Enforcing Vocabulary k-Anonymity by Semantic Similarity Based Clustering

Junqiang Liu*†, Ke Wang*
*Simon Fraser University, Burnaby, British Columbia V5A 1S6, Canada
Email: {jjliu, wangk}@cs.sfu.ca
†Zhejiang Gongshang University, Hangzhou, 310018, China

***Abstract*—Web query logs provide a rich wealth of information, but also present serious privacy risks. We consider publishing vocabularies, bags of query-terms extracted from web query logs, which has a variety of applications. We aim at preventing identity disclosure of such bag-valued data. The key feature of such data is the extreme sparsity, which renders conventional anonymization techniques not working well in retaining enough utility. We propose a semantic similarity based clustering approach to address the issue. We measure the semantic similarity between two vocabularies by a weighted bipartite matching and present a greedy algorithm to cluster vocabularies by the semantic similarities. Extensive experiments on the AOL query log show that our approach retains more data utility than existing approaches.**

***Keywords*-Anonymity; privacy; bag-valued data; query logs**

I. Introduction

Web search has become the most essential tool for people to find information in their daily lives. Web query logs retained by search engines provide a rich wealth of information extremely useful as a research or marketing tool [4][16][19], but they also present serious privacy risks [1]. A query log published in its entirety is extremely vulnerable to privacy attacks [13][11]. Publication scenarios and sound privacy guarantees are application dependent.

We publish the vocabularies extracted from a web query log and protect privacy in such a scenario. A *vocabulary* is the bag of query-terms derived by merging queries issued by the same user. Such data has a variety of applications, including web search personalization, advertisement, query suggestion, and query spelling correction [4][19]. We allow a spectrum of granularities in merging queries since a different application may require a different granularity level [19]. For example, for web search personalization the granularity may be the user level, i.e., all the queries by each user may be merged into one vocabulary, while for query suggestion one possible granularity can be the session level, i.e., all the queries of each user at one session comprise an independent vocabulary.

We treat a vocabulary as a bag rather than a set since the number of occurrences of a query-term is very important. For example, consider two users who both issued 100 queries. For the first user, 99 queries included *apple* and 1 query included *bread*, while for the second user the reverse is true (1 *apple* and 99 *bread*). The two users have quite different interests,which can only be documented by bags.

The main privacy risk in publishing such bag-valued vocabularies is that an adversary may identify his/her target individual's vocabulary based on his/her knowledge about the presence and the number of occurrences of certain query-terms in the target vocabulary, and hence may be able to make further inference. We are concerned with preventing such identity disclosure as it is impossible for web search users to reach an agreement on what query-terms are sensitive and what are non-sensitive. Thus, we extend the *k-anonymity* principle [17] to our scenario to ensure that every vocabulary for a given granularity is indistinguishable from at least $k-1$ other vocabularies. Let us call such a requirement the *vocabulary k-anonymity* principle.

A key feature of vocabularies is the extreme sparsity as people hardly use the same query-terms even for the same concepts. This feature renders the conventional techniques, such as the generalization techniques [8][18] and the suppression technique [20], not working well in achieving vocabulary k-anonymity in the sense of retaining data utility, as explained as follows.

A. Motivations

As an anecdotal example, consider publishing vocabularies extracted from the AOL query log [16] with the granularity at the user level. For instance, the first row in Table I shows in part the original vocabulary of a user consisting of 181 distinct query-terms. We applied the multi-dimensional generalization algorithm LG [8] on the extracted vocabularies to enforce the vocabulary k-anonymity with $k = 5$. Unfortunately, the resulting content by LG [8] is overly generalized. For instance, the anonymized vocabulary of the same user, as in the second row, shrinks into 21 general terms, such as activity, artifact, attribute, etc., which are too general to convey useful information. We also applied another common approach, i.e., suppression [20]. The result is that over 90% query-terms have to be suppressed.

The existing techniques cannot retain enough data utility in anonymizing such bag-valued data because of two issues that have yet to be addressed properly.

The first issue is that generalization and suppression work well only in specific situations, which however is not the case with vocabularies extracted from web query logs. Generalization [8][18] can remove an outlier term by generalizing it together with its sibling terms on a taxonomy

Table I
ONE USER'S VOCABULARY EXTRACTED FROM AOL QUERY LOG

Original vocabulary	care:1, package:1, movie:2, dog:3, blue:2, book:1, school:1, child:1, supply:1, . . .
Generalized one by LG [8]	activity, artifact, attribute, organism, social-event, . . .

tree. Thus, generalization works well when the taxonomy tree of terms is deep (a large tree height and small fan-outs). Unfortunately, in the context of web search, the taxonomy tree of query-terms is quite shallow relative to the huge domain of query-terms. Suppression [20] can remove each outlier term independently, thus, works well when there are a small number of outlier terms. However, vocabularies extracted from web query logs are extremely sparse and are full of outliers.

The second issue is that existing works treat bags as sets and underestimate the information loss in such a treatment [8][18]. Vocabularies are bags, if treated as sets, they shrink drastically with generalization [8][18]. For example, when an *apple* and an *orange* in a set are both generalized to *fruits*, only one occurrence of *fruits* is kept in the generalized set by [8][18], whereas the information loss is computed as if the two occurrences of *fruits* were all kept by [8][18]. In the foregoing anecdotal example, the information loss computed by [8] is 12% in LM [10], however the actual information loss is much higher, that is, the distortion to the numbers of occurrences of terms is ignored (more discussion in Section V-A).

B. Our Contributions

We propose the vocabulary k-anonymity principle. Such a notion is tailored in the context of web query logs, and we allow a different granularity in merging queries into vocabularies for a different application. The subtlety is that vocabularies are a collection of bags rather than a collection of sets. Anonymization of such bag-valued data is a new problem different from anonymization of set-valued data [8][18][20][6] and anonymization of relational data [17].

We propose the semantic similarity based clustering approach for addressing the first issue with the conventional techniques, i.e., to retain more data utility. The intuition behind our approach is that although people hardly use the same query-terms even for the same concepts, those terms should be semantically similar to each other. We measure the semantic similarity between two vocabularies by a bipartite matching with the minimum semantic distance. Such an approach retains more data utility than the state of the art approach in terms of information loss metric and the usefulness in frequent pattern mining.

To address the second issue with generalization techniques in anonymizing such bag-valued data, we show how to extend the generalization technique [8] for bag-valued data, and present a bag-valued version of the general loss metric

Id	Query	Time	Rank	URL
01	Wine	1:00Jan1	3	a.c
01	Wine	1:00Jan1	8	b.c
01	Jackets, Boots	1:15Jan1	3	c.c
01	Jackets, Boots	1:15Jan1	5	d.c
01	Wine	2:05Jan1	4	x.c
02	...	...	...	...

(a) Q: original web query log

Id	Vocabulary of each user
t_1	Wine, Wine, Jackets, Boots
t_2	Vino, Vino, Jackets, Shoes
t_3	Wine, Vino, Raw-milk, Jackets, Shoes
t_4	Vino, Raw-milk, Raw-milk, Homo-milk
t_5	Raw-milk,Homo-milk,Homo-milk,Jackets,Pants

(b) T: vocabularies extracted from Q

Figure 1. Vocabularies extracted from a query log

[10] to gauge the information loss properly in such a context as detailed in Section V-A.

The rest of the paper is organized as follows. Section II defines the privacy principle and our anonymization model, Section III discusses the measurement of semantic similarity, Section IV presents our greedy algorithm, Section V discusses data utility metrics and revisits the generalization techniques, Section VI evaluates our approach, Section VII surveys related works, and Section VIII concludes the paper.

II. PRIVACY NOTION AND ANONYMIZATION MODEL

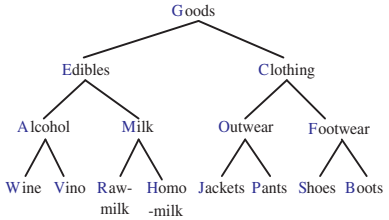
A. Vocabulary k-Anonymity

A search service provider wants to release the collection T of vocabularies extracted from a query log Q. As the number of occurrences of a query-term is important information, the provider treats the vocabularies as bags instead of sets. The provider considers a spectrum of granularities in merging queries into vocabularies.

One way to specify the granularity is by the gap between the times queries were posted, which is an approach also documented in web mining literature. Given a query log Q in the format as shown in Figure 1 (a) where for q in Q, $q.Id$ is the anonymous ID of the user issuing q, $q.Query$ is the set of query-terms in q, and $q.Time$ is the time q was posted for search, vocabularies can be extracted based on an application dependent logical session gap, $sGap$, as follows.

Definition 1 (Collection T of vocabularies): Given a logical session gap $sGap$ and $Q = \{q_1, q_2, \ldots, q_n\}$ with queries in ascending order of Id and $Time$, queries in Q are distributed into groups G_j. Suppose a query q_i is distributed to G_j. The next query q_{i+1} is distributed to G_{j+1} if $q_i.Id < q_{i+1}.Id$ or $q_i.Time + sGap < q_{i+1}.Time$, otherwise q_{i+1} is also distributed to G_j. The bag containing all the query-terms in $q.Query$ for $q \in G_j$ is the jth vocabulary in $T = Vocabulary(Q, sGap)$. $\diamond$

By Definition 1, each vocabulary consists of all the terms in the queries issued by the same user within $sGap$. For example, Figure 1 (a) shows three queries issued by a user, the first query about {Wine} has two clicked search results, i.e., the first two lines in Figure 1 (a) with URL "a.c" and "b.c", and so on. Figure 1 (b) shows the vocabularies extracted with $sGap = \infty$, i.e., one vocabulary per user.



(a) Taxonomy for generalization

Id	LG	LG (bag)	Ours
t_1	G	G, G, G, G	W, V, J, S
t_2	E, C	E, E, C, C	W, V, J, S
t_3	E, C	E, E, C, C	W, V, J, S
t_4	G	G, G, G, G	R, R, H
t_5	E, C	E, E, C, C	R, R, H

(b) T^*: anonymized vocabularies

Figure 2. Anonymizing vocabularies by LG and our approach

The provider protects the identities of search service users by observing the following privacy principle.

Definition 2 (Vocabulary k-anonymity): Given a query log Q and a logical session gap $sGap$, the vocabulary k-anonymity principle is observed if for $t \in T = Vocabulary(Q, sGap)$, there are at least $k-1$ other vocabularies in T that are indistinguishable from t. $\diamond$

B. Anonymization by Semantic Similarity Based Clustering

We partition $T = Vocabulary(Q, sGap)$ into a set of clusters by the semantic similarities among vocabularies such that each cluster contains at least k vocabularies. The centre of each cluster is used to derive the typical vocabulary to represent all the vocabularies in the cluster. In clustering vocabularies and selecting the typical vocabularies, we allow setting a semantic distance threshold ε to control the similarity degree that is required for vocabularies to comprise a cluster and for one typical query-term to represent others.

Running Example: Consider enforcing vocabulary 2-anonymity on the collection T of vocabularies in Figure 1 (b). To compare with generalization techniques, the taxonomy tree in Figure 2 (a) is always used to measure the semantic relationship between query-terms. We partition the five vocabularies into two clusters. The first cluster consists of the first three vocabularies, and the second cluster is made of the last two vocabularies. The typical vocabulary of each cluster is elected on a *majority vote* basis, that is {Wine:1, Vino:1, Jackets:1, Shoes:1} for the first cluster, and {Raw-Milk:2, Homo-Milk:1} for the second. The last column in Figure 2 (b) shows the anonymized vocabularies by our approach.

Our approach can retain more data utility than conventional approaches for two reasons.

First, instead of generalizing query-terms, we substitute query-terms by semantically similar, *original* query-terms. One may argue that generalizing Wine and Vino into Alcohol can keep more *truth* than distorting Wine into Vino. However, because of the extreme sparsity, Wine and Vino will be generalized into Edibles or even Goods, instead of Alcohol, which convey little information. We will have more discussion in Section V-A.

Second, the query-terms that do not match with the others in a cluster will be left out from the cluster, i.e., local suppression is an inherent component of our approach. Basically, the typical vocabulary of a cluster is elected on a majority vote basis. As an alternative, we can also determine the typical vocabulary of a cluster by locally generalizing semantically matched terms in the cluster, and hence our approach becomes the integration of local suppression and local generalization, which implies that we take advantage of the two techniques.

III. MEASURE SEMANTIC SIMILARITY

The key technique with our approach is how to measure the semantic similarities between vocabularies based on the similarities between query-terms.

A. Semantic Distance between Two Terms

The semantic similarity between two query-terms can be depicted by the semantic distance derived from a semantic network. The smaller the semantic distance is, the more similar the two query-terms are. One extreme case is that the distance is 0, meaning that the two are identical. If the distance is quite large, the two are irrelevant.

Definition 3 (Semantic distance between two terms): For two query-terms i and j, the semantic distance between i and j, denoted by $distance(i, j)$, is given externally. Given a user-defined semantic distance threshold ε, i and j are semantically *relevant* if $distance(i, j) \leq \varepsilon$, otherwise i and j are semantically *irrelevant*. $\diamond$

The following are semantic networks that can be used in computing distances between query-terms.

Online lexical dictionaries: For example, WordNet [5] is such a dictionary with which the specific meaning, called a sense, of a word is in a different set of synonyms, called a synset. Words and synsets are connected to one another through explicit semantic relations to form a network.

Given taxonomy tree: For example, the taxonomy tree in Figure 2 (a) is such a simplified network, with which distance(Wine, Vino) = 2, distance(Wine, Jackets) = 6, and so forth.

B. Distance between Two Vocabularies

It is an open problem to measure the semantic similarity (distance) between two vocabularies as vocabularies have no fixed structure. Metrics in the literature, such as Jaccard coefficient, Hamming distance, and Cosine similarity, could be possible solutions. However, with those metrics, query-terms are treated independently in that only common query-terms shared by two vocabularies are counted toward the similarity of the two vocabularies. In other words, those metrics do not take into account the semantic relationships between query-terms.

We want to exploit such semantic relationships between query-terms. Our anonymization approach is to publish typical vocabularies, each of which is semantically similar to and hence represents some original vocabularies. It follows

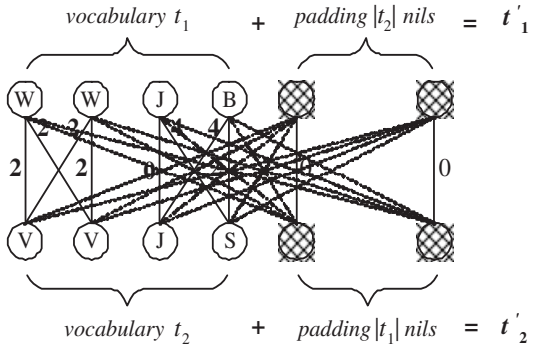


Figure 3. $distance(t_1, t_2)$ by a weighted bipartite matching

that the semantic similarity between two vocabularies should be determined based on a weighted bipartite matching, since a term in one vocabulary could be semantically similar to multiple terms in the other, with the weight between two terms being their distance as discussed in Section III-A.

Definition 4 (Semantic distance between two vocabularies): For two vocabularies $t_1, t_2 \in T$, the semantic distance between t_1 and t_2, denoted by $distance(t_1, t_2)$, is the minimum weight sum of the weighted bipartite matching between t_1 and t_2 where the weight of an edge across t_1 and t_2 is given by Definition 3. $\diamond$

The idea is to find such a matching between t_1 and t_2 that the sum of distances between matched query-terms is minimized. To be flexible, we allow two occurrences of a term in t_1 to be matched with two different terms in t_2. Thus, the duplicate occurrences of a term in a vocabulary t are handled as different terms hereafter. For the brevity of discussion, $i \in t$ merely denotes an occurrence of the term i in the vocabulary t, and $|t|$ denotes the sum of all occurrences.

There are two slightly different bipartite models. The first model is the minimum weight maximum matching model with the bipartite constructed as follows: t_1 and t_2 make the two parts; for a term i in t_1 and a term j in t_2, if i and j are semantically relevant, there is an edge connecting i and j with a weight equal to $distance(i, j)$.

For example, the left half of Figure 3 shows the bipartite made of the first two vocabularies in Figure 1 (b) with the user-given ε being 4. As $distance$(Wine, Vino) $= 2 \leq \varepsilon$, there is an edge connecting Wine and Vino with a weight 2. As $distance$(Wine, Jackets) $= 6 > \varepsilon$, there is no edge connecting Wine and Jackets, and so on. The distance between the two vocabularies is 6 by the first model.

In general, not all query-terms in the two vocabularies will be matched. The unmatched terms contribute to the *dissimilarity* between the two vocabularies. It is a reasonable treatment to charge each unmatched term with a uniform cost ω. We can think of ω as the distance between an unmatched query-term and a special nil term.

The second model is the minimum weight *perfect* matching model with an extended bipartite derived by padding some special nil terms into the basic bipartite. As shown in Figure 3, the first part t'_1 is t_1 padded with $|t_2|$ special nil terms, and the second part t'_2 is t_2 padded with $|t_1|$ special nil terms. A special nil term in t'_1 is connected to every term in t_2 with a weight ω, and to a special nil term in t'_2 with a weight 0. So does a special nil term in t'_2.

While the first model tends to match as many query-terms as possible, the second model tends to get the weight sum as small as possible. When $\omega >> \varepsilon$, the values of $distance(t_1, t_2)$ computed by the two models are the same.

Algorithm 1 SSC (T, k)

Input: the set T of vocabularies, the privacy requirement k
Output: the collection CL of clusters of vocabularies
1: **while** $|T| \geq k$ **do**
2: a new cluster $C \leftarrow$ the first t in T
3: **for** $l = 2$ to k **do**
4: find r in T nearest to $centre(C)$
5: move r from T to C
6: update $centre(C)$ with r
7: add C into CL
8: **for** each remaining t in T **do**
9: find C in CL with $centre(C)$ nearest to t
10: move t from T to C and update $centre(C)$
11: **return** CL

Figure 4. The pseudo code of the SSC algorithm

IV. A GREEDY ALGORITHM

We partition vocabularies into a set of clusters such that each cluster contains at least k vocabularies that are semantically similar. We present a greedy algorithm as such a constrained clustering problem is NP-hard [2].

A. Algorithm SSC

Our greedy algorithm, *SSC*, standing for *Semantic Similarity* based *Clustering*, creates one cluster at a time and makes the sum of semantic distances between vocabularies in the cluster as small as possible, as shown in Figure 4 where T is the set of vocabularies, C is the cluster currently being created, and CL is the collection of final clusters.

SSC works in 2 phases. In the first phase (line 1 to 7), SSC creates one cluster at a time when there are at least k vocabularies in T. SSC picks the first vocabulary t in T to initialize a new cluster C by moving t from T to C (line 2). SSC adds $k-1$ nearest vocabularies into C (line 3 to 6). The clusters are placed in CL (line 7). In the second phase (line 8 to 11), SSC assigns each remaining vocabulary t in T to an existing cluster nearest to t.

For the running example, in the first phase, SSC creates the first cluster by initializing it with the first vocabulary in Figure 1 (b), i.e., t_1 = {Wine, Wine, Jackets, Boots}. Then, SSC finds the nearest vocabulary, i.e., t_2 = {Vino, Vino,

Jackets, Shoes}. SSC continues to build the second cluster, which groups the last two vocabularies together. At the end of the first phase, the vocabulary remaining in T is the third, i.e., t_3 = {Wine, Vino, Raw-milk, Jackets, Shoes}, which is assigned to the first cluster in the second phase.

B. Computing the Centre of a Cluster

Computing $centre(C)$ for a cluster C is a core component of SSC, which serves two purposes. First, $centre(C)$ is used to select the nearest vocabulary to join C (line 4 of SSC in Figure 4). Second, $centre(C)$ is used to derive the typical vocabulary to represent C.

Computing centre(C) by a heuristic, weighted m-partite matching. Ideally, $centre(C)$ should be based on a weighted m-partite matching given the size of C is m, which however is NP-hard for $m \geq k \geq 3$. Therefore, we propose a heuristic solution that is to run a weighted bipartite matching in m iterations. Let gt_l be the result obtained in the l-th iteration. We treat gt_l as a hyper vocabulary that is a collection of size-l bags. Each bag contains l matched query-terms from l different vocabularies, which we call a hyper query-term.

In iteration $l+1$, the hyper vocabulary gt_l is matched with a vocabulary t in C that is unmatched yet. A query-term i in t is semantically relevant to a hyper query-term (a bag) in gt_l if i is relevant to every term in the bag. The distance between i and a bag is the sum of the distance between i and every term in the bag. If a query-term in t is matched with a hyper query-term in gt_l, the former will be put into the latter to make a new hyper query-term in gt_{l+1}. The bags in gt_l and the query-terms in t that are not matched will be suppressed.

Deriving the typical vocabulary from centre(C). Given the heuristic solution of $centre(C)$, i.e., gt_m computed in the above, we have much flexibility in deriving the typical vocabulary for C. The followings are a few methods. The first method is to compute on a majority vote basis. We first select the typical term for each bag of matched query-terms in gt_m, i.e., select the term that are closest to other terms in the bag in the sense of weighted semantic distances. We then piece together all the typical terms to get the typical vocabulary. The second method is to directly publish gt_m as the typical vocabulary. The third method is to locally generalize each bag in gt_m.

For the running example, the centre and typical vocabulary of the first cluster are computed as follows. First, t_1 is picked to make gt_1 = {{Wine}, {Wine}, {Jackets}, {Boots}}. Then, t_2 joins and is matched with gt_1, which results in gt_2 = {{Wine, Vino}, {Wine, Vino}, {Jackets, Jackets}, {Boots, Shoes}}. Finally, t_3 joins and is matched with gt_2, which yields gt_3 = {{Wine, Vino, Wine}, {Wine, Vino, Vino}, {Jackets, Jackets, Jackets}, {Boots, Shoes, Shoes}}. With the first method, {Wine, Vino, Jackets, Shoes} is elected as the typical vocabulary.

V. DATA UTILITY METRICS

We adapt the general loss metric [10] to measure the information loss in anonymizing bag-valued data, and present two indicators for measuring utility of anonymized data in frequent pattern mining.

A. Bag-valued Variant of LG and Information Loss Metric

For a fair comparison, we first extend the set-valued generalization algorithm LG [8] to the bag model.

Bag-valued Variant of LG [8]: Simply put, we can extend LG [8] to a bag-valued variant, LG(bag), by *keeping* duplicate (generalized) query-terms as long as they do not destroy the grouping. For the running example with T in Figure 1 (b) and the taxonomy tree in Figure 2 (a), the vocabularies anonymized by LG(bag) are shown in the third column in Figure 2 (b), while those by the original LG [8] are shown in the second column.

Loss Metric for the Bag-valued Vocabulary: LM [10] was proposed to quantify information loss in generalizing relational data. An adaptation of LM [10] for the bag-valued vocabularies, abbreviated as bLM, could be as follows.

First, SSC does not generalize query-terms while LG(bag) does, so the published results of the two approaches cannot be *directly* compared in LM [10]. However, we can use LM [10] to measure the bag of matched query-terms represented by each published term with our approach, e.g., {Wine:2, Vino:1} represented by Wine in the first cluster, and gauge the bag of query-terms represented by each generalized term with LG(bag), e.g., {Wine:2, Vino:1, Raw-milk:2, Homo-milk:1, Jackets:1, Boots:1} represented by Goods.

Second, each occurrence of a query-term that is not published should be considered as suppressed, whose information loss should be no less than generalizing it ($\geq 100\%$).

B. Data Utility in Frequent Pattern Mining

The data utility in frequent pattern mining is measured by $Recall$ and $sDist$ (semantic distance) of the top-n frequent closed patterns [7] in the original data, denoted by $OFCPs$, with regard to those in the anonymized data, denoted by $AFCPs$, as follows.

$$Recall = \frac{\sum_{O \in OFCPs} \max_{A \in AFCPs} |O \cap A|}{\sum_{O \in OFCPs} |O|}.$$

For a frequent pattern $O \in OFCPs, \max |O \cap A|$ is its largest subset that is a frequent pattern in the anonymized data. $Recall$ reflects the degree to which the frequent patterns in the original data are retained. Similarly,

$$sDist = \frac{\sum_{O \in OFCPs} \min_{A \in AFCPs} distance(O, A)}{\sum_{O \in OFCPs} |O|}.$$

For a frequent pattern $O \in OFCPs, \min distance(O, A)$ is the shortest semantic distance of O from $AFCPs$. $sDist$ is an indicator of precision in that it reflects the semantic closeness between the patterns in the original data and those in the anonymized data.

VI. EXPERIMENTAL EVALUATION

Our algorithm SSC anonymizes vocabularies that are bag-valued data while there is no prior work on anonymizing bag-valued data. To evaluate SSC by comparative experiments, we extended the set-valued algorithm LG [8] to the bag-valued model and call this competitor LG(bag) as in Section V-A. We evaluated SSC by comparing with LG(bag) in anonymizing vocabularies extracted from the AOL query log [16] while [6][18][20][3][15] are not comparable as they cannot be adapted to anonymize bag-valued data. We employed WordNet [5] in measuring the semantic distance between query-terms and in creating the taxonomy tree that is used by LG(bag) for generalization. Extensive experiments demonstrated that SSC retains much more data utility than LG(bag) in terms of information loss and usefulness in frequent pattern mining in the anonymized vocabularies.

The detail can be found in the technical report [14].

VII. RELATED WORKS

Query log anonymization: [21] is the most similar one, which however considers a privacy notion different from ours. [1][9][12]consider publication scenarios different from ours. [1] proposed a cryptography approach. [9] enforces k^{δ}-$anonymity$ by clustering queries and users. [12] publishes a query-click graph following *differential privacy*.

Set-valued data anonymization: [8] anonymizes data by local generalization to follow k-anonymity. [20] uses global suppression in enforcing (k, h, p)-$coherence$. [18] employs global generalization for k^m-$anonymity$. [6] applies the band matrix technique for $privacy\ degree$. [15] integrates generalization and suppression to observe k^m-$anonymity$. [3] employs generalization and suppression to enforce ρ-uncertainty that safeguards against sensitive associations.

VIII. CONCLUSION

We proposed the vocabulary k-anonymity principle. A vocabulary is the bag of query-terms extracted from web search queries issued by a user within a period of time. Such bag-valued data is extremely sparse, which makes conventional techniques cannot retain enough utility in anonymizing the data. We proposed the semantic similarity based clustering to address the issues, which outperforms the existing approaches.

ACKNOWLEDGMENT

Ke Wang's work is supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. Junqiang Liu's work is supported in part by the Science and Technology Development Plan of Zhejiang Province, China (2006C21034), and in part by the Natural Science Foundation of Zhejiang Province, China (Y105700).

REFERENCES

[1] E. Adar. User 4XXXXX9: Anonymizing query logs. Query Log Analysis Workshop, In WWW, 2007.

[2] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In PODS, 2006.

[3] J. Cao, P. Karras, C. Raissi, K. Tan. ρ-uncertainty: Inference-proof transaction anonymization. In VLDB, 2010.

[4] A. Cooper. A survey of query log privacy-enhancing techniques from a policy perspective. ACM Trans. on the Web, Vol. 2 , Issue 4 (2008).

[5] C. Fellbaum. WordNet, an electronic lexical database. MIT Press, Cambridge MA, 1998.

[6] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data. In ICDE, 2008.

[7] J. Han, J. Wang, Y. Lu, P. Tzvetkov. Mining top-k frequent closed patterns without minimum support. In ICDM, 2002.

[8] Y. He and J. Naughton. Anonymization of set-valued data via top-down, local generalization. In VLDB, 2009.

[9] Y. Hong, X. He, J. Vaidya, N. Adam, and V. Atluri. Effective anonymization of query logs. In CIKM, 2009.

[10] V. Iyengar. Transforming data to satisfy privacy constraints. In KDD, pages 279-288, 2002.

[11] R. Jones, R. Kumar, B. Pang, A. Tomkins. I know what you did last summer. In Query Logs User Privacy, CIKM, 2007.

[12] A. Korolova, K. Kenthapadi, N. Mishra, A. Ntoulas. Releasing search queries and clicks privately. In WWW, 2009.

[13] R. Kumar, J. Novak, B. Pang, A. Tomkins. On anonymizing query logs via token-based hashing. In WWW, 2007.

[14] J. Liu, K. Wang. Enforcing vocabulary k-anonymity by semanitc similarity based clustering (Technical Report). School of Computing Science, Simon Fraser University, 2010.

[15] J. Liu, K. Wang. Anonymizing transaction data by integrating suppression and generalization. In PAKDD, 2010.

[16] G. Pass, A. Chowdhury, C. Torgeson. A picture of search. In the 1st Intl. Conf. on Scalable Information Systems, Hong Kong, June, 2006.

[17] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In PODS, 1998.

[18] M. Terrovitis, N. Mamoulis, P. Kalnis. Privacy preserving anonymization of set valued data. In VLDB, 2008.

[19] L. Xiong and E. Agichtein. Towards privacy-preserving query log publishing. In Query Logs Workshop at WWW, 2007.

[20] Y. Xu, K. Wang, A. Fu, and P.S. Yu. Anonymizing transaction databases for publication. In KDD, 2008.

[21] Y. Zhu, L. Xiong, C. Verdery. Anonymizing user profiles for personalized web search. In WWW, 2010.