SORT Special issue: Privacy in statistical databases, 2011, 41-58

Semantic microaggregation for the anonymization of query logs using the open directory project

Arnau Erola[1], Jordi Castellà-Roca[1], Guillermo Navarro-Arribas[2] and Vicenç Torra[3]

Abstract

Web search engines gather information from the queries performed by the user in the form of query logs. These logs are extremely useful for research, marketing, or profiling, but at the same time they are a great threat to the user's privacy. We provide a novel approach to anonymize query logs so they ensure user *k*-anonymity, by extending a common method used in statistical disclosure control: microaggregation. Furthermore, our microaggregation approach takes into account the semantics of the queries by relying on the Open Directory Project. We have tested our proposal with real data from AOL query logs.

MSC: 68T35

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

1. Introduction

Web Search Engines play a decisive role in the Internet nowadays. For instance, there is an estimate of over 113 billion searches conducted globally on the Internet during July 2009, which is up by 41% percent compared to July 2008 (SearchEngineWatch, 2009). These numbers give some insight on the relevance and growth rate use of Web search engines (WSE). Major WSE such as Google, Yahoo!, Baidu, or Microsoft's Bing serve most of the searches in the global Internet with respective shares of 67.5%, 7.8%, 7.0%,

[1] 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

[2] Department of Information and Communications Engineering, Universitat Autónoma de Barcelona, 08193 Bellaterra (Catalonia, Spain). gnavarro@deic.uab.cat

[3] IIIA, Institut d'Investigació en Intel·ligència Artificial, CSIC, Consejo Superior de Investigaciones Científicas, Campus UAB s/n, 08193 Bellaterra (Catalonia, Spain). vtorra@iiia.csic.es

Received: November 2010
Accepted: March 2011

and 2.9% in 2008. This share is more proportional if we look for example at US figures, where in September 2010 the share of searches was 65.4% (Google), 17.4% (Yahoo), and 11.1% (Microsoft) (SearchEngineWatch, 2010). Web search is not only important in the global Internet, as most sites, corporate intranets, or community portals provide local WSEs.

The information gathered by a WSE is stored and can be used to provide personalized search results (Gauch and Speretta, 2004), to conduct marketing research (Hansell, 2006), or provide personalized advertisement. These data, normally referred to as *search* or *query logs*, are a great economic source for the WSE, for instance, Google had a revenue of 21 128.5 million dollars in 2008 from advertisements (Google, 2008), which is strongly based in the information gathered by their search engine. WSEs also charge law enforcement agencies for access to user or group profiles (Summers, 2009; Zetter, 2009).

The detailed information that can be obtained from query logs, make these data an important threat to the privacy of the users. For instance, in 2006, AOL Research, in an attempt to help the information retrieval research community, released over 21 million queries from over 650, 000 subscribers over a 3 month period. Although the data were previously anonymized, they still carried enough information to be an important threat to the subscribers' privacy. Journalists from the New York Times were able to locate an individual (Barbaro and Zeller, 2006) from the query logs, and several other sensitive information was exposed. The case ended up not only with an important damage to AOL users' privacy, but also with a major damage to AOL itself, with several class action suits and complaints against the company (EFF, 2009; Mills, 2006).

In this paper, we address the privacy problem exposed by the WSE query logs when they are made publicly available, transferred to third parties, or stored for future analysis. The main objective is to preserve the utility of the data without risking the privacy of their users. To that end, we follow the same ideas found in statistical disclosure control (SDC), proposing a novel microaggregation method to anonymize query logs. This approach ensures a high degree of privacy, providing k-anonymity at user level, while preserving some of the data usefulness. Moreover, and unlike most of the previous work, our approach takes into account the semantics of the queries made by the user in the anonymization process making use of information obtained from the Open Directory Project (2010).

The paper is organized as follows. Section 2 introduces microaggregation and our motivation and approach for the semantic anonymization of query logs. In Section 3 we detail our proposal, and Section 4 presents our results in terms of protection and utility. Section 5 discusses the related work, and finally, Section 6 concludes the paper.

1.1. Privacy Problems

The privacy problem of query logs is given by the fact that they can contain personal information (Soghoian, 2007). For instance, a user may have searched for her city, a local team, a disease suffered by herself, adult content, or she can make a vanity query, for which the user searches for her own name (Kumar *et al.*, 2007; Soghoian, 2007). This information, either by itself or with help of more information can allow to re-identify the user (Frankowski *et al.*, 2006). So the main threat exposed by a query log is to be able to link user queries with a real identity. The anonymization process can remove a lot of information to provide a high level of privacy to the user, but the resulting log might not be very useful. On the other hand, a more useful log can be obtained if less information is removed. So, there is a privacy-utility tradeoff (Adar, 2007). Query logs should be properly protected with an anonymization process and data should remain useful.

Accordingly, any release of query logs must ensure two requirements:

- **Anonymity**: queries alone or with external information cannot be used to re-identify any user.
- **Usefulness**: queries must contain enough true information to bear likeness to the reality and to be minimally useful. If the information is very damaged, it loses its reliability and value.

To make the personal information retrieval difficult, queries are usually combined with other ones that obfuscate them. Microaggregation (Defays and Nanopoulos, 1993) is a popular statistical disclosure control technique, which provides privacy by means of clustering the data into small clusters and then replacing the original data by the centroids of the corresponding clusters.

Microaggregation provides privacy comparable with k-anonymity (Samarati, 2001; Sweeney, 2002), i.e., a query of a certain user cannot be distinguished from at least $k-1$ queries generated by other users. So, the identification of a user must be imprecise. In terms of usefulness, the larger k is, the less achieved usability because the microaggregated log keeps less information of each user (see Section 4.1).

2. Towards a semantic microaggregation for query logs

In this paper, we propose a novel microaggregation method for query logs taking into account the semantics of the queries made by the users. In this section, we overview microaggregation and discuss the motivations of our proposal.

2.1. Microaggregation

In microaggregation, privacy is ensured because all clusters have at least a predefined number of elements, and therefore, there are at least k records with the same value. Note that all the records in the cluster replace a value by the value in the centroid of the cluster. The constant k is a parameter of the method that controls the level of privacy. The larger the k, the more privacy we have in the protected data.

Microaggregation was originally defined for numerical attributes (Defays and Nanopoulos, 1993), but later extended to other domains, for example, to categorical data in Torra (2004) (see also Domingo-Ferrer and Torra, 2005), and in constrained domains in Torra (2008).

From the operational point of view, microaggregation is defined in terms of partition and aggregation:

- **Partition.** Records are partitioned into several clusters, each of them consisting of at least k records.
- **Aggregation.** For each of the clusters, a representative (the centroid) is computed, and then original records are replaced by the representative of the cluster to which they belong.

From a formal point of view, microaggregation can be defined as an optimization problem with some constraints. We give a formalization below using u_{ij} to describe the partition of the records in the sensitive data set X. That is, $u_{ij} = 1$ if record j is assigned to the ith cluster. Let v_i be the representative of the ith cluster, then a general formulation of microaggregation with g clusters and a given k is as follows:

$$\begin{array}{ll} \text{Minimize} & SSE = \sum_{i=1}^{g} \sum_{j=1}^{n} u_{ij} (d(x_j, v_i))^2 \\ \text{Subject to} & \sum_{i=1}^{g} u_{ij} = 1 \text{ for all } j = 1, \ldots, n \\ & 2k \geq \sum_{j=1}^{n} u_{ij} \geq k \text{ for all } i = 1, \ldots, g \\ & u_{ij} \in \{0, 1\} \end{array}$$

For numerical data, it is usual to require that $d(x, v)$ is the Euclidean distance. In the general case, when attributes $\mathbf{V} = (V_1, \ldots, V_s)$ are considered, x and v are vectors, and d becomes $d^2(x, v) = \sum_{V_i \in \mathbf{V}} (x_i - v_i)^2$. In addition, it is also common to require for numerical data that v_i is defined as the arithmetic mean of the records in the cluster, that is, $v_i = \sum_{j=1}^{n} u_{ij} x_i / \sum_{j=1}^{n} u_{ij}$. As the solution of this problem is NP-Hard (Oganian and Domingo-Ferrer, 2001) when we consider more than one variable at a time (multivariate microaggregation), heuristic methods have been developed. One such method is MDAV (*Maximum Distance to Average Vector*) (Domingo-Ferrer and Mateo-Sanz, 2002).

Note that when all variables are considered at once, microaggregation is a way to implement k-anonymity (Samarati, 2001; Sweeney, 2002).

2.2. Motivations of our proposal

In order to ensure the privacy of the users, we provide k-anonymity at user level in the protected query logs. That is, in the protected logs there will be at least k indistinguishable users.

```
Open Directory Categories  (1-5 of 5)
   1. Sports: Soccer: UEFA: Spain: Clubs: Barcelona   (11 matches)
   2. World: Polski: Sport: Sporty pilki i siatki: Pilka nozna: Kluby: Hiszpan'skie: (...)
   3. World: Español: Regional: Europa: España: Deportes y tiempo libre: Deportes: (...)
   4. World: Deutsch: Sport: Ballsport: Fuball: Vereine: Spanien   (3)
   5. World: Français: Sports: Balles et ballons: Football: Regional: Europe: Espagne   (3)
```

Figure 1: *Example of ODP query result.*

A key point, thus, for the microaggregation of search logs is to determine how the users are clustered. If the users in the same cluster do not share any interest, the protected query logs can be useless, that is, the resulting search logs are too much distorted and we cannot obtain useful information from them.

For example, we can consider two soccer supporters, and two anti-sports users. If we create a cluster of size two with a soccer supporter and an anti-sports user, we can obtain non-valid results. The entries of the protected query logs are confusing. On the other hand, if the two soccer supporters are in the same cluster, the protected logs provide more reliable results.

Thus, we should create the groups of users taking into consideration their interests. The users with common interests between them should be grouped in the same cluster. In order to do so, we should be able to determine if their interests are closer, that is, we need a tool to compute the semantic distance of two queries.

In this work, we use the Open Directory Project (ODP) (ODP, 2010) to compute the semantic distances between users. The ODP is the most widely distributed database of Web content classified by humans. ODP data powers the core directory services for some of the most popular portals and search engines on the Web, including AOL Search, Netscape Search, Google, Lycos, and HotBot, and hundreds of others. Thus, a query result using them is hardly influenced by the ODP classification. ODP uses a hierarchical ontology structure to classify sites according to their themes. For example, when we search for *Barcelona FC*, ODP returns a list of categories to which the query belongs (Figure 1). Each result starts with a root category followed by deeper categories in the ODP tree.

Our proposal groups users with common interests using the ODP classification. We consider that the users with common interest are those who have more terms in the same categories.

2.3. An ODP similarity measure

In order to be able to microaggregate users from the query logs, we have to define a distance or similarity measure between users. We introduce a similarity coefficient based on the common categories shared between queries from each user. We also introduce some notation here to formalize the process.

We consider the set of n users $U = \{u_1, \ldots, u_n\}$ from the query log, and their respective set of queries $Q = \{Q_1, \ldots, Q_n\}$, where $Q_i = \{q^i_1, \ldots, q^i_{m_i}\}$ are the queries of the user u_i. Each query q^i_j has several terms $q^i_j = \{t_1, \ldots, t_{r_j}\}$.

Given a term t_s, we can obtain its classification in the ODP at a given depth. When querying the ODP, the returned categories can be divided in depth levels. Let l be the parameter that identifies the depth level in the ODP hierarchy. For example, if we have the classification $Sports : Soccer : UEFA : Spain : Clubs : Barcelona$ and $l = 1$, we only work with the root category $Sports$; when $l = 2$ we work with $Sports : Soccer$; and so on. We will consider a maximum depth L to restrict the search space, so $l \in \{1,\dots,L\}$.

We denote as $C_l = \{c^l_1,\dots,c^l_{p_l}\}$ the set of possible categories at level l in the ODP. Given a user u_i we can obtain all the categories at level l from all queries of the user. We denote as $C_l(u_i)$ the set of categories for user u_i at level l. Note that considering all queries of user u_i, $Q_i = \{q^i_1,\dots,q^i_{m_i}\}$, and their respective sets of terms $q^i_j = \{t_1,\dots,t_{r_j}\}$ for $j = 1\dots m_i$, the number of categories for user u_i at level l is given by $|C_l(u_i)| = r_1 + \dots + r_{m_i}$.

We can then define a similarity coefficient ODP_{sim} between two given users u_i and u_j as:

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

This similarity coefficient between two users computes the common categories between them for all considered levels, that is levels up to L. Note that OPD_{sim} is symmetric and ranges from 0 (there is no similarity between the users) to $\sum_{l=1}^{L} |C_l|$ (maximum similarity between two users).

3. ODP-based microaggregation of query logs

The method we propose to protect the query logs is a microaggregation that follows the outline of Section 2 with an extra step of data preparation. That is, our approach consists of the following steps:

1. Data preparation.
2. Partition.
3. Aggregation.

These steps are described in detail in the following sections.

3.1. Data preparation

To easy the computation of the protected data, the data is prepared by pre-querying the ODP to classify the user queries. Following the notation introduced in Section 2.3, for every term t_s, we can obtain its classification for all levels $l \in \{1,\dots,L\}$ using the ODP.

This allows us to obtain all the categories associated to all the users in all levels, that is $C_l(u_i)$ for all user $u_i \in U$, and all considered levels. Next, we create a *classification matrix* that contains the number of queries for each user and category at level l, $M_{U \times C_l}$. Please, note that, we obtain one matrix for every level $l \in \{1,\ldots,L\}$. So, $M_{U \times C_l}(i,j)$ is the number of times that category c_j^l is found in the queries of user u_i.

Finally, we use the $M_{U \times C_l}$ matrices in order to compute the *incidence matrix* that contains the semantic similiarity of the users $M_{U \times U}$. Given the incidence matrix $M_{U \times U}$, $M_{U \times U}(i,j)$ is the number of common categories between users u_i, and u_j for all depth levels $l \in \{1,\ldots,L\}$. Moreover note that the incidence matrix corresponds to the similarity coefficient described in Section 2.3, that is, $M_{U \times U}(i,j) = ODP_{sim}(u_i,u_j)$.

The process works as follows:

1. Obtain the classification matrices $M_{U \times C_l}$ using Algorithm 1.
2. Obtain the incidence matrix $M_{U \times U}$ using Algorithm 2, i.e. the similarity coefficient between users.

Algorithm 1 Algorithm for computing the classification matrices $M_{U \times C}^L$ where $L = \{1,\ldots,l\}$

```
Require: the maximum depth $L$ for the ODP categories
Require: the set of users $U = \{u_i,\ldots,u_n\}$
Require: the set of queries $Q_i = \{q_1^i,\ldots,q_{m_i}^i\}$ of each user $u_i$
Require: the set of terms $\{t_1,\ldots,t_{r_j}\}$ of each query $q_j$
Ensure: $\{M_{U \times C_1},\ldots,M_{U \times C_L}\}$, i.e. for every level $l$, the matrix $M_{U \times C_l}$ with the number of queries for each
  category and user in the depth $l$
  for $l \in \{1,\ldots,L\}$ do
    for $u_i \in \{u_1,\ldots,u_n\}$ do
      for $q_j^i \in Q_i = \{q_1^i,\ldots,q_{m_i}^i\}$ do
        for $t_s \in q_j^i = \{t_1,\ldots,t_{r_j}\}$ do
          obtain the categories $c_t$ at depth $l$ for the term $t_s$ using ODP;
          for each $c_t$ do
            if $c_t \in M_{U \times C_l}$ then
              $M_{U \times C_l}(u_i,c_t) = M_{U \times C_l}(u_i,c_t) + 1$;
            else
              add the column $c_t$ to $M_{U \times C_l}$;
              $M_{U \times C_l}(u_i,c_t) = 1$;
            end if
          end for
        end for
      end for
    end for
  end for
  return $\{M_{U \times C_1},\ldots,M_{U \times C_L}\}$.
```

Algorithm 2 Algorithm for computing the incidence matrix $M_{U\times U}$

```
Require: the classification matrices $\{M_{U\times C_1},\dots,M_{U\times C_L}\}$
Ensure: $M_{U\times U}$
  Initialize $M_{U\times U}(i,j) \leftarrow 0$ for all $i,j = 1\dots n$;
  for $M_{U\times C_l} \in \{M_{U\times C_1},\dots,M_{U\times C_L}\}$ do
    for each column $c_j \in M_{U\times C_l}$ do
      for each row $u_i \in M_{U\times C_l}$ do
        for each row $u_\rho \in M_{U\times C_l}$ do
          $M_{U\times U}(u_i,u_\rho) \leftarrow M_{U\times U}(u_i,u_\rho) + \min(M_{U\times C_l}(u_i,c_j), M_{U\times C_l}(u_\rho,c_j))$;
        end for
      end for
    end for
  end for
  return $M_{U\times U}$.
```

3.2. Partition

The partition step creates groups of k users with similar interests using Algorithm 3.

Let us assume that u_i and u_ρ are the most similar users in the set. We calculate the users' similarity ODP_{sim} using the incidence matrix $M_{U\times U}$, (see Section 3.1). The most similar users are those that have the highest similarity coefficient in the matrix. Next, we include u_i and u_ρ to the cluster. If the group size k is two, we delete u_i and u_ρ records from the incidence matrix and we repeat the process to obtain a new cluster. When the group size is bigger than two, we merge the columns and rows of u_i and u_ρ creating a new user u'. u' is the addition of both users, u_i and u_ρ. Let us assume, that u_ξ is the most similar user with u'. Next, we include u_ξ to the cluster with u_i and u_ρ. The method executes this process $k-2$ times.

Algorithm 3 Algorithm for computing the clusters $Z = \{z_1,\dots,z_\gamma\}$ of users

```
Require: the set of users $U = \{u_1,\dots,u_n\}$
Require: the incidence matrix $M_{U\times U}$
Require: the clusters size $k$
Ensure: the clusters $Z = \{z_1,\dots,z_\gamma\}$ of users for $\gamma = \lceil n/k \rceil$
  $M'_{U\times U} \leftarrow M_{U\times U}$;
  $U' \leftarrow U$;
  while $|U'| \leq k$ do
    obtain the cluster $z$ of $k$ users using the Algorithm 4 and $M'_{U\times U}$;
    remove the users $u_i \in z$ form $U'$;
    remove the columns and the rows of the users $u_i \in z$ form $M'_{U\times U}$;
    add $z$ to the set $Z$;
  end while
  return $Z = \{z_1,\dots,z_\gamma\}$.
```

Algorithm 4 Algorithm for computing a cluster z of k users

```
Require: a incidence matrix $M'_{U\times U}$
Require: the clusters size $k$
Ensure: a cluster $z$ of $k$ users
  $z \leftarrow \emptyset$;
  obtain the two most similar users $(u_i, u_\rho)$, i.e. the cell of $M'_{U\times U}$ with the highest value;
  add $(u_i, u_\rho)$ to the set $z$;
  while ($|z| < k$) and ($columns(M'_{U\times U}) > 0$) do
    for each column $c_s \in M'_{U\times U}$ do
      $M'_{U\times U}(c_s, u_\rho) = M'_{U\times U}(c_s, u_\rho) + M'_{U\times U}(c_s, u_i)$;
    end for
    for each row $r_s \in M'_{U\times U}$ do
      $M'_{U\times U}(u_i, r_s) = M'_{U\times U}(u_i, r_s) + M'_{U\times U}(u_\rho, r_s)$;
    end for
    delete the column $u_\rho$ of matrix $M'_{U\times U}$;
    delete the row $u_\rho$ of matrix $M'_{U\times U}$;
    obtain the new $u_i$'s most similar user $u_\rho$, i.e. the cell of the user $u_i$ with the highest value;
    add $u_\rho$ to the set $z$;
  end while
  return $z$.
```

3.3. Aggregation

For every cluster z_j formed in the partition step, we compute its aggregation by selecting specific queries from each user in the group. That is, given the cluster of users $z_j = \{u_1, \ldots, u_k\}$, we obtain a new user u_{z_j} as the representative (or centroid) of the cluster, which summarizes the queries of all the users of the cluster. The selection of queries is based on the following principles:

1. We give priority to queries semantically close between them.
2. The number of queries a user contributes to the cluster representative is proportional to the number of queries of the user.

The first principle is considered in the partition step described in Section 3.2, since clusters are composed of users with semantically similar queries. The second principle is formalized defining some indexes as described below.

First, the number of queries of the centroid is the average of the number of queries of each user u_i of the cluster z_j. Then, the contribution of a user u_i ($Contrib_i$) to the centroid of a cluster with k users, depends on her number of queries $|Qi|$. This contribution is as follows:

$$Contrib_i = \frac{|Qi|}{\sum_{i=1}^{k} |Qi|} \tag{2}$$

Algorithm 5 Algorithm to aggregate the k users of the cluster z

```
Require: a cluster $z$ of $k$ users
Require: the quota $Quota_i$ of each user of the cluster $z$
Require: the contribution $Contrib_i$ of each user of the cluster $z$
Require: the set of queries $Q_i$ of each user of the cluster $z$
Require: the queries list $SL$
Require: the microagregged log $ML$
Ensure: the centroid of the cluster $z$
  $ML \leftarrow \emptyset$
  for each user $u_i \in z$ do
    $SL \leftarrow$ sort $Q_i = \{q_1^i, \ldots, q_{m_i}^i\}$ by query repetitions.
    while not reach $Quota_i$ do
      Add the first query $q_1^i$ with a probability $Contrib_i \times \#q_1^i\_repetitions$ to $ML$.
      Delete $q_1^i$ of $SL$.
    end while
  end for
  return $ML$.
```

Thus, the quota of each user u_i in the new centroid u_{z_j} can be computed as:

$$Quota_i = \frac{|Qi|}{k} \tag{3}$$

More formally, the aggregation method runs the Algorithm 5 for each cluster. First, it sorts logs from all users descending by query repetitions. Then, for each user u_i of the cluster and while not reaching $Quota_i$ do:

1. Add the first query of her sorted list with a probability $Contrib_i \times \#q_j\_repetitions$. For example, if u_i has a query repeated 3 times, and $Contrib_i$ is 0.4, as $3 \cdot 0.4 = 1.2$, the method adds one query to the new log and then randomly chooses to add it again or not according to the presence probability 0.2.
2. Delete the first query of the list.

4. Evaluation

We have tested our microaggregation method using real data from the AOL logs released in 2006, which correspond to the queries performed by 650 000 users over three months. We randomly select 1 000 users, which correspond to 55 666 lines of query logs. The usefulness evaluation and the results are presented below.

4.1. Usefulness evaluation method

For each user we have her original set of queries and the corresponding protected ones by means of our microaggregation method. All queries can be classified in categories, that is, each query is classified in the L first depth levels of the ODP.

In order to verify that our method preserves the usefulness of the data (i.e., does not introduce too much perturbation), we count the number of queries of each category, for a given level l, that are in the original log as well as in the centroid, ρ. This number is divided by the number of original queries in l, χ, obtaining a *semantic remain percentage* (*SRP*) in the level.

$$SRP = \frac{\rho}{\chi} \tag{4}$$

To summarize, our evaluation method does not only match two equal terms in both logs, but also a term in the protected log that replaces one with closest semantic in the original log. Using a random partition algorithm, users of each cluster might not be semantically close.

Consider, as an example of the worst case, a cluster of k users $\{u_1,\ldots,u_k\}$ with respective queries $Q = \{Q_1,\ldots,Q_k\}$, such that $Q_i \cap Q_j = \emptyset$ for all $i \neq j$. Thus, only the queries of a single user in a specific topic will appear in the centroid.

In this case, the number of queries of u_i that appear in the centroid can be calculated using formula 3 and it is known that the sum of all quotas is χ. Therefore, in the worst case when no common interests between users exists, we can calculate the average *SRP* as:

$$\frac{\sum_{i=1}^{k} \frac{|Q_i|}{\chi}}{k} = \frac{1}{k} \tag{5}$$

4.2. Results

As discussed in Section 2.2, ODP returns a list of categories for every term (or query), and each category is composed of various hierarchical levels. In our method, one or all categories can be used and, for each category, either all hierarchical levels or some of them can be considered. Intuitively, the more categories and levels (deeper levels) that are used, the higher the computational cost should be, and, perhaps, a better SRP can be achieved. Thus, we want to study how these parameters influence the SRP and the computational cost:

- ODP levels: every term has a categorization up to a hierarchical level, and the deepest level can be different for every term. The deeper the level is, the less terms that have information in this level there are. We want to know the deepest level that gives information for a majority of terms.

- SRP vs. ODP-categories: we want to know the SRP value when we use more or less categories; that is, if we use more categories, the SRP can be either higher, or have approximately the same SRP.
- Computational cost vs. ODP-categories: supposing that more categories are used, the higher the computational cost will be, but the extra cost should be known. If the extra cost is not significant and a better SRP is obtained, more categories can be used.

4.2.1. ODP levels

In the ODP, not all terms rank up to a certain level. For example, our working set of queries has terms with two levels (minimum) and others with twelve levels (maximum). In the study of the above mentioned relations (SRP vs. ODP-categories and computational-cost vs. ODP-levels), levels that do not have a ranking for the majority of terms can be ignored because such levels only give information to improve the SRP for a reduced number of terms. Thus, we consider a level if it has information for, at least, the 50% of the terms (queries).

In this sense, we have calculated for every level the percentage of terms that have a result for the level, and Figure 2 shows the percentage of queries (our working set of queries) that can be classified up to a certain depth level in the ODP tree. It can be observed that only 57% of queries can be classified up to the level 5. So, we only run tests up to this level.

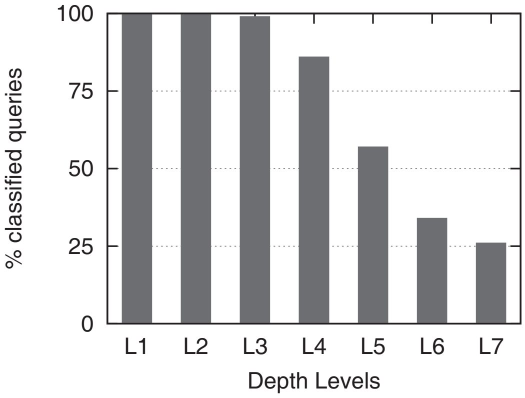

Figure 2: *Percentage of queries that can be classified up to a certain level in ODP.*

4.2.2. SRP vs. ODP-categories

Besides some initial tests (Erola *et al.*, 2010), we have calculated the percentage of semantically similar queries as the accumulation of the levels; that is, we add the coincidences of level 1 and 2 to calculate the percentage of semantically similar queries at level 2. In this current work, we have changed the evaluation method because we think that

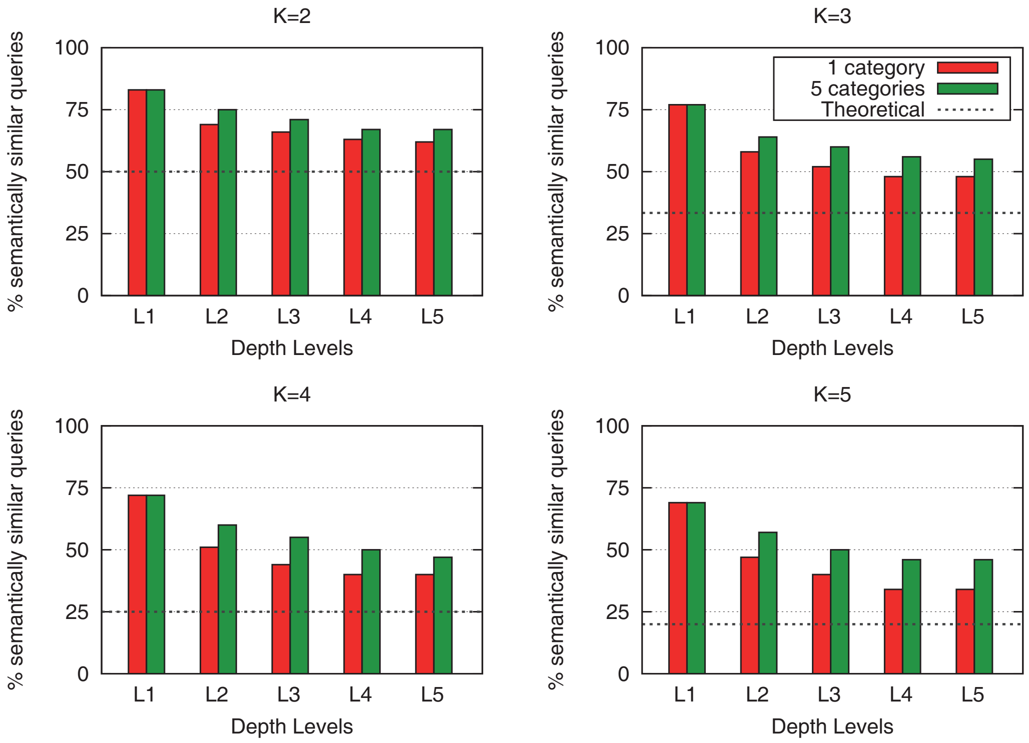

Figure 3: *Semantic similarity percentage of microaggregated logs using either the first category or the five first categories returned by the ODP.*

to evaluate each level separately is better to understand the remaining similarity of the queries in that level.

We have compared the results obtained (SRP) by either using the first five categories returned by the ODP or using only the first one. The range is enough in order to evaluate the SRP behaviour when we use more categories. Note that the first category that gives ODP is the most significative for the introduced term. Figure 3 shows, for cluster sizes 2, 3, 4 and 5, the average *SRP* that users obtain for various levels L. The red colour represents the obtained results using the first category returned by the ODP and the green colour represents the obtained results using the first five categories. It can be observed that both tests improve the theoretical *SRP* (see Section 4) with all depth levels. Using more categories in the ODP classification we achieve less similarity loss for deeper levels and larger cluster sizes. For instance, when $L = 1$, the same gain is obtained in all cases, but when $L = 5$ and $k = 5$, the difference gain is approximately 10% using the first five categories instead of only the first one.

4.2.3. Computational cost vs. ODP-categories

The computation cost is larger when more categories are used. Figure 4 shows the average time required to microaggregate logs for cluster sizes $k = 2, \ldots, 5$ for various levels. It can be determined that using the first five ODP categories, the average time is three times larger than using only the first one.

Tests were run on a Pentium Core 2 Duo 2.2Ghz without source code parallelization. Figure 4 demonstrates that the required time increases linearly with the number of user queries. Nonetheless, the program could be parallelized as follows:

- **Data preparation:** as each user has her queries, the classification matrices $M_{U\times C}$ can be computed simultaneously. Then, each cell of the incidence matrix $M_{U\times U}$ can be calculated independently, since we have available the classification matrix of each user.
- **Partition:** the partition process is linear and cannot be parallelized, but it is a negligible part of the whole process. The time required for its calculation is less than one percent of the total time.
- **Aggregation:** as users are divided into k groups, the logs' aggregation of each group can be run simultaneously.

Thus, the program parallelization could make the proposal scalable for very large systems.

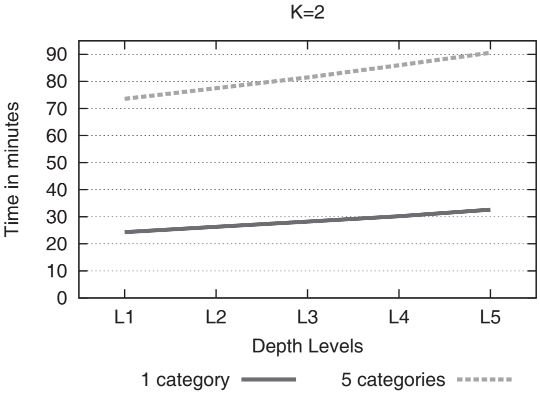

Figure 4: *Average required time to microaggregate logs using our method for various ODP levels.*

4.2.4. Considerations

It should be taken into consideration that we have repeated the tests of the previous initial work (Erola *et al.*, 2010) and we have observed that the results have improved because the ODP is constantly getting better. It now classifies more words. Furthermore, notice that we are working with a set of 1 000 users, randomly selected from the AOL files. We expect to achieve greater *SRP* values working with a larger set, because more similar users may be grouped.

It is important to remark that our proposal achieves k-anonymity (Samarati, 2001; Sweeney, 2002) at user level, which guarantees that at least k users are indistinguishable in the protected version. This guarantees a high degree of privacy, preventing the famous privacy leaks of the AOL logs.

To some readers our proposal might resemble agglomerative hierarchical clustering methods such as the well known Ward method (Ward, 1963). This method has been also adapted to perform microaggregation, although in another context, in Domingo-Ferrer and Mateo-Sanz (2002).

5. Related work

There are several approaches to anonymize query logs in the literature (Cooper, 2008), but they are normally reduced to the deletion of specific queries or logs. For instance, in (Adar, 2007) the authors propose a technique to remove infrequent queries, while in Poblete *et al.* (2008) a more sophisticated technique is introduced to remove selected queries to preserve an acceptable degree of privacy, or in the case of Korolova *et al.* (2009) to choose the publishable queries. Common techniques used in statistical disclosure control (SDC) have not been applied to this specific problem until very recently (Navarro-Arribas and Torra, 2009; Hong *et al.*, 2009; Navarro-Arribas *et al.*, in press, 2011). Moreover, these systems use spelling similarities to link users; that is, two users would be grouped if they had submitted syntactic similar queries. Therefore, they cannot distinguish different senses of a term, if it has more than one.

The use of supporting semantic taxonomies to anonymize query logs was considered in He and Naughton (2009) where the authors anonymize the set of queries made by a user by generalizing the queries using WordNet (Miller, 2009). WordNet is a generic lexical database of the English language, where concepts are interlinked by means of conceptual-semantic and lexical relations. The problem of relying on WordNet when facing the anonymization of query logs is that the query introduced by the user, despite the fact that they might not be in English, can be meaningless in a generic dictionary. We think that better results can be obtained for query logs by gathering semantic information from the Open Directory Project (ODP), which its main purpose is precisely to serve as a catalogue of the Web by providing a content-based categorization or classification of Web pages. This will be the case in general for data which is composed of uncommon words, which could not be found in WordNet. Note that if all words in the query logs were present in WordNet, the use of the WordNet framework will presumably give good results as well. Nevertheless, we need to introduce novel approaches to make the information obtained from the ODP useful. Unlike WordNet, which already has lots of published and tested distances functions, or aggregation operations, ODP lacks this extensive previous work.

6. Conclusions

The existing microaggregation techniques for query logs do not usually take into account the semantic proximity between users, which is negatively reflected in the usefulness of the resulting data. This paper presents a new microaggregation method for query logs based on a semantic clustering algorithm. We use ODP to classify the queries of all users and then aggregate the most semantically close logs. As we have seen, the resulting logs achieves higher usefulness while preserving *k*-anonymity.

We have tested our proposal with real query logs from AOL, showing some good results. Both in terms of information loss and in terms of protection, which is guaranteed because our method ensures *k*-anonymity at user level. As future work, new evaluation methods such as as Domingo-Ferrer and Solanas (2009), will be tested to better assess the quality of the results obtained using our system.

Acknowledgment

Partial support by the Spanish MICINN (projects eAEGIS TSI2007-65406-C03-02, TSI2007-65406-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, Audit Transparency Voting Process PT-430000-2010-31and N-KHRONOUS TIN2010-15764), the Spanish Ministry of Industry, Commerce and Tourism (projects eVerification TSI-020100-2009-720 and SeCloud TSI-020302-2010-153), and the Government of Catalonia (grant 2009 SGR 1135) is acknowledged. G. Navarro-Arribas enjoyed a Juan de la Cierva grant (JCI-2008-3162) from the Spanish MICINN.

The authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization.

References

Adar, E. (2007). User 4xxxxx9: Anonymizing query logs. In *Query Logs workshop*.

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

Cooper, A. (2008). A survey of query log privacy-enhancing techniques from a policy perspective. *ACM Transactions on the Web*, 2.

Defays, D. and Nanopoulos, P. (1993). Panels of enterprises and confidentiality: the small aggregates method. In *Proceedings of 92 Symposium on Design and Analysis of Longitudinal Surveys, Statistics Canada*, 195–204.

Domingo-Ferrer, J. and Mateo-Sanz, J.M. (2002). Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14, 189–201.

Domingo-Ferrer, J. and Torra, V. (2005). Ordinal, continuous and heterogeneous *k*-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11, 195–212.

Domingo-Ferrer, J. and Solanas, A. (2009). Erratum: Erratum to “a measure of variance for hierarchical nominal attributes”. *Information Sciences*, 179, 3732. Elsevier Science Inc. New York. http://dx.doi.org/10.1016/j.ins.2009.06.019.

EFF. (2009). AOL's massive data leak. Electronic Frontier Foundation. http://w2.eff.org/Privacy/AOL/.

Erola, A., Castellà-Roca, J., Navarro-Arribas, G. and Torra, V. (2010). Semantic microaggregation for the anonymization of query logs. In *Proceedings Privacy in Statistical Databases (PSD 2010)*, 6344 of LNCS, 127–137.

Frankowski, D., Cosley, D., Sen, S., Terveen, L. and Riedl, J. (2006). You are what you say: privacy risks of public mentions. In *Annual ACM Conference on Research and Development in Information Retrieval*, 565–572, Seattle Washington.

Gauch, S. and Speretta, M. (2004). Personalized search based on user search histories. In *Proceedings of International Conference of Knowledge Management-CIKM'04*, 622–628.

Google (2008). 2008 annual report. http://investor.google.com/order.html.

Hansell, S. (2006). Increasingly, Internet's data trail leads to court. The New York Times.

He, Y. and Naughton, J. (2009). Anonymization of set-valued data via top-down, local generalization. *Proceedings of the VLDB Endowment*, 2, 934–945.

Hong, Y., He, X., Vaidya, J., Adam, N. and Atluri, V. (2009). Effective anonymization of query logs. In *CIKM'09: Proceedings of the 18th ACM conference on Information and knowledge management*, 1465–1468.

Korolova, A., Kenthapadi, K., Mishra, N. and Ntoulas, A. (2009). Releasing search queries and clicks privately. In *WWW'09: Proceedings of the 18th international conference on World wide web*, 171–180.

Kumar, R., Novak, J., Pang, B. and Tomkins, A. (2007). On anonymizing query logs via token-based hashing. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, 629–638.

Miller, G. (2009). WordNet-about us. WordNet. Princeton University. http://wordnet.princeton.edu.

Mills, E. (2006). AOL sued over web search data release. CNET News. http://news.cnet.com/8301-10784_3-6119218-7.html.

Navarro-Arribas, G. and Torra, V. (2009). Tree-based microaggregation for the anonymization of search logs. In *WI-IAT'09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 155–158.

Navarro-Arribas, G., Torra, V., Erola, A. and Castellà-Roca, J. (in press, 2011). User *k*-anonymity for privacy preserving data mining of query logs. *Information Processing & Management*. DOI:10.1016/j.ipm.2011.01.004

ODP. (2010). Open directory project. http://www.dmoz.org.

Oganian, A. and Domingo-Ferrer, J. (2001). On the complexity of optimal microaggregation for statistical disclosure control. *Statistical Journal of the United Nations Economic Commision for Europe*, 18, 345–353.

Poblete, B., Spiliopoulou, M. and Baeza-Yates, R. (2008). Website privacy preservation for query log publishing. In *First International Workshop on Privacy, Security, and Trust in KDD (PinKDD 2007)*, 80–96.

Samarati, P. (2001). Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13, 1010–1027.

SearchEngineWatch. (2009). Global search market share, july 2009 vs. july 2008. http://searchenginewatch.com/3634922.

SearchEngineWatch. (2010). Top search providers for september 2010. http://searchenginewatch.com/3641456.

Soghoian, C. (2007). The problem of anonymous vanity searches. *I/S: A Journal of Law and Policy for the Information Society*, 3.

Summers, N. (2009). Walking the cyberbeat. Newsweek. http://www.newsweek.com/id/195621.

Sweeney, L. (2002). *k*-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10.

Torra, V. (2004). Microaggregation for categorical variables: a median based approach. In *Proceedings Privacy in Statistical Databases (PSD 2004)*, 3050 of LNCS, 162–174.

Torra, V. (2008). Constrained microaggregation: adding constraints for data editing. *Transactions on Data Privacy*, 1, 86–104.

Ward, J.H. (1963). Hierarchical Grouping to optimize an objective function. *Journal of the American Statistical Association*, 58, 236–244.

Zetter, K. (2009). Yahoo issues takedown notice for spying price list. Wired.
http://www.wired.com/threatlevel/2009/12/yahoo-spy-prices/#more-11725.