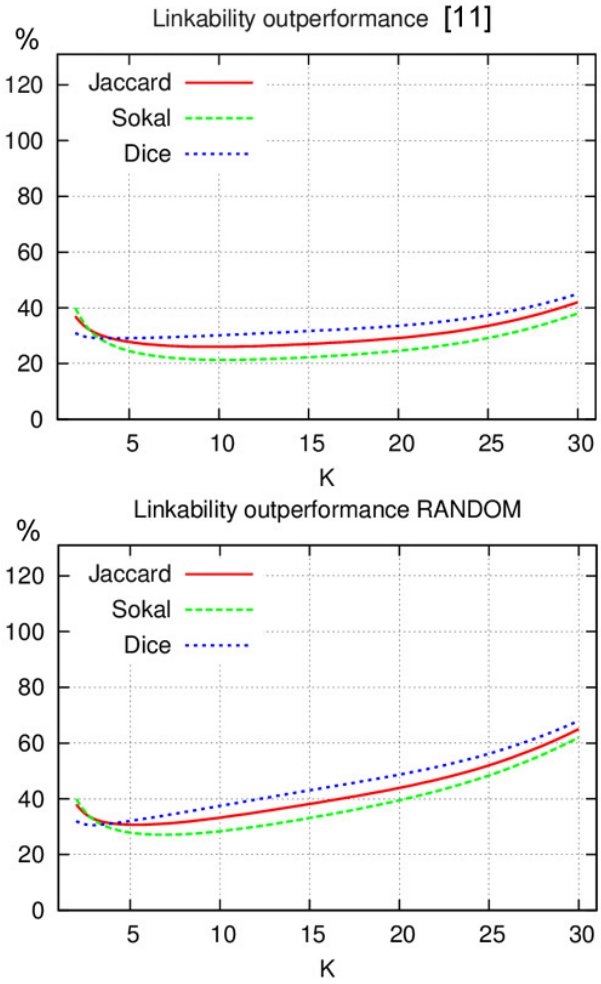


Fig. 3: Linkability outperformance.

7 Conclusions

Query logs need to be anonymized in order to guarantee the privacy of the users. However, this process involves data utility loss. Query logs utility is related with the semantics of the queries, which represent the users' interests. Existing privacy-protection proposals use query terms in order to extract their semantics. However, as we stated in the results section, this reduces the homogeneity of the clustered data.

We have proposed a microaggregation method that uses a WSE and ODP in order to interpret the semantics of the queries. Hence, the in-

terpretation of the real meaning of the queries is more precise. To the best of our knowledge, it is the first time that search results are used to anonymize query logs.

As we have seen in the results section, our method reduces the introduced data perturbation due to the fact that we do not lose information by classifying the queries, but we provide more data utility than the previous version [11] and a random microaggregation. Moreover, generated logs disclose less real information about the user, thus providing more privacy.

This study has established that it was a flaw to interpret queries in a knowledge base, because in some cases, the real interest of the user cannot be obtained. It is not only due to the terms ambiguity, but due to the limitations of the knowledge base. The use of *url*s in order to interpret the user's interests give us the possibility to explore new knowledge representation forms. One possibility is to use a bipartite graph of queries and *urls*, where the *urls* with more edges in common (*url* - query) can represent closer topics or interests. However, the amount of data needed to create a coherent graph should be studied.

Acknowledgements

This work was partly supported by the European Commission under FP7 project Inter-Trust, by the Spanish Ministry of Science and Innovation (through projects eAEGIS TSI2007-65406-C03-01, CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, Audit Transparency Voting Process IPT-430000-2010-31 and BallotNext IPT-2012-0603-430000) and by the Government of Catalonia (under grant 2009 SGR 1135).

References

1. M. Richardson. Learning about the world through long-term query logs. *ACM Trans. Web*, pages 1–27, 2008.
2. L. Xiong and E. Agichtein. Towards Privacy-Preserving Query Log Publishing. In Einat Amitay, Craig G. Murray, and Jaime Teevan, editors, *Query Log Analysis: Social And Technological Challenges. A workshop at the 16th International World Wide Web Conference (WWW 2007)*, 2007.
3. Y. He and J. Naughton. Anonymization of set-valued data via top-down, local generalization. *Proceedings of the VLDB Endowment*, 2(1):934–945, 2009.
4. E. Adar. User 4XXXXX9: Anonymizing Query Logs. In *Query Log Analysis: Social And Technological Challenges. A workshop at the 16th International World Wide Web Conference (WWW 2007)*, 2007.

5. Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy, 1997.
6. J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W. Fu. Utility-based anonymization for privacy preservation with less information loss. *SIGKDD Explor. Newsl.*, 8(2):21–30, December 2006.
7. D. Defays and P. Nanopoulos. Panels of enterprises and confidentiality: the small aggregates method. In *Proc. of 92 Symposium on Design and Analysis of Longitudinal Surveys, Statistics Canada*, pages 195–204, 1993.
8. Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri. Effective anonymization of query logs. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1465–1468, 2009.
9. G. Navarro-Arribas, V. Torra, A. Erola, and J. Castellà-Roca. User k-anonymity for privacy preserving data mining of query logs. *Information Processing & Management*, 48(3):476 – 487, 2012.
10. A. Erola, J. Castellà-Roca, G. Navarro-Arribas, and Vicenç Torra. Semantic microaggregation for the anonymization of query logs. In *Proc. Privacy in Statistical Databases (PSD 2010)*, volume 6344 of *LNCS*, pages 127–137, June 2010.
11. A. Erola, J. Castellà-Roca, G. Navarro-Arribas, and V. Torra. Semantic microaggregation for the anonymization of query logs using the open directory project. *SORT-Statistics and Operations Research Transactions*, 35(Special issue):25–40, Sep 2011.
12. P. Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.
13. J. Domingo-Ferrer and J.M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14:189–201, 2002.
14. J. Domingo-Ferrer, F. Sebé, and A. Solanas. A polynomial-time approximation to optimal multivariate microaggregation. *Comput. Math. Appl.*, 55(4):714–732, February 2008.
15. A. Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. *ACM Transactions on the Web*, 2(4), 2008.
16. A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 171–180, 2009.
17. B. Poblete, M. Spiliopoulou, and R. Baeza-Yates. Website privacy preservation for query log publishing. In *First International Workshop on Privacy, Security, and Trust in KDD (PinKDD 2007)*, pages 80–96, 2008.
18. G. Miller. WordNet - about us. WordNet. Princeton University, 2009.
19. ODP. Open directory project, 2011.
20. R. Sætre, A. Tveit, T.S. Steigedal, and A. Lægreid. Semantic annotation of biomedical literature using google. In *ICCSA (3)*, pages 327–337, 2005.
21. R. Gligorov, Z. Aleksovski, W. Kate, and F. Van Harmelen B. Using google distance to weight approximate ontology matches. In *In: Proc. WWW-07. (2007*, pages 767–776. ACM Press, 2007.
22. iprospect.com, inc,. iProspect Blended Search Results Study, 2009. `http://www.iProspect.com`.