Information Sciences 242 (2013) 49–63

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Utility preserving query log anonymization via semantic microaggregation

Montserrat Batet *, Arnau Erola, David Sánchez, Jordi Castellà-Roca

Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Avda. Països Catalans 26, 43007 Tarragona, Spain

ARTICLE INFO

Article history:
Received 25 July 2012
Received in revised form 5 April 2013
Accepted 12 April 2013
Available online 20 April 2013

Keywords:
Privacy-preservation
Query logs
Data utility
Semantics
Microaggregation

ABSTRACT

Query logs are of great interest for scientists and companies for research, statistical and commercial purposes. However, the availability of query logs for secondary uses raises privacy issues since they allow the identification and/or revelation of sensitive information about individual users. Hence, query anonymization is crucial to avoid identity disclosure. To enable the publication of privacy-preserved – but still useful – query logs, in this paper, we present an anonymization method based on semantic microaggregation. Our proposal aims at minimizing the disclosure risk of anonymized query logs while retaining their semantics as much as possible. First, a method to map queries to their formal semantics extracted from the structured categories of the Open Directory Project is presented. Then, a microaggregation method is adapted to perform a semantically-grounded anonymization of query logs. To do so, appropriate semantic similarity and semantic aggregation functions are proposed. Experiments performed using real AOL query logs show that our proposal better retains the utility of anonymized query logs than other related works, while also minimizing the disclosure risk.

1. Introduction

Query logs gathered and released by Web Search Engines (WSEs) provide a deep insight into human behavior and serve as a foundation to improve the user experience on the Web, bringing many benefits for researchers and companies [15,23,37]. However, query log publication also implies a real privacy threat. Since queries can contain sensitive information related with the users' life, such as places near their town, illnesses, and sexuality, they may expose the user's privacy when these data are univocally associated to her identity.

Although query logs can be submitted to a sweep process that removes identifying information prior to their publication, several authors [1,22,37] have discussed how the combination of the remaining data can disclose enough information to re-identify some users. The well-known AOL case [3] is an example of this, in which published queries made by 658,000 users during 3 months were submitted to a poor sweep process, resulting in the identification of some of the users and in a major damage to AOL itself.

To minimize the disclosure risk of published data, anonymization/masking methods have been proposed [11]. They perform transformations over potentially identifying data reducing the level of specificity and/or making groups of individuals indistinguishable. Since this process incurs in a degree of information loss which directly influences the utility of output data, anonymization methods should balance the trade-off between information loss and disclosure risk [11]. To achieve this goal, within the Statistical Disclosure Control (SDC) community, authors have proposed many techniques to anonymize structured

* Corresponding author. Tel.: +34 977 559657; fax: +34 977 559710.
E-mail address: montserrat.batet@urv.cat (M. Batet).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.04.020

databases, consisting on records with several univalued attributes [11,21,30,32]. In these methods, identifier attributes are removed, and quasi-identifier attributes (i.e. those non-identifiers whose combination may identify an individual, such as "birth date + birth place + job") are anonymized. This is usually done to fulfill the k-anonymity property, that is, making each record indistinguishable from, at least, $k-1$ other ones with respect to their quasi-identifiers [39,48].

Query logs, which are usually referred as *set-valued data* [50], are quite different to relational databases, since they do not constitute well-defined sets of attributes and they have variable length and high dimensionality. Hence, specific anonymization methods should be designed [15,50]. Moreover, query logs are expressed in free text, whereas attributes in relational databases are usually either numerical or categorical [28,51]. This raises new challenges because, on the contrary to numerical data which can be compared and transformed by means of mathematical operators, textual data require from operators that take into account their semantics [30,51]. To do so, knowledge sources like taxonomies, folksonomies and more general ontologies [18] can be exploited to analyze and interpret textual data from a semantic perspective. Ontologies, manually o automatically created [26,40], are widely used to interpret and manage textual data in areas such as document categorization or clustering [4,25], semantic annotation [46], event reasoning [56], privacy preservation [29,44] or medical research [36,52].

In this paper, we present a new method to anonymize user query logs, giving special attention to the preservation of their semantics in order to maximize their utility. To that end, we propose a method that maps queries with their formal semantics, using the Open directory Project (ODP) as knowledge base, and a set of operators that permits to perform semantically-coherent comparisons and aggregations of queries. Then, we adapt the MDAV microaggregation method [13], a well-known method proposed by the SDC community to anonymize relational databases, in such a way that it can be applied to set-valued query logs. The evaluation, performed on a set of query logs corresponding to real users, shows that a proper interpretation of the semantics of queries better preserves the utility of the anonymized results, while maintaining a reasonable level of disclosure risk.

The rest of the paper is organized as follows. Section 2 surveys and reviews related works in query log anonymization. Section 3 introduces the basis of data anonymization via microaggregation. Section 4 presents the new method for query anonymization. Section 5 summarizes and compares evaluation results against related works. The final section contains the conclusions.

2. Related work

In the literature, we can find several approaches dealing with the query log anonymization problem based on query removal or hashing. In [10] a survey of these naïve methods is given. First, some systems simply remove old query sets assuming that user logs will be not large enough to enable identity disclosure [45]. This simple criterion hardly preserves privacy in front of highly identifying queries. A more appropriate approach suggests deleting only infrequent queries [1], assuming that those are more likely to refer to identifying or quasi-identifying information. This is, however, challenging due to the difficulty of setting deletion thresholds and due to the fact that the vast majority of queries occur a small number of times [6], which may result in eliminating a substantial amount of non-identifying queries.

Other removal-based methods focus on removing identifying data (e.g. IP addresses) associated to queries and/or identifying/private information found within queries (e.g. SS numbers, credit cards, addresses, etc.) [10]. However, as discussed above, the combination of remaining non-identifying data (i.e. quasi-identifiers) may end with disclosure identity, like in the case of the infamous AOL incident. In a different approach inspired in secret sharing methods, Adar [1] proposes splitting user queries, assigning them to fictional user ids. This, however, limits the usability of results, invalidating the conclusions extracted by a horizontal analysis of queries (e.g. user profiling methods) [20]. Finally, other systems rely on the application of hashing functions to identifying information (e.g. IP address) and/or to queries themselves [24]. Even though this makes difficult the identity disclosure, it also requires from revealing hashing functions to parties willing to perform query analysis. Moreover, some works have also shown the ineffectiveness of hash-based schemes, such as token-based hashing [24].

More elaborated approaches [23,37] remove only those queries that result in clicking common URLs, assuming that those may be dependent (i.e. quasi-identifiers). Poblete et al. [37] represent query logs as a graph in which nodes are queries, and two nodes are connected by one edge if the intersection of their clicked URLs sets is not empty. In particular, they propose a graph disconnection heuristic focused on the elimination of queries (and the corresponding edges). The complete anonymization process is iterative and encompasses the removal of (i) all vulnerable queries, i.e. all queries that return less than k documents (overrestrictive queries) and queries that contain the target URL or at least the site of the URL as a keyword (well-target queries), (ii) those returning documents from less than Kp sites and (iii) all queries that contribute to a nonzero density of the query graph. The density of the graph is the likelihood of finding an edge among any two nodes. The method stops when the value of density is zero. Korolova et al. [23] propose generating a private query click graph. In a nutshell, for every user of the query log, it keeps only the first d queries posed by the user. This step limits users' activity. Then, for each remaining query q_i which appears n_i times, the method outputs the query q_i and n_i', where n_i' is n_i plus a random variable drawn independently from the Laplace distribution with mean zero. The remaining queries are considered safe to publish. Although the method has several parameters, the authors provide some indications and lemmas to compute them according to the parameter d. Even though both approaches focus on the removal of the most informative queries, the obvious drawback is that their utility is completely lost [10].

More recent approaches rely on SDC methods to anonymize query logs while minimizing the amount of query removal [15,33]. To do so, they group similar users together (according to the similarity of their query logs), and then their queries are replaced by a prototype query log, becoming indistinguishable. In this manner, users and queries are preserved, even though the latter are transformed to minimize the disclosure risk. For example, Navarro-Arribas et al. [33] propose several measures to compute the distance between queries, i.e. distance between two timestamps; distance between two domain names; Edit distance between two strings; Edit distance between terms; and queries length. The averaged query distance is used in the MDAV [13] method to create groups of k users. Once groups are created, the prototype query log is computed, preserving the query frequency from the original dataset. The above system is, however, hampered by the evaluation of query similarities, which is solely based on spelling and syntactical matching.

None of the above-mentioned methods consider the semantics of queries during the anonymization process: they execute random or distribution-based transformations over data. As stated in the introduction, and mentioned by several authors [30,51], these compromise the utility of anonymization results.

Semantics have been scarcely considered in related works. The pioneer work by Terrovitis et al. [50] deals with textual set-valued data proposing generalizations of input values according to ad hoc constructed Value Generalisation Hierarchies (VGH), which iteratively generalize input values up to a common node. Authors generalize groups of input queries to a common conceptual abstraction (e.g. "sailing" and "swimming" → "water sports"), until users who performed those queries become indistinguishable (i.e. k-anonymous). He and Naughton adapted the previous method to the anonymization of query logs [20] but starting from the most abstract generalization and progressively specializing it. Due to the dimensionality and unbounded nature of query logs, which makes the construction of ad hoc VGHs unfeasible, they use WordNet [16] (a general purpose semantic network modeling around 100,000 concepts) to assist the query generalization process. The main problem of this approach, as acknowledge by the authors, is that proper nouns (e.g. celebrities, company names, etc.), which are commonly referred in queries, are omitted in WordNet because it models abstract concepts rather than individuals (e.g. it models the concept "Actor", but it does not cover the individual "Tom Cruise"). Moreover, authors independently match each noun found in a query to concepts in WordNet, omitting their relation, which results in a loss of semantics (e.g. "water sports" was mapped to the concepts "water" and "sport"). In any case, both previous methods are affected by the large information loss resulting from concept generalization. This is especially evident for heterogeneous data (like query logs) in which the need to generalize outliers results in abstract concepts (e.g., the root node of the ontology) and high information loss [31].

Finally, Erola et al. [15] present a basic semantic aggregation method that uses the Open Directory Project (ODP) as knowledge source. ODP is the largest directory service on the Web, providing a set of categories to which a given term belongs.[1] Authors look for each word of each query in ODP and retrieve the categories to which they belong. Next, groups of k users are created according to the number of identical categories retrieved for the set of user queries. The prototype query log for each group to be published is created by randomly selecting a subset of queries from the group. Even though this method compares queries at a conceptual level (i.e. according to their categories), it fails to retain the meaning of the complex queries with several words or noun phrases, since those are processed word by word. Moreover, the prototype generation step does not consider semantics during the selection of queries to publish, hampering their utility.

3. Data anonymization via microaggregation

In order to reduce excessive information loss resulting from query removal, as proposed in some of the above-described methods [15,33], our proposal will be based on *data microaggregation*. Microaggregation is a SDC technique that perturbs input data to generate k-anonymous datasets. To that end, input records (query logs, in this case) are clustered into groups, at least, of size k (*data partition*) and replaced by the cluster centroid (*data anonymization*). In this manner, each record becomes indistinguishable from, at least, $k - 1$ other ones.

To maximize the utility of anonymized data, similar records should be clustered together, so that the information loss resulting from the replacement by their centroid can be minimized. However, it has been proved that finding the optimal data partition is in general NP-hard [35]. Hence, approximation algorithms to optimal microaggregation have been proposed. One of them is MDAV (*Maximum Distance Average Vector*), which will be the base of our query anonymization method [13]. MDAV stands out from other microaggregation methods since it is specifically designed to minimize the information loss [12,28].

The behavior of the method is depicted in Algorithm 1. *Data partition* begins by calculating the centroid of the whole dataset and selecting the most distant record (x_r) to it. Then, a cluster is constructed with the $k - 1$ least distant records to x_r. After that, the most distant record x_s to x_r is selected and a new cluster is constructed. The process is repeated until less than $2k$ records remain ungrouped. Remaining records are grouped together in a last cluster. As a result, all clusters will have k records, except for the last one, which may have from k to $2k - 1$ records (considering that the input dataset may not be divisible by k). Finally, *data anonymization* is performed by replacing each record of each cluster by the centroid of the cluster.

[1] http://www.dmoz.org/docs/en/about.html [Last accessed: July 9th, 2012].

Algorithm 1. MDAV

```
Require: $X$: original data set, $k$: integer
Result: $X'$: anonymized data set

begin
  $X' = X$;
  /*Data Partition*/
  while ($|X| \geqslant 3 * k$) do
    Compute the centroid $c_x$ of all records in $X$;
    Find the most distant record $x_r$ to the centroid $c_x$;
    Form a cluster in $X$' that contains $x_r$ together with its $k - 1$ least distant records;
    Remove these records from $X$;
    Find the most distant record $x_s$ to $x_r$;
    Form a cluster in $X$' that contains $x_s$ together with its $k - 1$ least distant records;
      Remove these records from $X$;
  end while
  if ($|X| \geqslant 2 * k$) then
    Compute the centroid $c_x$ of all records in $X$;
    Find the most distant record $x_r$ to the centroid $c_x$;
    Form a cluster in $X'$ that contains $x_r$ together with its $k - 1$ least distant;
    Remove these records from $X$;
  end if
  Form a cluster in $X$' with the remaining records;
  /*Data anonymization*/
  for each cluster $q$ in $X'$ do
    Compute the centroid $c_q$ of all records in $q$;
    Replace all records of $q$ in $X'$ by their centroid $c_q$;
  end for
end
```

MDAV, like most SDC methods, have been originally designed to deal with numerical data. As discussed in the introduction, numbers can be easily managed by means of arithmetical operators. In this case, the Euclidean distance and the arithmetic average have normally been used to compare and anonymize numerical data [14]. However, the application of MDAV to free textual data such as query logs is not straightforward, since semantically-grounded operators are needed to accurately compare and transform them. Even though some authors have applied the MDAV method to categorical data, most of them are limited to terminological comparisons [14], neglecting data semantics.

Another problem arises from the fact that MDAV has been designed to deal with structured databases with univalued attributes. Set-valued data like query logs are not directly supported due to their variable cardinality.

Both problems will be tackled in the following section, enabling a semantically coherent application of the MDAV method to the anonymization of query logs.

4. Query anonymization method

In this section, we present a query log anonymization method (Fig. 1) that focuses on minimizing information loss. Contrary to some related works [20,33], it carefully considers the semantics of all kind of queries (i.e. from one-word to multiple noun phrases containing proper nouns) and adapts the MDAV microaggregation algorithm so that it can be coherently applied to set-valued datasets like query logs. Special care has been put in the creation of anonymized query logs to also minimize the disclosure risk while retaining their utility. Given a set of records, each one corresponding to the set of queries performed by each user, the basic steps of the method are:

(1) *Query processing and conceptual mapping*: to semantically interpret textual queries, these are processed so that syntactical constructions (i.e. noun phrases) can be mapped to their conceptual abstractions modeled in a knowledge base.
(2) *Semantic data partition*: clusters of query logs of at least k-users are created (to fulfill k-anonymity) by means of the MDAV microaggregation algorithm. The cluster construction process and the centroid calculus method on which the MDAV method relies, have been adapted so that they consider query semantics and the distributional properties of set-valued data.
(3) *Semantic query anonymization*: clustered query logs are replaced by a synthetic set of queries that represents both their meaning and their distribution. As a result, users whose query logs belong to each cluster become, at least, k-anonymous. The synthetic query log is constructed so that the information loss and the disclosure risk associated to that replacement are minimized.

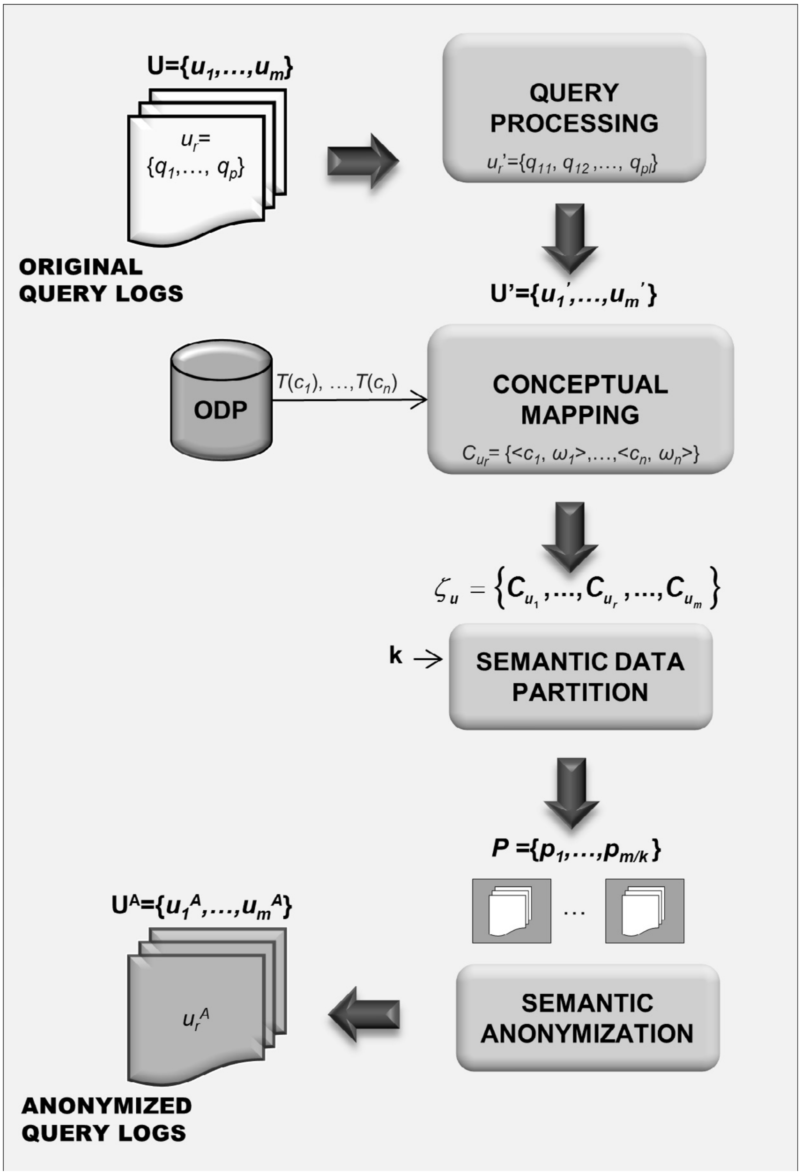

Fig. 1. Semantic query log anonymization method.

4.1. Query processing and conceptual mapping

To semantically interpret users' queries, we need to map them to their formal semantics (i.e. conceptual abstractions), modeled in a background knowledge base. Users' queries, being free text strings, could be problematic to manage. They may directly correspond to an individual concept (e.g. "computers"), to a specialization (e.g. "Apple computer"), but also to a concatenation of several concepts, within a syntactically coherent sentence (e.g. "stores offering bargain Apple computers"), or even a raw list of unconnected terms (e.g. "computers Apple store bargain"). These last examples of *complex queries*, which are usually performed by users of web search engines [45], cannot be directly mapped to concepts in a knowledge base.

Works dealing with query logs have not deeply addressed the analysis of *complex queries* [45]. Some of them [55] simply avoid queries than cannot be directly found in the knowledge base. This reduces the posterior analysis and anonymization to queries with single terms (e.g. "computer") or simple noun phrases (e.g. "cell phone"). Other works [15,33] simply extract individual words from complex queries, mapping each one to their corresponding concept. This method fails to properly interpret noun phrases composed by several words (e.g. the meaning of "water sports" is different to "water" + "sports").

In this work, we apply several linguistic analyses to query logs in order to improve the recall and accuracy of the conceptual mapping. This will help to better characterize the user in the posterior anonymization and, hence, to better retain data utility.

4.1.1. Query processing

Let $U = \{u_1, \ldots, u_m\}$ be the set of users represented by their query logs, and let $u_r = \{q_1, \ldots, q_p\}$ be the queries extracted from the query log of the user u_r (see the input of the method in Fig. 1).

Each of her queries, q_j, is morpho-syntactically analyzed to extract *semantic units*. A semantic unit is a piece of text that refers to a unique concept. In this work, we focus on noun phrases (NPs) consisting of a set of words in which at least one of them (i.e. the one most on the right) is a noun. This noun, which corresponds to a concept (e.g. "sports"), can be specialized by adding other nouns or adjectives on the left (e.g. "water sports"). To extract NPs, we apply several natural language processing methods[2]: *sentence detection*, *tokenization* (i.e. word detection, separating contractions, for example), *part-of-speech (POS) tagging* and *syntactic parsing* (i.e. POS-tagged words are put together according to their role, such as Noun phrase or Verb phrase).

As a result of this process, a query q_j is split into several ones $q_j = \{q_{j1}, \ldots, q_{jl}\}$, each one corresponding to a NP. In this manner, user queries with several NPs are treated as several *individual* queries for anonymization purposes $u'_r = \{q_{11}, \ldots, q_{j1}, \ldots, q_{pl}\}$ (the output of the query processing step in Fig. 1), considering that each one contributes to the semantic characterization of the user.

Example 1. Let the query log of user $u_r = \{q_1, q_2\}$ be:

$$q_1 = \text{“diving in the Mediterranean”},\ q_2 = \text{“windsurfing in the Mediterranean”}$$

After the query processing, the first query is split into two individual queries:

$$q_{11} = \text{“diving”},\ q_{12} = \text{“Mediterranean”}$$

The second one is also split into two individual queries:

$$q_{21} = \text{“windsurfing”},\ q_{22} = \text{“Mediterranean”}$$

Hence

$$u'_r = \{q_{11}, q_{12}, q_{21}, q_{22}\} = \{\text{“diving”}, \text{“Mediterranean”}, \text{“windsurfing”}, \text{“Mediterranean”}\}$$

4.1.2. Conceptual mapping

In order to map individual queries to their conceptual abstractions in a knowledge base, we look for query-concept label matchings. Since words and NPs could be expressed (both in the queries and in the knowledge base) with different linguistic/morphological variations (e.g. "water sport", "water sports", "this water sports", etc.), we apply additional analyses to detect equivalent formulations of the same concept.

1. *Stop words*, that is domain independent words with very general meanings like determinants, prepositions and adverbs, are removed from NPs (e.g. "this water sports" = "water sports").
2. Both queries and concept labels in the knowledge based are *stemmed* [38] to remove derivational affixes of the same root word (e.g. plurals), identifying equivalent terms (e.g. "water sports" = "water sport").
3. In the case that a query composed by several words is not found in the knowledge base, we try simpler forms by progressively removing adjectives/nouns starting from the one most on the left (e.g. "exciting water sports" → "water sports"). The fact that NPs incorporate qualifiers is quite common in texts but these are hardly covered in knowledge structures which try to model concepts in a general way. With this strategy, we improve the recall of the conceptual mapping while maintaining, up to a degree, the core semantics of the query.

4.1.3. Knowledge base

Semantically-grounded related works either use WordNet [20] or ad-hoc *Value Generalization Hierarchies* (VGHs) [50] in order to map query terms to concepts. As discussed in Section 2, both of them present limitations. On the one hand, WordNet has a low coverage of proper nouns/named entities, which very commonly appear in query logs [45]. On the other hand, the construction of ad-hoc VGHs is costly and unfeasible in environments in which large query log sets are dynamically compiled.

Consequently, we used the Open Directory Service (ODP) as the knowledge base in this work. ODP is a multilingual open content directory of World Wide Web links. The purpose of ODP is to list and categorize web sites. It uses an ontology scheme to classify sites into different subjects. Manually created *categories* are *taxonomically structured* and associated with related web resources. It is constructed and maintained by a vast, global community of volunteer editors. The advantage is its large size and high recall, with more than 1 million categories covering up-to-date recently minted terms and named entities. ODP data files can be downloaded in SQL format and categories can be consulted off-line efficiently. Hence, users are mapped to categories, using ODP.

[2] http://opennlp.sourceforge.net/projects.html (last checked on January 18th, 2013).

Being $u'_r = \{q_{11}, \ldots, q_{j1}, \ldots, q_{pl}\}$ the query log of the user u'_r after the query processing, let $C_{u_r} = \{\langle c_1, \omega_1 \rangle, \ldots, \langle c_i, w_i \rangle, \ldots, \langle c_n, \omega_n \rangle\}$ be the categories obtained from the query log of the user u'_r (see the conceptual mapping step in Fig. 1), where $\langle c_i, w_i \rangle$ define a *value tuple* in which c_i is each distinct category obtained from the set of queries of user u'_r, and ω_i is its number of repetitions. Note that different queries could be mapped to the same category. As a result, $\zeta_u = \{C_{u_1}, \ldots, C_{u_r}, \ldots, C_{u_m}\}$ is the set of users represented by categories.

Finally, let us define $T(c_i) = \{c_j \in ODP | c_j \textit{ generalizes } c_i\} \cup \{c_i\}$ as the taxonomic generalizations of the category c_i in ODP, including c_i, which summarizes the meaning of c_i.

Example 2. Following Example 1, we have:

$$u'_r = \{q_{11}, q_{12}, q_{21}, q_{22}\} = \{\text{“diving”}, \text{“Mediterranean”}, \text{“windsurfing”}, \text{“Mediterranean”}\}$$

where individual queries are:

$$q_{11} = \text{“diving”}, q_{12} = \text{“Mediterranean”}, q_{21} = \text{“windsurfing” and } q_{22} = \text{“Mediterranean”}$$

Those are mapped to ODP obtaining categories:

$$C_{u_r} = \{\langle \text{“Swimming and Diving”}, 1 \rangle, \langle \text{“Mediterranean”}, 2 \rangle, \langle \text{“Windsurfing”}, 1 \rangle\}$$

Corresponding to each category, the following taxonomic generalizations are also retrieved from ODP:

$$\begin{aligned} T(\text{“Swimming and Diving”}) &= \{\text{“Sports”}, \text{“Water Sports”}, \text{“Swimming and Diving”}\} T(\text{“Mediterranean”}) \\ &= \{\text{“Regional”}, \text{“Europe”}, \text{“Regions”}, \text{“Mediterranean”}\} \\ T(\text{“Windsurfing”}) &= \{\text{“Sports”}, \text{“Water Sports”}, \text{“Windsurfing”}\} \end{aligned}$$

4.2. Semantic data partition

Data partition pursues to create clusters of users' query logs so that each cluster contains, at least, k users (k is a parameter required by microaggregation methods, see Fig. 1). The MDAV microaggregation method has been used to achieve this goal. As explained in Section 3, MDAV relies on two basic functions that depend on the type of data to be processed: a *comparison* operator that measures the *distance* between records to add new ones in a cluster, and an *averaging* function to calculate the *centroid* used to create clusters.

Due to the characteristics of query logs, the adaptation of MDAV to this kind of data is not trivial. First, contrary to numerical data that can be compared, averaged and transformed by means of mathematical functions, textual queries require from operators that consider their semantics [28]. Moreover, as queries define set-valued datasets with variable length and possible value repetitions, the coherent comparison/aggregation of query logs with different cardinalities and value distributions is also challenging.

In this section, we propose semantically-grounded comparison and averaging operators that are able to consider the distributional characteristics of set-valued data. Our goal is to make the MDAV-based data partition to capture both the meaning and the distribution of query logs, so that the information loss resulting from the posterior anonymization can be minimized.

4.2.1. Query comparison

As a result of the query processing and the conceptual mapping, the query log of each user is represented by a set of categories with their corresponding taxonomical generalizations. This semantics should be considered to coherently compare *individual* categories to which queries belong. In this section, we present a measure that computes the *semantic distance* between categories, according to their taxonomical trees.

In the area of Computational Linguistics, it is worth mentioning the research related to the estimation of *semantic similarity/distance* between terms, based on the exploitation of the taxonomical structure of an ontology [5,41–43,53]. These measures better and more accurately capture the semantic resemblance of textual terms than methods based on terminological comparisons, which are limited to equality/inequality predicates [30].

Classical ontology-based methods estimate the distance between terms, according to the number of taxonomical generalizations/specializations that are needed to go from one term to another. This is equivalent to computing the length of the minimum taxonomical path defined between the pair of terms. However, due to their simplicity, they omit much of the taxonomical knowledge explicitly modeled in the knowledge base, achieving a relatively low accuracy [43]. More recent works [5,43] significantly improve these basic methods by evaluating all the taxonomical ancestors of the compared terms: they measure the distance between terms as a function of the amount of their shared and non-shared taxonomical generalizations.

Considering that the ontological scheme of ODP provides detailed taxonomical structures, our distance measure is designed based on the same principles. Given a pair of categories c_1, c_2 (each one representing a query), we evaluate their distance $\delta_s(c_1, c_2)$ according to the amount of non-shared taxonomical generalizations in ODP. Moreover, we can also presume that category pairs that have many generalizations in common are less distant than those sharing a little amount of generalizations. Hence, the semantic distance is computed as the ratio between the amount of non-shared categories and the sum of shared and non-shared categories (1).

Definition (*Semantic distance between categories δs*). The semantic distance between a pair of categories c_1, c_2 is defined as:

$$\delta_s(c_1, c_2) = \frac{|T(c_1) \cup T(c_2)| - |T(c_1) \cap T(c_2)|}{|T(c_1) \cup T(c_2)|} \tag{1}$$

Note that, by including the compared categories in *T*, we are able to distinguish different categories that have all their generalizations in common from two identical categories.

Example 3. Given the pair of categories:

c_1 = "Swimming and Diving"
c_2 = "Windsurfing"

We have, as shown in Example 2, that:

$T(c_1) = \{$"Sports", "Water Sports", "Swimming and Diving"$\}$
$T(c_2) = \{$"Sports", "Water Sports", "Windsurfing"$\}$

Hence, the semantic distance between categories (Eq. (1)) is computed as:

$\delta_s(c_1, c_2) = ((4-2)/4) = 0.5.$

4.2.2. User query log comparison

The proposed measure (1) has to be extended to semantically compare pairs of user query logs (instead of individual queries) so that they can be clustered by means of the MDAV algorithm. However, since query logs of different users may have different lengths and value distributions, it is necessary to integrate, in a coherent manner, distance values between sets of different cardinalities.

The coherent integration is based on the fact that psychological studies have shown that people pay more attention to the similar features between entities rather than their differences [17]. Considering categories of user queries as user features, given a category c_i of u_1, we compare it against all categories of u_2, taking the minimum distance value as the result of this comparison, because it states the *highest evidence of similarity* between users with respect to feature c_i.

Note that multiple occurrences of the same category may appear in a query log, either because a user repeats a query or because several queries are mapped to the same category (i.e. they represent the same concept, such as "Varicella" and "Chicken Pox"). The distribution of queries/categories in a query log is also an important feature of the user and, together with category semantics, should be considered and preserved by anonymization methods [11]. In order to consider the category distribution, distance measurements for each different category are multiplied by its number of repetitions (2).

$$\Delta_s(c_i, C_{u_2}) = \omega_i \times \underset{j=1}{\overset{|C_{u_2}|}{Min}}(\delta_s(c_i, c_j)), \quad < c_i, \omega_i > \in C_{u_1} \tag{2}$$

where the cardinality $|C_{u_2}|$ is the number of different categories of user u_2.

By repeating the process and adding the distance value between each c_i of u_1 against u_2, we obtain the aggregated distance from u_1 to u_2. Note that this distance may be different when evaluating it from u_2 to u_1. Hence, the final distance between u_1 and u_2 will be the sum between the distances computed from u_1 to u_2 and from u_2 to u_1 (3)

$$D_s(C_{u_1}, C_{u_2}) = \sum_{i=1}^{|C_{u_1}|} \omega_i \times \underset{j=1}{\overset{|C_{u_2}|}{Min}}(\delta_s(c_i, c_j)) + \sum_{j=1}^{|C_{u_2}|} \omega_j \times \underset{i=1}{\overset{|C_{u_1}|}{Min}}(\delta_s(c_i, c_j)) \tag{3}$$

where $|C_{u_1}|$ is the number of different categories of user u_1.

To obtain normalized distance values between query logs (in the [0, 1] interval) so that different user pairs can be compared regardless the cardinality of their logs, we divide it by the number of categories (including repetitions) of both users.

Definition (*Semantic distance between users* D_s). The semantic distance between users (i.e. the set of categories C_{u_1} and C_{u_2} obtained from the queries of user u_1 and u_2, respectively) is defined as:

$$D_s(C_{u_1}, C_{u_2}) = \frac{\sum_{i=1}^{|C_{u_1}|} \omega_i \times \underset{j=1}{\overset{|C_{u_2}|}{Min}}(\delta_s(c_i, c_j)) + \sum_{j=1}^{|C_{u_2}|} \omega_j \times \underset{i=1}{\overset{|C_{u_1}|}{Min}}(\delta_s(c_i, c_j))}{\sum_{i=1}^{|C_{u_1}|} \omega_i + \sum_{j=1}^{|C_{u_2}|} \omega_j} \tag{4}$$

Example 4. Given the categories of users query logs:

$C_{u_1} = \{<$ "*Swimming and Diving*", $1 >, <$ "*Mediterranean*", $1 >\}$
$C_{u_2} = \{<$ "*Windsurfing*", $1 >, <$ "*Mediterranean*", $2 >\}$

Considering the following taxonomic generalizations:

T("Swimming and Diving") = {"Sports", "Water Sports", "Swimming and Diving" }
T("Windsurfing") = {"Sports", "Water Sports", "Windsurfing"}
T("Mediterranean") = {"Regional", "Europe", "Regions", "Mediterranean"},

The distance between users is computed as:

$$D_s(C_{u_1}, C_{u_2}) = (((1 \times 0.5) + (1 \times 0) + (1 \times 0.5) + (2 \times 0))/(2+3)) = 1/5$$

4.2.3. Centroid calculus

As detailed in Section 3, the MDAV algorithm creates clusters (the output of the semantic data partition in Fig. 1) by picking up the most distant record to the dataset centroid. The centroid is understood as the value (or value set in our case) that minimizes the distance against all records in the dataset.

When dealing with continuous-scale numerical data, the centroid can be accurately computed by averaging values. However, for textual data, the centroid must necessarily be discretized. In this case, some authors [14] select the centroid of textual/categorical datasets by picking up the most frequently appearing (i.e. the mode). However, this approximation omits the semantics of data.

Given that the distance measure presented above evaluates both the semantic and distributional features of query logs, we use it to compute the dataset centroid, which is selected as the user query log that minimizes the sum of distances to all other query logs in the dataset.

Definition (*Centroid*). Given the distance function D_s, the centroid of a set of users represented by categories $\zeta_u = \{C_{u_1}, \ldots, C_{u_r}, \ldots, C_{u_m}\}$ is defined as:

$$centroid(C_{u_1}, \ldots, C_{u_r}, \ldots, C_{u_m}) = \arg\min_{C_{u_r} \in \zeta_u} \left\{ \sum_{i=1}^{m} D_s(C_{u_r}, C_{u_i}) \right\} \tag{5}$$

As a result of applying the MDAV algorithm, users' query logs will be grouped into $d = m/k$ clusters of, at least, k elements (i.e., users). We define $P = \{p_1, \ldots, p_d\}$ as the partition of clusters obtained (see Fig. 1).

4.3. Query log anonymization

To fulfill the *k*-anonymity property, the last step (see Fig. 1) requires to replace all query logs of each cluster by a representative query set. Ideally, this representative should be the most similar (or least distant) to all elements in the cluster, so that the information loss resulting from that replacement can be minimized.

In a general microaggregation scenario, this representative is usually calculated as the centroid of the cluster [13,28]. However, in the query log anonymization context, selecting a representative as given by Eq. (5) may lead to undesirable consequences. First, the fact that the centroid corresponds to the query log of a concrete user may excessively expose her, especially if an attacker has partial knowledge (e.g. some user queries are known) [20]. Moreover, since the representative can only be picked from the set of original users, it may not accurately represent the average distributional features of the represented group.

To palliate some of these problems, in some works [50] the representative is synthetically built by replacing *queries* in clusters with *concepts* that generalize all/some of them, according to a background taxonomy. Hence, anonymized query logs would be composed by sets of concepts rather than real queries. This fact hampers the utility of the anonymized logs in some environments in which queries (instead of their conceptual abstraction) are needed, such as query formulation analysis [2,54].

In this work, we have also opted for creating synthetic query log representatives that do not correspond to the log of any concrete user to minimize her disclosure risk. However, we maintain individual queries untouched while retaining, as much as possible, the semantic and distributional characteristics of the represented cluster. The creation of synthetic query log representatives is formalized as follows.

Let $C_{p_t} = \cup_{C_{u_r} \in p_t} C_{u_r}$ be the set of users (represented by their categories) belonging to a cluster p_t. The construction of the representative for that cluster, that is $\bar{p}_t$, follows these steps:

1. First, we select a sole category that semantically represents the cluster p_t. This category, z_t, is the one that minimizes the sum of distances δ_s to all other categories corresponding to the users in the cluster, that is, C_{p_t}. Hence, z_t corresponds to the centroid category of the cluster p_t. In order to consider both semantic and distributional features in the selection of z_t, semantic distances, δ_s, are weighted by the number of repetitions of each category in the cluster, as follows:

   Definition. *(Centroid Category zt)*. Following the same arguments as in Section 4.2.3, the centroid category z_t of a cluster p_t is calculated as:

$$z_t = arg\ min_{c_i \in C_{p_t}} \left\{ \sum_{j=1}^{|C_{p_t}|} \omega_j \times \delta_s(c_i, c_j) \right\} \tag{6}$$

2. Then, the representative $\bar{p}_t$ of a cluster p_t is built from the subset of categories in the cluster that are most similar to the centroid category z_t. This is done from two perspectives:
   a. First, the categories of each user are sorted according to its distance with the above-selected centroid category z_t (1).
   b. Second, to construct $\bar{p}_t$ in a way that it also reflects the distribution of categories in the cluster, we compute the contribution (*quota*) that each user u_r in p_t should have in the representative. This *quota* states the number of semantically similar categories from C_{u_r} (with respect to z_t, according to Eq. (1)) that a user u_r in p_t will contribute to $\bar{p}_t$. The *quota* of each user u_r is computed as the ratio between her number of categories (including repetitions) in C_{u_r} and the number of users in the cluster p_t.

Definition (*Quotaur*). The *Quotaur* of a user u_r is calculated as:

$$Quota(u_r) = \frac{\sum_{i=1}^{|C_{u_r}|} \omega_i}{|p_t|} \tag{7}$$

According to the above criteria, we build the representative $\bar{p}_t$ by picking up *Quota(u_r)* categories (considering their number of repetitions as shown in Eq. (7)) from the sorted list of categories (with respect to the centroid category z_t) of each user u_r in the cluster. In this manner, a proportional number of user categories which are the most semantically similar to the centroid category z_t will be incorporated to the representative. The number of categories picked up for each user in the representative will reflect the original distribution of queries in the input dataset, that is, users with many queries will contribute more than those with a few of them.

3. The final step consists in replacing categories in $\bar{p}_t$ by appropriate queries picked up from the original dataset, so that anonymized data can still be useful for query-analysis tasks [2,54]. Specifically, each category in $\bar{p}_t$ is replaced by a query taken from the whole original dataset corresponding to that category. To minimize the risk of disclosure of individual users, queries are *randomly* picked up from the set of suitable ones (since different queries can correspond to the same category). Thanks to the random criterion, we minimize the chance that the cluster representative contains exact subsequences of queries of individual users, a circumstance that may compromise her anonymity [20]. At the same time, since query's categories match, we also retain semantics in anonymized data.

As a result of the above process, user query logs of each cluster are replaced with a synthetic query log ($\bar{p}_t$) that contains a representative distribution of those queries found in the original dataset, which are also the most semantically similar to all users logs in the cluster. Hence, as shown in the output of the method in Fig. 1, the final result is a set of anoymized user query logs $U^A = \{u_1^A, \ldots, u_m^A\}$.

Notice that since the computational complexity of MDAV is O(m^2) and because the proposed semantic adaptation (using semantic similarity distances to compare query logs) does not affect the overall complexity of the algorithm, our method scales quadratic with respect to the dataset size, making it comparable in terms of scalability to other aggregation-based anonymization methods [15,33].

5. Evaluation

In this section, the evaluation of the proposed method is detailed and compared against related works. First, we present the evaluation measures used to quantify the disclosure risk and utility of anonymized query logs (Section 5.1.). Section 5.2 introduces other query anonymization strategies implemented to compare our results. Section 5.3 presents and discusses the results for the evaluated methods.

5.1. Evaluation measures

As stated in the introduction, anonymization methods should maintain a trade-off between two apposite dimensions: data utility, as an inverse function of information loss, and disclosure risk, that is, the chance of an intruder to disclosure the identity of an individual or de-identification. In this section, the measures used to evaluate these dimensions are detailed.

From a general perspective, the utility of anonymized data is retained if the same conclusions can be extracted from the analysis of the original and anonymized datasets. To evaluate up to which point a query anonymization method retains the utility of data (or minimizes the information loss) in an objective and practical way, we rely on data mining techniques. Data mining aims at extracting useful information by characterizing user profiles or preferences. To do so, data mining techniques, and clustering methods in particular, are used to create groups of homogeneous users. Classically, clustering has focused on numerical and categorical data. However, recent works [4] have developed semantic clustering algorithms to manage both numerical and textual data. These algorithms have previously been used to evaluate the information loss resulting from the anonymization of textual data [30], showing that the degree of data utility retained by the anonymization process can be quantified from a semantic perspective by quantifying the differences between the cluster set obtained from original data against that obtained from the masked version.

In our case, we compute the *information loss* (L) of an anonymization method by comparing the partitions resulting from original and masked query logs when applying a semantic hierarchical clustering using also ODP as the knowledge source. That is, the higher the distance between cluster sets, the more different the data analysis conclusions are and, hence, the higher the information loss and the lower the retained utility. To select the most adequate clustering for the given data, the well-known Calinski and Harabasz index [8] has been used. Differences are quantified according to a well-known distance measure between partitions [27]. Formally, being P_A a partition of the original data, and P_B a partition of the anonymized one, this distance can be defined as (8):

$$dPart(P_A, P_B) = \frac{2 * I(P_A \cap P_B) - I(P_A) - I(P_B)}{I(P_A \cap P_B)} \tag{8}$$

where $I(P_A)$ is the average information of P_A which measures the randomness of the distribution of elements over the set of classes of the partition (similarly for and $I(P_B)$), and $I(P_A \cap P_B)$ is the mutual average information of the intersection of two partitions [27].

Notice that the distance values obtained are normalized in the $[0 \cdots 1]$ interval, where 0 indicates identical clusters and 1 maximally different ones. Also, note that the scale is logarithmic, so that it grows according to the amount of differences observed.

Definition (*Information Loss L*). The percentage of information loss L of masked query logs is quantified as the distance between the obtained partitions for the original (P_A) and anonymized data (P_B), as:

$$L = d_{Part}(P_A, P_B) \times 100 \tag{9}$$

To measure the *disclosure risk* (DR) of anonymized query logs, we rely on one of the most common measures: *Record Linkage* (RL) [34]. It quantifies the amount of records (i.e. query logs) that can be correctly matched between the original dataset and the anonymized one. To do so, it assumes that a potential attacker would match the least distant original and anonymized records by comparing their queries and picking up the pair or pairs (if several result in the same value) with the highest amount of queries in common. Formally, being u_r an original record (i.e., the original query log of an user), u_r^A its anonymized version, and $P_{rl}(u_r^A)$ the record linkage probability of an anonymized record to be disclosed, the percentage of RL (10) is calculated as follows:

$$\begin{aligned} RL &= \frac{\sum_{r=1}^{m} P_{rl}(u_r^A)}{m} \times 100 \\ \text{where } P_{rl}(u_r^A) &= \begin{cases} 0 & if\ u_r \notin G \\ \frac{1}{|G|} & if\ u_r \in G \end{cases} \end{aligned} \tag{10}$$

where G is the set of original records that have been linked to u_r^A. That is, each u_r^A is compared to all records of the original dataset, picking up the pair or pairs with the highest amount of queries in common with u_r^A, thus obtaining the G set of matched records. If u_r is in G, then the probability of record linkage is computed as the probability of finding u_r in G. Otherwise, the record linkage probability is 0.

5.2. Comparison

To evaluate the contribution of our proposal against related works on query anonymization, we have implemented and tested those methods (already introduced in Section 2) that, as ours, are based on microaggregation and those more recent/elaborated methods based on query removal:

- Korolova et al. [23]: a method based on the removal of scarcest queries (i.e. a priori, the most identifying queries) from the dataset.
- Poblete et al. [37]: another method based on query removal, in this case, relying on a graph-based representation of queries.
- Navarro-Arribas et al. [33]: a method based on query microaggregation, proposing several syntactical measures to compare queries. In addition, other query features (like clicked URLs or timestamps) are considered to better differentiate and compare query logs.
- Erola et al. [15]: a method that compares queries at a conceptual level by using ODP categories. Even though a degree of semantics is considered during query aggregation, prototypes are randomly generated.

In addition to the above methods, we have also implemented a simplified version of our proposal in which *no semantics* are considered at all. In this case, neither query processing nor ODP are used. Queries are treated as simple strings and compared according to their equality/inequality. In the aggregation step, the representative query log is built only considering the distribution of queries. This method aims at evaluating the degree of data utility that can be retained when the query aggregation is solely focused on data distribution.

5.3. Results

The evaluation has been done using real query logs extracted from the AOL log released in 2006. From these, the query logs of 1000 users have been randomly been taken. They contain about 56,000 individual queries, of which near the 61% can be considered as complex queries. Even though personal identifiers have been removed from published query logs, as stated in the introduction, queries themselves may enable identity disclosure due to their specificity and personal nature [3].

This dataset has been anonymized by means of our method and those introduced in Section 5.2. For methods based on the k-anonymity model (our proposal, the *non-semantic* version introduced above, and the ones of Navarro et al. and Erola et al.), k-values between 2 and 7, which resulted in 500–142 clusters, have been tested. Note that, due to the high heterogeneity and unbounded nature of query log data (i.e., all query logs define a unique set of queries), even the lowest k-value results in a modification of *all* original query logs after the aggregation process. Hence, most changes are observed for the tested k-value range. Other methods based on query removal (Korolova et al. and Poblete et al.) have been tested varying their corresponding anonymization parameters in reasonable margins (a d value ranging from 1 to 20 for Korolova et. al and a Kp value ranging from 2 to 40 for Poblete et al.). Anonymized datasets for the different approaches and anonymization degrees have been evaluated and compared according to their *information loss* (Fig. 2) and *record linkage* (Fig. 3), as detailed in Section 5.1. Since k-anonymity values and those of the d and Kp parameters used in Korolova et al. and Poblete et al. approaches are not directly comparable, results are shown in different figures. To measure information loss as a function of the distance between data clusterization results, partitions with 80 clusters (according to the Calinski–Harabasz index) have been created.

After the analysis of information loss figures, we immediately observe a very noticeable difference between the Korolova et al. method and the other ones. The former results in high information loss figures, which state that the conclusions of the analysis of anonymized data will significantly differ from those obtained for the analysis of original query logs. As a result, the utility of masked data is severely hampered. The high degree of query removal performed by Korolova et al. (from a 94% of query removal for $d = 1$ to a 97.5% for $d = 20$) is, in fact, the least desirable solution from the data analysis perspective, since the semantics provided by the large amount of removed queries are completely lost in the anonymized dataset. Since it is based on the removal of scarcest queries of the dataset and, considering that most queries appeared once, even for the more relaxed anonymization parameter (d), it resulted in the complete removal of *all* queries for a considerable amount of users (i.e., for $d = 1$, about a 46% of users had all their queries removed, whilst this figure increases to 65% for $d = 5$).

In comparison, the method by Poblete et al. results in lower information loss figures, since its removal strategy is more focused on queries producing too little results, when queried in a search engine (parameter Kp), than on their distribution in the dataset. As a result, it removes a significantly lower amount of queries (i.e., for $Kp = 30$ only a 10% of users had all their queries removed).

In general, methods based on query log aggregation better retained the utility of query logs. Moreover, since all of them are based on the k-anonymity model, their results can be compared under the same conditions. Among them, the worst was

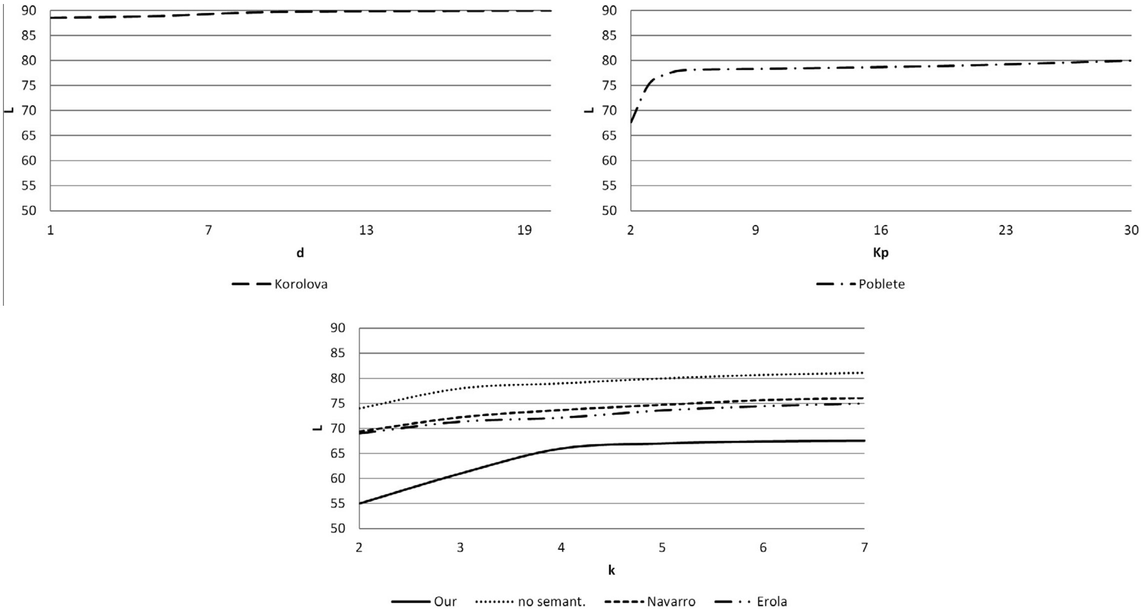

Fig. 2. Information Loss (L).

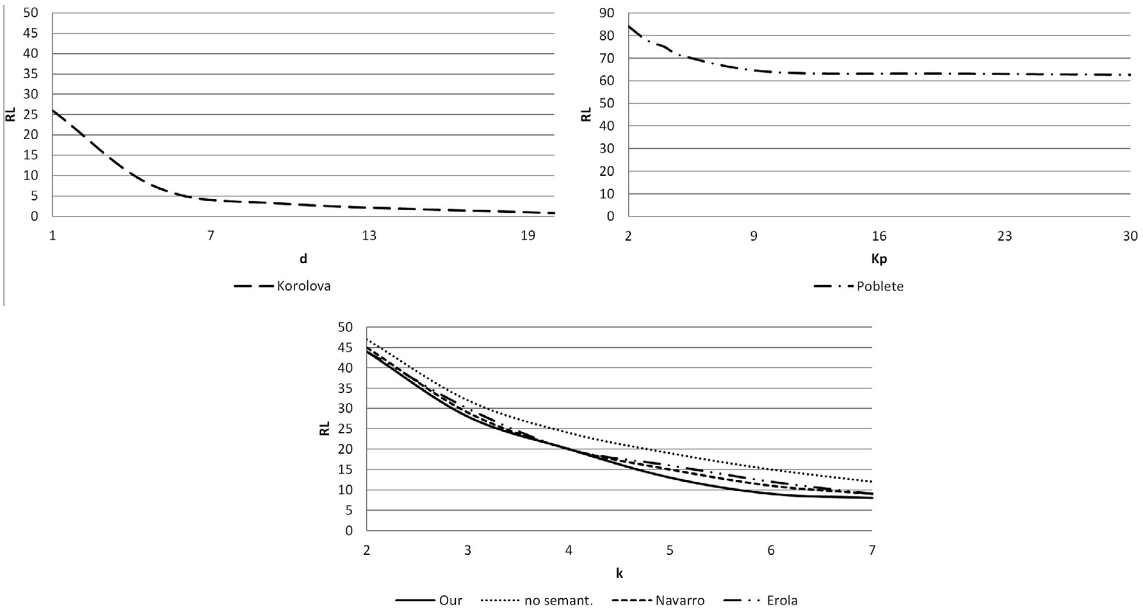

Fig. 3. Record Linkage (RL).

the one based solely on boolean comparisons between queries (i.e., the *non-semantic* implementation of our proposal). Since queries can only be evaluated as identical or different, and, since most queries in the dataset are unique, few evidences of similarity that can guide the partition and aggregation processes can be gathered. In this case, only query distribution (i.e. their number of repetitions) is considered. Better results are obtained for the method by Navarro et al. Even though it is also based on terminological comparisons, it enables a more fuzzy evaluation of query similarity thanks to the use of the Edit distance to compare Strings. This, combined with the evaluation of other query features such as the timestamps or clicked URLs, enables a finer grained comparison between queries and, hence, a more accurate aggregation.

These approaches, however, do not consider query semantics in an explicit way and, even though terminological resemblance is an evidence of semantic similarity, it poorly captures and evaluates the meaning of queries. The approach by Erola et al. exploits ODP to retrieve categories to which queries refer, and evaluates the number of terminologically identical categories between query logs as a measure of similarity. Since categories define a more constrained set of modalities than query logs, the chance of discovering terminological matchings increases and so do the evidences of similarity. Moreover, since categories are conceptualizations of textual queries, this approach enables a semantically-coherent partition of query logs. However, no semantic evidences are used during the aggregation process which is solely focused on the preserving of the distribution of queries.

Finally, our method provides the lowest information loss for all k-values. Also note that since the $d_{Part}(P_A,P_B)$ function used to compute information loss has a logarithmic scale, the absolute differences in partitions for high information loss values are greater than for lower ones. The obtained improvement is the result of considering both the semantics of queries and their distribution in *all* stages of the query anonymization process (i.e. comparison between queries, centroid selection, cluster construction and data aggregation), relying on a semantically-coherent distance measure and ODP categories. Also note that the query processing stage also contributes to interpret semantics of complex queries (i.e. those with several word or noun phrases) more coherently, in comparison with methods that treat queries word by word [15,33].

Disclosure risk evaluates the chance of an intruder to match original and anonymized records, and directly depends on the degree of distortion introduced in the anonymized dataset. Since disclosure risk is based on the degree of overlapping between query logs, the more different they are with respect to original ones, the more private the results will be. After analyzing disclosure risk figures, we observe that the method by Korolova et al. results in the lowest figures (between 25% of matched records for $d = 1$ to almost 0% for $d = 20$). As stated above, this is caused by the great amount of user logs for which *all* queries have been removed. Obviously, if no queries are available, no linkage is possible, but it also becomes useless for statistical and data analysis due to the high information loss.

The method by Poblete et al. results in higher figures (around 63% of linkages for $Kp > 5$). This method removes a significantly lower amount of queries logs (i.e., for $Kp = 30$ only a 10% of the query logs have been completely removed). These results, in combination with the fact that non-removed queries appear "as is" in the masked dataset (i.e. no transformation, swapping or replacement is done), increase the chance of discovering correct linkages with original logs.

Methods based on query aggregation obtain a quite comparable amount of linkages, which decrease almost linearly as the k-anonymity level increases. From these, the one solely based on query distribution (i.e., the *non-semantic* version of our proposal) results in the highest amount of linkages. In this case, the fact that queries can only be evaluated in a Boolean fashion and the fact that query logs are aggregated according to their distributional properties, limit the amount of distortion introduced in the anonymized data and increase the chance of linkage.

Our method is able to minimize the information loss of anonymized data and at the same time it is also able to maintain the amount of linkages as low as the methods based on query aggregation do. In this case, even though semantics of anonymized data are better retained, the fact that the aggregation is made by randomly rearranging queries of different users (while maintaining their distribution and semantics) contributes to maintain record linkage at levels comparable to those of related works. Hence, queries of the masked log can be *different* from those in the original one, but they will cover *similar* topics.

6. Conclusions

Query logs, even though anonymized to mask individual-level information, are of great interest for scientists [2] to study and test new IR algorithms, to learn about user information needs and query formulation approaches [23] and also to investigate on the use of language in queries [49]. Companies and advertisers can also exploit query logs to characterize general user profiles and search habits [7,37] to gain better understanding of their competitors, to improve keyword advertising campaigns [23] and to extract market tendencies and trending topics [19].

This paper presents a novel query log anonymization method based on semantic microaggregation. While most of the research on privacy protection of queries focuses on minimizing disclosure risk, our method has been especially designed to retain the utility of anonymized queries. To achieve this goal, queries are semantically interpreted by extracting their conceptualizations from the ODP structured set of categories. This enables to aggregate query logs semantically by means of an adaptation of the MDAV algorithm. Suitable semantic operators to compare and average query logs have been proposed for that purpose. Finally, a method to generate synthetic query logs composed by *real* queries, which replace original ones, is also proposed. This method preserves the semantics and distribution of original queries, while keeping disclosure risk at a reasonable level.

The evaluation carried out with a set of real query logs extracted from the AOL dataset sustains the suitability of our method. Compared to related works based on query removal or non-semantic query aggregation, our proposal better retained data utility while maintaining a desirable level of disclosure risk.

As future work, we plan to combine ODP with other general-purpose knowledge bases, such as DBPEDIA,[3] in order to improve the recall of the conceptual mapping of queries. Since our method is general enough to be applied to set-valued data other than query logs, we plan to test its behavior in other domains in which textual transactional data is available (such as electronic health-care records), exploiting domain-specific knowledge bases (such as biomedical terminologies like SNOMED-CT [47]).

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, ICWT TIN2012-32757, 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), by the Spanish Ministry of Industry, Commerce and Tourism (through projects eVerification2 TSI-020100-2011-39 and SeCloud TSI-020302-2010-153) and by the Government of Catalonia (under Grant 2009 SGR 1135).

References

[1] E. Adar, User 4xxxxx9: Anonymizing query logs, in: Proceedings of the Query Log Analysis: Social and Technological Challenges Workshop at the 16th World Wide Web Conference, WWW2007, Banff, Alberta, Canada, 2007.
[2] J. Bar-Ilan, Access to query logs – an academic researcher's point of view, in: Proceedings of the Query Log Analysis: Social and Technological Challenges Workshop at the 16th World Wide Web Conference, WWW2007, Banff, Alberta, Canada, 2007.
[3] M. Barbaro, T. Zeller, A face is exposed for AOL searcher no. 4417749, in. The New York Times, 2006.
[4] M. Batet, Ontology-based semantic clustering, AI Communications 24 (2011) 291–292.
[5] M. Batet, D. Sánchez, A. Valls, An ontology-based measure to compute semantic similarity in biomedicine, Journal of Biomedical Informatics 44 (2011) 118–125.
[6] S.M. Beitzel, E.C. Jensen, A. Chowdhury, D. Grossman, O. Frieder, Hourly analysis of a very large topically categorized web query log, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research And Development in Information Retrieval, ACM New York, Sheffield, UK, 2004, pp. 321–328.
[7] D.J. Brenes, D. Gayo-Avello, Stratified analysis of AOL query log, Information Sciences 179 (2009) 1844–1858.
[8] R.B. Calinski, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics 3 (1974) 1–27.
[10] A. Cooper, A survey of query log privacy-enhancing techniques from a policy perspective, ACM Transactions on the Web 2 (2008) 19:11–19:27.
[11] J. Domingo-Ferrer, A survey of inference control methods for privacy preserving data mining, in: Privacy Preserving Data Mining: Models and Algorithms, 2008.
[12] J. Domingo-Ferrer, A. Martínez-Ballesté, J.M. Mateo-Sanz, F. Sebé, Efficient multivariate data-oriented microaggregation, The VLDB Journal 15 (2006) 355–369.

[3] http://dbpedia.org/About.

[13] J. Domingo-Ferrer, J.M. Mateo-Sanz, Practical data-oriented microaggregation for statistical disclosure control, IEEE Transactions on Knowledge and Data Engineering 14 (2002) 189–201.
[14] J. Domingo-Ferrer, V. Torra, Ordinal, continuous and heterogeneous *k*-anonymity through microaggregation, Data Mining and Knowledge Discovery 11 (2005) 195–212.
[15] A. Erola, J. Castellà-Roca, G. Navarro-Arribas, V. Torra, Semantic microaggregation for the anonymization of query logs, in: Privacy in Statistical Databases. Lecture Notes in Computer Science, 6344, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 127–137.
[16] C. Fellbaum, WordNet: An Electronic Lexical Database, MIT Press, Cambridge, Massachusetts, 1998.
[17] R.L. Goldstone, Similarity, in: R.A. Wilson, F.C. Keil (Eds.), The MIT Encyclopedia of the Cognitive Sciences, MIT Press, Cambridge, MA, 1999.
[18] N. Guarino, Formal ontology in information systems, in: N. Guarino (Ed.), 1st International Conference on Formal Ontology in Information Systems, FOIS-98, IOS Press, Trento, Italy, 1998, pp. 3–15.
[19] S. Hansell, Increasingly, internet's data trail leads to court, in: The New York Times, 2006.
[20] Y. He, J.F. Naughton, Anonymization of set valued data via top down, local generalization, PVLDB (The Proceedings of the VLDB Endowment) 2 (2009) 934–945.
[21] X. Jing, N. Zhang, G. Das, ASAP: Eliminating algorithm-based disclosure in privacy-preserving data publishing, Information Systems 36 (2011) 859–880.
[22] R. Jones, R. Kumar, B. Pang, A. Tomkins, I know what you did last summer – query logs and user privacy, in: Proceedings of the Sixteenth Conference on Information and Knowledge Management, CIKM2007, ACM New York, Lisbon, Portugal, 2007, pp. 909–914.
[23] A. Korolova, K. Kenthapadi, N. Mishra, A. Ntoulas, Releasing search queries and clicks privately, in: Proceedings of the 18th International World Wide Web, WWW2009, ACM, Madrid, Spain, 2009, pp. 171–180.
[24] R. Kumar, J. Novak, B. Pang, A. Tomkins, On anonymizing query logs via token-based hashing, in: Proceedings of the 16th International World Wide Web Conference, WWW2007, ACM, Banff, Alberta, Canada, 2007, pp. 629–638.
[25] J.N.K. Liu, Y.-L. He, E.H.Y. Lim, X.-Z. Wang, Domain ontology graph model and its application in Chinese text classification, Neural Computing and Applications 19 (2012) 1–20.
[26] J.N.K. Liu, Y.-L. He, E.H.Y. Lim, X.-Z. Wang, A new method for knowledge and information management domain ontology graph model, IEEE Transactions on Systems, Man, and Cybernetics: Systems 43 (2013) 115–127.
[27] R. López de Mántaras, A distance-based attribute selection measure for decision tree induction, Machine Learning 6 (1991) 81–92.
[28] S. Martínez, D. Sánchez, A. Valls, Semantic adaptive microaggregation of categorical microdata, Computers & Security 31 (2012) 653–672.
[29] S. Martínez, D. Sánchez, A. Valls, A semantic framework to protect the privacy of electronic health records with non-numerical attributes, Journal of Biomedical Informatics 46 (2013) 294–303.
[30] S. Martínez, D. Sánchez, A. Valls, M. Batet, Privacy protection of textual attributes through a semantic-based masking method, Information Fusion 13 (2012) 304–314.
[31] S. Martínez, A. Valls, D. Sánchez, Semantically-grounded construction of centroids for datasets with textual attributes, Knowledge Based Systems 35 (2012) 160–172.
[32] N. Matatov, L. Rokach, O. Maimon, Privacy-preserving data mining: a feature set partitioning approach, Information Sciences 180 (2010) 2696–2720.
[33] G. Navarro-Arribas, V. Torra, A. Erola, J. Castellà-Roca, User k-anonymity for privacy preserving data mining of query logs, Information Processing and Management 48 (2012) 476–487.
[34] J. Nin, J. Herranz, V. Torra, On the disclosure risk of multivariate microaggregation, Data Knowledge Engineering 67 (2008) 399–412.
[35] A. Oganian, J. Domingo-Ferrer, On the complexity of optimal microaggregation for statistical disclosure control, Statistical Journal of the United Nations Economic Commision for Europe 18 (2001) 345–353.
[36] S. Paul, P. Maji, Gene ontology based quantitative index to select functionally diverse genes, International Journal of Machine Learning and Cybernetics 27 (2012) 1–18.
[37] B. Poblete, M. Spiliopoulou, R. Baeza-Yates, Privacy-preserving query log mining for business confidentiality protection, ACM Transactions on the Web 4 (2010) 10:11–10:26.
[38] M.F. Porter, An algorithm for suffix stripping, in: Readings in Information Retrieval, Morgan Kaufman Publishers Inc., San Francisco, 1997. pp. 313–316.
[39] P. Samarati, Protecting respondents' identities in microdata release, IEEE Transactions on Knowledge and Data Engineering 13 (2001) 1010–1027.
[40] D. Sánchez, A methodology to learn ontological attributes from the Web, Data & Knowledge Engineering 69 (2010) 573–597.
[41] D. Sánchez, M. Batet, A new model to compute the information content of concepts from taxonomic knowledge, International Journal on Semantic Web and Information Systems 8 (2012) 34–50.
[42] D. Sánchez, M. Batet, D. Isern, Ontology-based information content computation, Knowledge-Based Systems 24 (2011) 297–303.
[43] D. Sánchez, M. Batet, D. Isern, A. Valls, Ontology-based semantic similarity: a new feature-based approach, Expert Systems with Applications 39 (2012) 7718–7728.
[44] D. Sánchez, M. Batet, A. Viejo, Automatic general–purpose sanitization of textual documents, IEEE Transactions on Information Forensics and Security, in press. http://dx.doi.org/10.1109/TIFS.2013.2239641.
[45] D. Sánchez, J. Castellà-Roca, A. Viejo, Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines, Information Sciences 218 (2012) 17–30.
[46] D. Sánchez, D. Isern, M. Millán, Content annotation for the semantic web: an automatic web-based approach, Knowledge and Information Systems 27 (2011) 393–418.
[47] K.A. Spackman, SNOMED CT milestones: endorsements are added to already-impressive standards credentials, Healthcare Informatics 21 (2004) 54–56.
[48] L. Sweeney, *K*-Anonymity: a model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge Based Systems 10 (2002) 557–570.
[49] B. Tancer, Click: What Millions of People Are Doing Online and Why it Matters, Hyperion, 2008.
[50] M. Terrovitis, N. Mamoulis, P. Kalnis, Privacy-preserving anonymization of set-valued data, Proceedings of the VLDB Endowment, PVLDB 1 (2008) 115–125.
[51] V. Torra, Towards knowledge intensive data privacy, in: Proceedings of the 5th International Workshop on Data Privacy Management, and 3rd International Workshop on Autonomous Spontaneous Security, DMP'10/SETOP'10., Springer-Verlag, Athens, Greece, 2011, pp. 1–7.
[52] A. Valls, K. Gibert, D. Sánchez, M. Batet, Using ontologies for structuring organizational knowledge in Home Care assistance, International Journal of Medical Informatics 79 (2010) 370–387.
[53] Z. Wu, M. Palmer, Verb semantics and lexical selection, in: 32nd annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Las Cruces, New Mexico, 1994, pp. 133–138.
[54] L. Xiong, E. Agichtein, Towards privacy-preserving query log publishing, in: Proceedings of the Query Log Analysis: Social and Technological Challenges Workshop at the 16th World Wide Web Conference, WWW2007, Banff, Alberta, Canada, 2007.
[55] Y. Xu, K. Zhang, Z. Chen, K. Wang, Privacy-Enhancing Personalized Web Search, in: Proceedings of the 16th International World Wide Web Conference, WWW2007, Banff, Alberta, Canada, 2007, pp. 591–600.
[56] Z. Zhong, Z. Liu, C. Li, Y. Guan, Event ontology reasoning based on event class influence factors, International Journal of Machine Learning and Cybernetics 3 (2012) 133–139.

Further reading

[9] R.L. Cilibrasi, P.M.B. Vitányi, The google similarity distance, IEEE Transactions on Knowledge and Data Engineering 19 (2006) 370–383.