Constructing plausible innocuous pseudo queries to protect user query intention

Article *in* Information Sciences · July 2015

DOI: 10.1016/j.ins.2015.07.010

Information Sciences 325 (2015) 215–226

Contents lists available at ScienceDirect

Information Sciences

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

Constructing plausible innocuous pseudo queries to protect user query intention

Zongda Wu [a,b], Jie Shi [c], Chenglang Lu [a], Enhong Chen [b], Guandong Xu [d], Guiling Li [e,f,*], Sihong Xie [g], Philip S. Yu [g]

[a] Oujiang College, Wenzhou University, Wenzhou, China
[b] School of Computer Science, University of Science and Technology of China, Hefei, China
[c] School of Information Systems, Singapore Management University, Singapore
[d] Faculty of Engineering and IT, University of Technology, Sydney, Australia
[e] State Key Laboratory of Biogeology and Environmental Geology, China University of Geosciences, Wuhan, China
[f] School of Computer Science, China University of Geosciences, Wuhan, China
[g] Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA

ARTICLE INFO

Article history:
Received 30 May 2014
Revised 15 May 2015
Accepted 4 July 2015
Available online 13 July 2015

Keywords:
Knowledge
Privacy model
User intention
Query protection

ABSTRACT

Users of web search engines are increasingly worried that their query activities may expose what topics they are interested in, and in turn, compromise their privacy. It would be desirable for a search engine to protect the true query intention for users without compromising the precision-recall performance. In this paper, we propose a client-based approach to address this problem. The basic idea is to issue plausible but innocuous pseudo queries together with a user query, so as to mask the user intention. First, we present a privacy model which formulates plausibility and innocuousness, and then the requirements which should be satisfied to ensure that the user intention is protected against a search engine effectively. Second, based on a semantic reference space derived from Wikipedia, we propose an approach to construct a group of pseudo queries that exhibit similar characteristic distribution as a given user query, but point to irrelevant topics, so as to meet the security requirements defined by the privacy model. Finally, we conduct extensive experimental evaluations to demonstrate the practicality and effectiveness of our approach.

1. Introduction

Web search engines such as Google, Yahoo! and Microsoft Bing are becoming increasingly important in people's daily activities. As pointed out in [1–3], while search engines enable users to retrieve information from the Internet intuitively and effectively, the queries issued by these users can potentially compromise their privacy, i.e., the queries themselves can lead to an undesirable disclosure of user activities and topics of interest, and even confidential personal or business profiles.

It has been pointed out in [2,4] that the problem of disclosing user query intentions cannot be solved by using an anonymization scheme (e.g., those in [5,6]) to process a query log. For example, in 2006, AOL released an anonymized query log of around hundreds of thousands of randomly selected users [1,7]. The log data had been anonymized by removing individual

* Corresponding author. Tel.: +86 5513601551.
E-mail addresses: zongda1983@163.com, wuzongda@ustc.edu.cn (Z. Wu), shijie1123@gmail.com (J. Shi), chenglang.lu@qq.com (C. Lu), enhc@ustc.edu.cn (E. Chen), guandong.xu@uts.edu.au (G. Xu), guiling@cug.edu.cn (G. Li), xiesihong1@gmail.com (S. Xie), psyu@uic.edu (P.S. Yu).

http://dx.doi.org/10.1016/j.ins.2015.07.010
0020-0255/

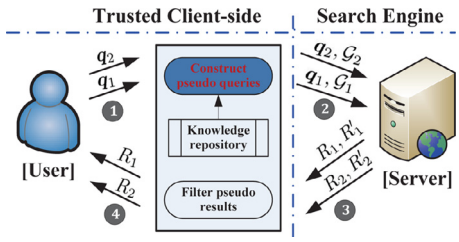

Fig. 1. The system model that we use, where the part "construct pseudo queries" is the key point.

identification information (e.g., IP address, username) associated with each user, while only keeping actual query text, timestamp, etc. However, such simple anonymization was proved ineffective, because user queries themselves still contained identification information [1]. It was shown that detailed user profiles (e.g., age, gender, location) could be constructed from the anonymized log data [8].

In addition, the problem also cannot be solved by other well-known solutions with regard to user privacy protection, such as cryptographic protocols [9–12], and Private Information Retrieval (PIR) [13,14]. As pointed out in [2,4], cryptographic protocols such as searchable encryption protocols [11] are not applicable to modern text search engines, because they cannot support similarity retrieval; moreover, PIR is also not practical, because it not only has high performance overheads, but also requires changes to existing search engines.

Recently, a system model is proposed to protect user privacy by masking the user query topics [1,2,15]. Its basic idea is to hide each user query among some pseudo queries, without any change to existing search engines. However, the system model seems to lack a practical implementation: for the approach proposed in [15] or [2], the generated pseudo queries are not meaningful, thus can be easily ruled out; for the approach proposed in [1], the results returned by the generated queries are not a superset of the genuine results, i.e., it is required to compromise the precision-recall performance (see Section 2 for detail).

In this work, we aim to prevent a search engine from identifying users' topics of interest (also called user intention) according to search terms, under the constraints of not compromising the precision-recall performance and not changing the search engine. To this end, we adopt the system model proposed in [1,2,15], that is, we attempt to mask the intention hidden in a user query by using well-designed pseudo queries. Fig. 1 presents the system model, which consists of an untrusted search engine and a number of clients (users). Each client accessing the search service trusts no one but himself/herself. As shown in Fig. 1, the pseudo queries $\mathcal{G}_i$ are constructed in a trusted client, and submitted together with the user query $\boldsymbol{q}_i$ to the search engine. Then, the search results R_i' that correspond to the pseudo queries $\mathcal{G}_i$ are discarded by the client, so only the search result R_i that corresponds to the user query $\boldsymbol{q}_i$ is returned to the user. It can be seen that the system model is transparent to both the search engine and the client, i.e., it requires no change to existing search engines; moreover, the result returned from the search engine is certainly a superset of the genuine result of a user query, thus it requires no compromise to the precision-recall performance.

However, it can also be seen that the quality of the pseudo queries generated by the client is very important in the system model, e.g., randomly constructed pseudo queries are often easy to be detected by the untrusted search engine, thus failing to hide the user query intention. To this end, given any user query, we aim to construct a group of pseudo queries that satisfy the following two requirements:

- **Plausibility**, i.e., the pseudo queries should exhibit similar characteristic distribution as the genuine user query. A user query is likely to include characteristic terms, e.g., synonymy, polysemy and high-specificity, thus, making it easy to be detected. For example, given two queries "X86 SSE4" and "puma cougar", where the first contains two terms of high-specificity, and the other contains two synonymous terms, such a characteristic distribution makes them unlikely to be randomly generated, so they are probably genuine.
- **Innocuousness**, i.e., the topics of the pseudo queries should be semantically-irrelevant to those of the user query, so that they are innocuous to the genuine user intention. For example, given a user query "Nike sneaker", an ideal pseudo query could be "Intel processor", because it has characteristics similar to the user query (plausible), but points to other irrelevant topics (innocuous).

The above requirements entail the following three challenges: (1) identifying the true intention for a user query; (2) capturing key characteristics inherent in the user query; and (3) constructing pseudo queries that have similar characteristics to the user query but point to innocuous topics.

It should be pointed out that in this work we mainly focus on protecting against a search engine. A search engine is deemed to be the most powerful potential adversary, because it possesses the most information, e.g., it hosts the plaintext corpus and executes the query processing algorithms [2]. However, we exclude tampering concerns posed by active adversaries, which have been addressed extensively in the context of query result authentication [16]. This work is also orthogonal to the privacy of user identity, which may be mitigated by query log anonymization, or by letting users connect to the search engine using an anonymous network [6].

In this paper, we propose an effective approach to protect user query intentions. It uses Wikipedia as an intermediate reference space to construct plausible but innocuous pseudo queries for a given user query. Then, these pseudo queries are issued to a search engine together with the user query, so as to protect the true user intention. Specifically, the main contributions of this paper are threefold. First, we present a privacy model to formulate plausibility and innocuousness, and then the requirements which should be satisfied to ensure that the user intention is protected against a search engine effectively. Second, based on a semantic reference space derived from Wikipedia, we propose an approach to determine the user intention, search for semantically-innocuous topics, and capture the main characteristic distributions behind a user query, consequently, obtaining a group of pseudo queries that satisfy the security requirements defined by the privacy model. Third, we conduct extensive experiments to evaluate the effectiveness of pseudo queries constructed by our approach with respect to plausibility and innocuousness.

The rest of this paper is organized as follows. Section 2 surveys related work. Section 3 formulates a privacy model for user query intention protection, and then presents an approach to well meet the privacy model. Section 4 presents and analyzes the experimental results. Finally, we conclude this paper in Section 5.

2. Related work

In the area of text search, query privacy protection has been studied extensively. A potential perfect solution to the problem is to use Private Information Retrieval (PIR) [13,14]. However, as pointed out in [1,2], the high computational complexity of PIR and the inability of a search engine to perform targeted advertising prevent its practical application. In addition, PIR requires changing existing search engine framework.

Cryptographic protocols such as symmetric key encryption with keyword search [9,10,12] and public key encryption with keyword search [11] can be extended to retrieve documents that exactly contain all the search terms of a user query (i.e., Boolean retrieval). In addition, privacy-preserving retrieval techniques have also been studied in the context of databases [17–19], which allow users to execute queries immediately over encrypted data. However, as pointed out in [2,20], neither of them is applicable to modern text search engines, which are designed to retrieve documents most similar to a given query (i.e., similarity retrieval).

It was suggested in [21,22] that user privacy may be protected by pushing the index and query processing of a search engine to a trusted third party, or by legally compelling the search engine to "forget" users' query activities right after they are served. However, as pointed out in [20], the risk of privacy disclosure remains when the trusted third party or search engine is infiltrated.

In [23], to safeguard a user query, the query was expressed as a vector of encrypted term weights in a server. Then, based on the query vector as well as all the document vectors on the server, the server produced a list of encrypted scores for the user. As this procedure has to be carried out on every document, it is too expensive for a search engine that needs to support large corpora.

In [20], the authors proposed to project the documents and queries from the term space into a synthetic factor space formed with Latent Semantic Indexing (LSI), to support privacy-preserving similarity retrieval. However, LSI is known to perform well only for small homogeneous corpora [24]. Therefore, it is not suitable for large document collections that span multiple subject domains.

TrackMeNot [15], as a browser extension, protects user queries from search engines by hiding them among randomly constructed "ghost" queries. The challenge in the mechanism, as the authors pointed out, is that the ghost queries can often be ruled out easily, because their term combinations are not meaningful.

In [1], the authors proposed to construct static groups of canonical queries, such that the queries in each group cover diverse topics. At runtime, a user query is substituted by the closest canonical query, while the other queries in the same group serve as cover queries to mask the user intention. However, one main problem of this approach is that substituting the user query with a canonical query degrades the precision–recall performance, as demonstrated by Murugesan and Clifton [1]. In [3], a similar approach was proposed for personalized search. Its basic idea is to substitute user queries with newly generated semantically-related ones, so as to protect user privacy while keeping the usefulness of personalized search.

In [25], a privacy-preserving solution was proposed to provide anonymity protection for the search terms of a user query and in turn the user intention. Its basic idea is to inject each user query with decoy terms pointing to alternative topics. The decoy terms are selected from a thesaurus to match the genuine terms in specificity and semantic association. In order to ensure usability, a retrieval protocol is provided, which enables the search engine to compute the correct document relevance scores from the genuine search terms, without interference from the decoy terms. A main drawback of the solution is that it requires changing search engines [2].

In [2], an approach was proposed to obfuscate the topics relevant to the user intention. The work introduced a privacy model, which allowed a user to stipulate what relevant topics should be obfuscated, and to what extent they should be obfuscated. Then, it presented an algorithm to achieve the privacy requirement by injecting ghost queries into each user query. However, the work only focuses on how to use ghost queries to obfuscate the topics relevant to a given user query, without taking into account the plausibility of ghost queries (e.g., the similarity of term specificity between ghost queries and genuine queries), thereby, making the ghost queries often easy to be ruled out according to characteristic search terms.

Table 1
Key notations.

Item	Meaning	Item	Meaning
$\mathcal{T}$	The term space	$\mathcal{T}(\boldsymbol{q})$	Terms contained in the query $\boldsymbol{q}$
$\mathcal{E}$	The concept space	$\mathcal{T}(p_i)$	Terms belonging to a topic p_i
$\mathcal{P}$	The topic space	$\mathcal{T}(e_j)$	Terms of titles of a concept e_j
$\mathcal{Q}$	The query space	$\mathcal{E}(\boldsymbol{q})$	Concepts relevant to the query $\boldsymbol{q}$
$\mathcal{P}(\boldsymbol{q})$	Topics relevant to the query $\boldsymbol{q}$	$\mathcal{E}(p_i)$	Concepts belonging to a topic p_i

3. Methodology

3.1. Privacy model

According to the system model given in Section 1, the quality of pseudo queries determines whether a user query can be protected from a search engine deducing the true user intention; however, a group of pseudo queries with good quality should be difficult to be distinguished from the genuine query, and semantically irrelevant to the genuine user intention. In this subsection, we define a privacy model under the system model. Specifically, we formulate what is the user intention, and the requirements which should be satisfied to protect the user intention against a search engine effectively. In Table 1, we summarize some key notations, which will be explained as they are used.

Definition 1. Let $\mathcal{Q}$ denote the query space, and $\mathcal{P}$ the topic space. Given any query $\boldsymbol{q} \in \mathcal{Q}$ and any topic $p_i \in \mathcal{P}$, a **query–topic relevance** function would return the measurement of semantic relevance between $\boldsymbol{q}$ and p_i, and it can be expressed as $Re(\boldsymbol{q}, p_i) : \mathcal{Q} \times \mathcal{P} \mapsto \mathbb{R}$.

It is obvious that a good query-topic function should be positively related to the genuine query-topic relevance, i.e., if $Re(\boldsymbol{q}, p_1) > Re(\boldsymbol{q}, p_2)$, then the query $\boldsymbol{q}$ should be more semantically relevant to the topic p_1 than to p_2. In Definition 1, however, we do not attempt to present a specific objective function for measuring the semantic relevance between a query and a topic. This work will be done in the next subsection. Based on Definition 1, we formulate the user intention for a query below.

Definition 2. The **user intention** pertaining to a query $\boldsymbol{q}$ comprises the topics that are relevant to $\boldsymbol{q}$, i.e., $\mathcal{P}(\boldsymbol{q}) = \{p_i \mid p_i \in \mathcal{P} \wedge Re(\boldsymbol{q}, p_i) > \lambda_u\}$, where the threshold λ_u is used to remove topics that are less relevant to the query $\boldsymbol{q}$.

Definition 3. Given any two topics p_1 and p_2 (where $p_1, p_2 \in \mathcal{P}$), a **topic–topic relevance** function would return the measurement of semantic relevance between p_1 and p_2, and it can be expressed as $Re(p_1, p_2) : \mathcal{P} \times \mathcal{P} \mapsto \mathbb{R}$.

Here, it is also required that if $Re(p_1, p_0) > Re(p_2, p_0)$, then the topic p_0 should be more semantically-relevant to the topic p_1 than to p_2. Based on Definitions 1–3, we formulate the innocuousness of any topic to a user query below.

Definition 4. Given any topic $p_i \in \mathcal{P}$, its innocuousness to a query $\boldsymbol{q} \in \mathcal{Q}$ is reversely correlated with its relevance to each topic relevant to $\boldsymbol{q}$ (i.e., each topic in $\mathcal{P}(\boldsymbol{q})$), so the **innocuousness** of the topic p_i to the query $\boldsymbol{q}$ can be defined as

$$Inn(p_i, \boldsymbol{q}) = \left(\sum\nolimits_{p_k \in \mathcal{P}(\boldsymbol{q})} Re(\boldsymbol{q}, p_k) \cdot Re(p_k, p_i) \right)^{-1}. \tag{1}$$

From Definition 4, it can be concluded that the greater the innocuousness of a topic to a user query, the more the topic should be semantically-irrelevant to the user query intention.

Definition 5. Let $\mathcal{T}$ denote the term space. Given any term $t_j \in \mathcal{T}$, a **characteristic function** is defined as $F(t_j) : \mathcal{T} \mapsto \mathbb{R}$, which returns a characteristic value for the term t_j. Let $\mathcal{T}(\boldsymbol{q})$ denote a set of terms contained in a query $\boldsymbol{q}$ (it is obvious that $\mathcal{T}(\boldsymbol{q}) \subseteq \mathcal{T}$). Given any query $\boldsymbol{q} \in \mathcal{Q}$, its **characteristic distribution** can be described by a vector:

$$\mathcal{F}(\boldsymbol{q}) = \left[F(t_1), F(t_2), \ldots, F(t_n) \right], \text{where } n = |\mathcal{T}(\boldsymbol{q})| \text{ and } t_1, t_2, \ldots, t_n \in \mathcal{T}(\boldsymbol{q}). \tag{2}$$

A search term may contain a number of characteristics, so we can establish multiple characteristic functions, e.g., a specificity function to measure the term specificity, or a polysemy function to judge whether a search term is polysemous. Thus, a query may also have multiple characteristic distributions. In this paper, we will consider three types of characteristics (i.e., synonymy, polysemy and high-specificity) inherent in search terms (see the next subsection for detail). Based on Definition 5, we formulate the characteristic plausibility between any two queries below.

Definition 6. Suppose that for any given query $\boldsymbol{q} \in \mathcal{Q}$, we would establish n characteristic distribution vectors: $\mathcal{F}_1(\boldsymbol{q})$, $\mathcal{F}_2(\boldsymbol{q}), \ldots, \mathcal{F}_n(\boldsymbol{q})$. Given any two queries $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ ($\boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathcal{Q}$), the **plausibility** between $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ can be measured by the similarity between their characteristic distribution vectors (where *Dist* denotes the Euler distance between two vectors):

$$Pla(\boldsymbol{q}_1, \boldsymbol{q}_2) = \left(\prod\nolimits_{k=1}^{n} \left(Dist(\mathcal{F}_k(\boldsymbol{q}_1), \mathcal{F}_k(\boldsymbol{q}_2)) + 1 \right) \right)^{-1}. \tag{3}$$

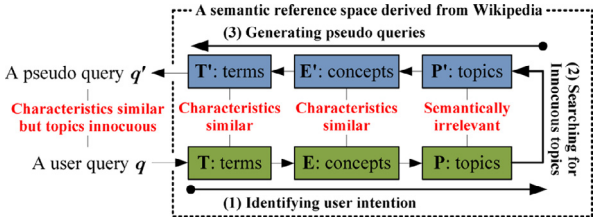

Fig. 2. The workflow of our approach, where we first map the user query $\boldsymbol{q}$ to related terms, concepts and topics; then, by using these entities as an intermediate reference, we search for innocuous topics and then pseudo terms that have similar characteristics to the genuine terms; and finally, we form the pseudo query $\boldsymbol{q}'$.

Now, based on Definitions 4 and 6, we formulate the requirements that have to be satisfied by a group of pseudo queries so as to protect the user intention hidden in a user query against a search engine effectively.

Definition 7. Let $\boldsymbol{q} \in \mathcal{Q}$ be a user query, and $\mathcal{G} \subseteq \mathcal{Q}$ a group of pseudo queries that are constructed specifically for $\boldsymbol{q}$. Then, it is deemed that the user intention behind the query $\boldsymbol{q}$ has been protected effectively by $\mathcal{G}$ if $\mathcal{G}$ satisfies the following two requirements.

- **Innocuousness**: Each topic p'_i relevant with any pseudo query $\boldsymbol{q}' \in \mathcal{G}$ should be innocuous to the user query $\boldsymbol{q}$, i.e., $(\forall \boldsymbol{q}' \in \mathcal{G}, \forall p'_i \in \mathcal{P}(\boldsymbol{q}') \rightarrow Inn(p'_i, \boldsymbol{q}) > \lambda_i)$. This condition ensures that the topics behind pseudo queries are semantically-irrelevant to the user query, such that the genuine user intention can be obfuscated by the pseudo queries.
- **Plausibility**: Each pseudo query $\boldsymbol{q}' \in \mathcal{G}$ should exhibit similar characteristics as the user query $\boldsymbol{q}$, i.e., $(\forall \boldsymbol{q}' \in \mathcal{G} \rightarrow Pla(\boldsymbol{q}', \boldsymbol{q}) > \lambda_a)$. This requirement ensures that when the user query is issued together with the pseudo queries, the user query is hard to be detected according to characteristic terms, such that the user query can be effectively hidden.

Above, we present Definitions 1–7 to formulate the privacy model. For the query-topic relevance (Definition 1), topic-topic relevance (Definition 3) and characteristic distribution (Definition 5), there is still a lack of specific objective functions. However, the remaining definitions (especially for Definition 7) are proposed based on the threes. Therefore, it is concluded that how to implement these objective functions as accurate as possible is the key to achieve the privacy model so as to protect user query intentions effectively.

3.2. Proposed approach

According to the privacy model presented in Section 3.1, in this subsection, we propose our approach, i.e., how the semantic knowledge derived from Wikipedia is used as an intermediate reference space to construct a group of plausible innocuous pseudo queries for a given user query, so as to meet the requirements presented in the privacy model.

Wikipedia is one of the world's largest human knowledge repositories, which has very broad knowledge coverage about different terms, due to the contributions by volunteers around the world. It mainly consists of concepts, categories, and various connections within concepts, within categories, or between concepts and categories. It uses an article to describe a single concept, where the article title represents one term corresponding to the concept, and uses redirect links to group synonymous terms to the same concept. Moreover, it contains a hierarchical categorization system, where each concept or category belongs to at least a parent category. All these features enable it to be exploited as a knowledge ontology for understanding query intention and capturing query characteristics [26] (see [27,28] for more detail on the structure of Wikipedia).

In the work, we use concept titles to denote search terms, and categories of higher generality (i.e., lower specificity) to denote topics. As a result, we can map each query into the reference space from Wikipedia. Fig. 2 shows the processing flow of how Wikipedia is used to construct a pseudo query for a user query. First, based on a variety of connections within concepts and categories, we identify the user intention and term characteristics from the user query. Second, from some topics that are innocuous to the user query, we search for a plausible pseudo query which exhibits similar characteristic distribution as the user query. Finally, we generate a pseudo query that meets the requirements of Definition 7.

In the following, we will describe the key steps shown in Fig. 2 in detail, i.e., identifying the user intention, searching for innocuous topics and generating pseudo queries. Moreover, the pre-process for the Wikipedia knowledge (e.g., the extraction of terms, concepts and categories) will be introduced in Section 4.1, so in this section we assume that the knowledge relevant to Wikipedia has been obtained in advance.

3.2.1. Identifying user intention

This step aims to identify the intention behind a given user query $\boldsymbol{q}$. As shown in Fig. 2, the basic idea is to leverage the concept space derived from Wikipedia as an intermediate reference to map the query $\boldsymbol{q}$ from the term space $\mathcal{T}$ into a group $\mathcal{P}(\boldsymbol{q})$ of relevant topics in the topic space $\mathcal{P}$. In the work, topics are considered as a special kind of categories of Wikipedia, which are generally located at higher levels of the hierarchical categorization system of Wikipedia, and assigned by the approach in advance. Table 2 in Section 4.1 presents our chosen topics in the experiments.

Table 2
All the topics in the topic space used in our experiments.

Agriculture	Arts	Animals	Belief	Business	Chronology
Culture	Education	Environment	Engineering	Economics	Food
Geography	Health	History	Humanities	Language	Law
Life	Mathematics	Medicine	Nature	People	Politics
Science	Society	Sports	Traditions	Technology	Tools

Let $\mathcal{E}$ denote the concept space, which consists of all the concepts in Wikipedia. Let $Re(e_j, p_i)$ denote the relevance between a concept $e_j \in \mathcal{E}$ and a topic $p_i \in \mathcal{P}$, and $Re(\boldsymbol{q}, e_j)$ the relevance between the query $\boldsymbol{q}$ and the concept e_j. Then, by using the concept space $\mathcal{E}$ as an intermediate reference, an objective function of query-topic relevance (refer to Definition 1) can be computed as

$$Re(\boldsymbol{q}, p_i) = \sum\nolimits_{e_j \in \mathcal{E}} Re(\boldsymbol{q}, e_j) \cdot Re(e_j, p_i). \tag{4}$$

Let $\mathcal{T}(e_j)$ denote a set of all the titles of any concept $e_j \in \mathcal{E}$ (a concept may have several titles due to synonymous terms etc.). Let $\mathcal{T}(\boldsymbol{q})$ denote a set of terms contained in the query $\boldsymbol{q}$, which can be obtained efficiently by using a similar method mentioned in [29] to map the query $\boldsymbol{q}$ into a set of concept titles (terms) in Wikipedia. Then, the relevance $Re(\boldsymbol{q}, e_j)$ between the concept e_j and the query $\boldsymbol{q}$ is defined as the number of the titles of the concept e_j that appear in the query $\boldsymbol{q}$, i.e.,

$$Re(\boldsymbol{q}, e_j) = |\mathcal{T}(\boldsymbol{q}) \cap \mathcal{T}(e_j)|. \tag{5}$$

Let $\mathcal{E}(p_i)$ denote a set of concepts which belong to a topic p_i, namely, each concept in $\mathcal{E}(p_i)$ is reachable to the topic p_i along the categorization system of Wikipedia. Let $depth(e_j, p_i)$ denote the length of the shortest path from a concept e_j to a topic p_i in the categorization system of Wikipedia. Then, the relevance $Re(e_j, p_i)$ between a concept $e_j \in \mathcal{E}$ and a topic $p_i \in \mathcal{P}$ can be described by the following three cases. (1) If the concept e_j is not contained in $\mathcal{E}(p_i)$, then it should be much less relevant to the topic p_i, and thus we simply set $Re(e_j, p_i) = 0$. (2) If the concept e_j is unambiguous (i.e., its meaning is unambiguous, e.g., 'osteosarcoma') and $e_j \in \mathcal{E}(p_i)$, then its relevance to the topic p_i is defined as

$$Re(e_j, p_i) = \left(\log_2 (depth(e_j, p_i)) + 1\right)^{-1}. \tag{6}$$

Eq. (6) shows that the relevance of a concept to its topic is reversely correlated with the depth of the concept in the topic. However, in the concept space $\mathcal{E}$, there is another special kind of ambiguous concepts caused by polysemous terms (e.g., 'cell', 'puma', i.e., whose meanings are ambiguous). To deal with them, disambiguation pages in Wikipedia are used, where various possible unambiguous concepts are presented for polysemous terms, e.g., for the ambiguous concept 'puma', its disambiguation page lists dozens of possible unambiguous concepts[1]. (3) If the concept e_j is ambiguous and $e_j \in \mathcal{E}(p_i)$, let $\mathcal{E}(e_j)$ denote a set of possible concepts that correspond to the ambiguous concept e_j, then the relevance of e_j to the topic p_i is defined as

$$Re(e_j, p_i) = \max\nolimits_{e_k \in \mathcal{E}(e_j)} \left\{Re(e_k, p_i)\right\}. \tag{7}$$

Now, after putting Eqs. (5)–(7) into Eq. (4), for the query $\boldsymbol{q}$, we determine an objective function of query-topic relevance, and hence a set $\mathcal{P}(\boldsymbol{q})$ of relevant topics (i.e., the user intention) according to Definition 2.

However, it is very likely that one concept (especially for an ambiguous concept) belongs to more than one topic, such that many topics in $\mathcal{P}(\boldsymbol{q})$ are not actually relevant to the query $\boldsymbol{q}$. For example, given a query "cougar puma", the polysemous term 'puma' corresponds to tens of concepts, so it is relevant to many topics. However, we can observe that only the topic 'Animals' is certainly relevant to the given query, while all the others are less relevant or irrelevant, and need to be removed. Below, we tackle such a situation.

Observation 1. For any concept e_j relevant to the query $\boldsymbol{q}$ (i.e., $Re(\boldsymbol{q}, e_j) \neq 0$), and two topics p_1 and p_2 relevant to the concept e_j (i.e. $e_j \in \mathcal{E}(p_1) \wedge e_j \in \mathcal{E}(p_2)$), if $Re(\boldsymbol{q}, p_1) > Re(\boldsymbol{q}, p_2)$, then for the query $\boldsymbol{q}$, it is more likely that the concept e_j is used to describe the topic p_1 than the topic p_2.

Rationale: If $Re(\boldsymbol{q}, p_1) > Re(\boldsymbol{q}, p_2)$, then it shows that p_1 is more relevant than p_2 to the query $\boldsymbol{q}$, i.e., compared with p_2, p_1 is more likely the topic that the user wants to query. It is also more likely that the user wants to describe the topic p_1 using the search terms corresponding to the concept e_j.

Let $\mathcal{E}(\boldsymbol{q})$ denote a group of concepts that are relevant to the user query $\boldsymbol{q}$, i.e., $\mathcal{E}(\boldsymbol{q}) = \{e_j \mid e_j \in \mathcal{E} \wedge Re(\boldsymbol{q}, e_j) \neq 0\}$. Based on Observation 1, for each concept $e_j \in \mathcal{E}(\boldsymbol{q})$, we determine the only topic that the concept e_j is most likely relevant to in the query $\boldsymbol{q}$, so as to remove unwanted topics. For example, given a query "cougar puma", although the term 'puma' is related to many topics, we only retain the topic 'Animals' due to its greatest relevance score to the query. Algorithm 1 describes this process, wherein, $\mathcal{P}(e_j)$ (on Line 7) denotes a set of topics that the concept e_j belongs to according to the category system of Wikipedia.

It can be observed that the time overhead of Algorithm 1 is mainly caused by the operation of mapping the user query $\boldsymbol{q}$ into terms in the term space (on Line 2), so the algorithm time complexity is $O(|\mathcal{T}(\boldsymbol{q})| \cdot \log_2 (|\mathcal{T}|))$, where $\mathcal{T}$ denotes the term space. Moreover, in Algorithm 3 of Section 3.2.3, we will use the function $topic(t_j, \boldsymbol{q})$ (on Line 6 of Algorithm 3) to return the topic that the term t_j is most likely relevant to in the query $\boldsymbol{q}$. Obviously, the function is built based on Algorithm 1.

[1] http://en.wikipedia.org/wiki/Puma .

Algorithm 1: Identifying the relevant topics for a user query.

Input: $\boldsymbol{q}$, a search query issued by a user of text search engine.

1 **begin**
2 $\mathcal{P}^* \leftarrow \left\{p_i \mid p_i \in \mathcal{P} \wedge Re(\boldsymbol{q}, p_i) > \lambda_u\right\}$; ;
3 $\mathcal{P}(\boldsymbol{q}) \leftarrow \emptyset$;
4 **while** $\mathcal{P}^* \neq \emptyset$ **do**
5 $p_0 \leftarrow$ the topic in $\mathcal{P}^*$ of the greatest relevance to the user query $\boldsymbol{q}$, that is, $\forall p_i \in \mathcal{P}^* \rightarrow Re(\boldsymbol{q}, p_0) \geq Re(\boldsymbol{q}, p_i)$;
6 **foreach** $e_j \in \mathcal{E}(p_0) \cap \mathcal{E}(\boldsymbol{q})$ **do**
7 **foreach** $p_i \in \mathcal{P}(e_j) - \{p_0\}$ **do**
8 $Re(\boldsymbol{q}, p_i) \leftarrow Re(\boldsymbol{q}, p_i) - Re(\boldsymbol{q}, e_j) \cdot Re(e_j, p_i)$;
9 **if** $Re(\boldsymbol{q}, p_i) = 0$ **then** $\mathcal{P}^* \leftarrow \mathcal{P}^* - \{p_i\}$; //remove unwanted ones.;
10 $\mathcal{P}^* \leftarrow \mathcal{P}^* - \{p_0\}$; $\mathcal{P}(\boldsymbol{q}) \leftarrow \mathcal{P}(\boldsymbol{q}) \cup \{p_0\}$; //remain the topic p_0.
11 **return** $\mathcal{P}(\boldsymbol{q})$; //**output**: the user intention (a set of topics relevant to the query $\boldsymbol{q}$).

3.2.2. Searching for innocuous topics

This step aims to search for topics as semantically irrelevant to the topics of the user query as possible, such that the pseudo terms selected from these irrelevant topics are innocuous to the user intention.

For any two topics p_1 and p_2 in the topic space ($p_1, p_2 \in \mathcal{P}$), the relevance between the two topics is positively correlated with the number of concepts that simultaneously belong to the two topics, so an objective function of topic-topic relevance (refer to Definition 3) can be defined as

$$Re(p_1, p_2) = \left(|\mathcal{E}(p_1) \cap \mathcal{E}(p_2)|\right) \cdot \left(|\mathcal{E}(p_1) \cup \mathcal{E}(p_2)|\right)^{-1}. \tag{8}$$

Above, $\mathcal{E}(p_i)$ denotes a set of concepts that belong to a topic p_i. Note that the relevance between any two topics in $\mathcal{P}$ would be calculated offline based on Wikipedia, so as to reduce running performance overhead.

Then, after putting Eq. (8) into Eq. (1) of Definition 4, the innocuousness of a topic $p_i \in \mathcal{P}$ to the query $\boldsymbol{q}$ can be determined explicitly. Now, our objective is to efficiently search for a set $\mathcal{P}'(\boldsymbol{q})$ of topics with the greatest innocuousness to the query $\boldsymbol{q}$ from $\mathcal{P} - \mathcal{P}(\boldsymbol{q})$. This search is described in Algorithm 2.

Algorithm 2: Searching for semantically-innocuous topics for a user query.

Input: $\mathcal{P}(\boldsymbol{q})$, the user intention of the query $\boldsymbol{q}$.

1 **begin**
2 $\mathcal{P}^* \leftarrow \mathcal{P} - \mathcal{P}(\boldsymbol{q})$; $\mathcal{P}'(\boldsymbol{q}) \leftarrow \emptyset$;
3 **while** $|\mathcal{P}'(\boldsymbol{q})| < |\mathcal{P}(\boldsymbol{q})|$ **do**
4 $p_0 \leftarrow$ the topic in $\mathcal{P}^*$ of the greatest innocuousness to the query $\boldsymbol{q}$, that is, $\forall p_i \in \mathcal{P}^* \rightarrow Inn(p_0, \boldsymbol{q}) \geq Inn(p_i, \boldsymbol{q})$;
5 $\mathcal{P}^* \leftarrow \mathcal{P}^* - \{p_0\}$;
6 $\mathcal{P}'(\boldsymbol{q}) \leftarrow \mathcal{P}'(\boldsymbol{q}) \cup \{p_0\}$; //obtain a pseudo topic.
7 **return** $\mathcal{P}'(\boldsymbol{q})$; //**output**: a set of topics semantically innocuous to the user query $\boldsymbol{q}$.

In Algorithm 2, we make the set $\mathcal{P}'(\boldsymbol{q})$ (i.e., the set of innocuous topics) with the same size as the set $\mathcal{P}(\boldsymbol{q})$ (i.e., the user intention). This is for ensuring the topic plausibility of pseudo queries constructed based on the set $\mathcal{P}'(\boldsymbol{q})$. Moreover, we sort each topic in $\mathcal{P}'(\boldsymbol{q})$ in descending order according to its innocuousness to the query $\boldsymbol{q}$, and sort each topic in $\mathcal{P}(\boldsymbol{q})$ in descending order according to its relevance to the query $\boldsymbol{q}$. Then, we pair p'_1 with p_1, p'_2 with p_2 and so on (where $p'_i \in \mathcal{P}'(\boldsymbol{q})$, $p_i \in \mathcal{P}(\boldsymbol{q})$, $\forall i \in 1, 2, \ldots, |\mathcal{P}(\boldsymbol{q})|$). In Algorithm 3 given in Section 3.2.3, we will use the function $pair(p_i, \boldsymbol{q})$ (on Line 6 of Algorithm 3) to return the paired innocuous topic p'_i for each topic $p_i \in \mathcal{P}(\boldsymbol{q})$. In addition, it can be seen that the time complexity of Algorithm 2 is $O(|\mathcal{P}(\boldsymbol{q})| \cdot |\mathcal{P}|)$, which is efficient, because of the small size of the topic space $\mathcal{P}$.

3.2.3. Generating plausible pseudo queries

A user query often includes some characteristic terms, which makes it easy to be distinguished. Therefore, this step aims to search the set $\mathcal{P}'(\boldsymbol{q})$ of innocuous topics for a group of pseudo terms that have highly similar characteristics to those in the genuine query $\boldsymbol{q}$, with the help of the thesaurus knowledge from Wikipedia, making that the pseudo queries constructed based on these pseudo terms are difficult to be distinguished from the genuine query $\boldsymbol{q}$.

In the work, we mainly take into account three types of characteristics of search terms, i.e., specificity, synonymy and polysemy, whose characteristic functions (refer to Definition 5) can all be captured in advance using the thesaurus knowledge derived from Wikipedia:

Algorithm 3: Generating a plausible innocuous pseudo query.

Input: $\mathcal{P}(\boldsymbol{q})$ from Algorithm 1, and $\mathcal{P}'(\boldsymbol{q})$ from Algorithm 2.

```
1  begin
2      foreach p'_i ∈ P'(q) do
3          divide T(p'_i) into three mutually disjoint categories: T_o(p'_i), T_y(p'_i) and T_n(p'_i), and then, for each category, sort
           terms according to their specificity values;
4      T'(q) ← ∅;
5      foreach t_j ∈ T(q) do
6          p'_i ← pair(topic(t_j, q), q); //topic and pair defined in Sections 3.2.1 and 3.2.2.
7          if F_o(t_j) = 1 then t'_j ← binarySearch(T_o(p'_i), t_j) ;
8          if F_y(t_j) = 1 then t'_j ← binarySearch(T_y(p'_i), t_j) ;
9          if F_o(t_j) = 0 ∧ F_y(t_j) = 0 then t'_j ← binarySearch(T_n(p'_i), t_j) ;
10         T'(q) ← T'(q) ∪ {t'_j};
11     based on T'(q), generate a pseudo query q';
12     return q'; //output: a pseudo query plausible to the user query q.
```

- $F_p(t_j)$: Given any term $t_j \in \mathcal{T}$, its **specificity** value is defined as the length of the shortest path from its corresponding concept e_j (i.e., e_j is named after the term t_j) to the root category in the categorization system of Wikipedia.
- $F_y(t_j)$: Given any term $t_j \in \mathcal{T}$, its **synonymy** value is defined as 1.0, if the number of the titles of its corresponding concept e_j is larger than one; otherwise its synonymy value is defined as 0.
- $F_o(t_j)$: Given any term $t_j \in \mathcal{T}$, its **polysemy** value is defined as 1.0, if its corresponding concept e_j is ambiguous; otherwise its polysemy value is defined as 0.

After putting the functions F_p, F_y and F_o into Eq. (2) of Definition 5, we obtain a group of characteristic distribution vectors for the query $\boldsymbol{q}$, i.e., $\mathcal{F}_p(\boldsymbol{q})$, $\mathcal{F}_y(\boldsymbol{q})$ and $\mathcal{F}_o(\boldsymbol{q})$. Then, we obtain the characteristic plausibility between any two queries (refer to Definition 6). Now, our objective is to search the set $\mathcal{P}'(\boldsymbol{q})$ of innocuous topics for a group of pseudo terms as efficiently as possible, such that the pseudo query constructed based on the pseudo terms have the greatest characteristic plausibility with the user query $\boldsymbol{q}$.

Let $\mathcal{T}(p_i)$ denote a set of terms whose corresponding concepts belong to the topic p_i (i.e., belong to $\mathcal{E}(p_i)$). Then, to obtain a solution close to the global optimum of plausibility for the query $\boldsymbol{q}$, the following heuristic search is adopted: (1) for each search term t_j contained in $\boldsymbol{q}$ (i.e., $t_j \in \mathcal{T}(\boldsymbol{q})$), we select a pseudo term from $\mathcal{P}'(\boldsymbol{q})$ (more specifically, from the term set $\{t'_j | p'_i \in \mathcal{P}'(\boldsymbol{q}) \wedge t'_j \in \mathcal{T}(p'_i)\}$) that has the closest characteristic values to the term t_j; and (2) based on these pseudo terms, we then form a pseudo query. However, it can be observed that the time complexity of generating a pseudo query based on such a search strategy is $O(|\mathcal{T}(\boldsymbol{q})| \cdot |\mathcal{T}|)$. Obviously, this is too time-consuming for online web search applications, due to the large size of the term space (the number of terms contained in Wikipedia is close to tens of millions). In fact, the time complexity can be reduced by advance sorting of all the terms in the term space according to their characteristic values.

Observation 2. Given any query $\boldsymbol{q} \in \mathcal{Q}$, its characteristic distribution vectors $\mathcal{F}_y(\boldsymbol{q})$ and $\mathcal{F}_o(\boldsymbol{q})$ are orthogonal to each other; formally, for any term $t_j \in \mathcal{T}(\boldsymbol{q})$, we have $F_y(t_j) \cdot F_o(t_j) = 0$.

Rationale: Based on the structure of Wikipedia, we know that each ambiguous concept has only one title (i.e., non-synonymous), so we have $\forall t_j \in \mathcal{T} \wedge F_o(t_j) = 1 \rightarrow F_y(t_j) = 0$; while each synonymy concept is unambiguous (i.e., non-polysemous), so we have $\forall t_j \in \mathcal{T} \wedge F_y(t_j) = 1 \rightarrow F_o(t_j) = 0$. As a result, we have $\forall t_j \in \mathcal{T}(\boldsymbol{q}) \rightarrow F_y(t_j) \cdot F_o(t_j) = 0$.

Based on Observation 2, we know that in the term space $\mathcal{T}$, there is no term whose synonymy value and polysemy value are both equal to 1.0, so we can divide the term space $\mathcal{T}$ into three mutually disjoint categories: (1) $\mathcal{T}_y$, consisting of synonymy terms, (2) $\mathcal{T}_o$, consisting of polysemy terms, and (3) $\mathcal{T}_n$, consisting of all the other terms. Then, we in advance sort the terms for each category according to the term specificity values. Lastly, for each search term t_j contained in the query $\boldsymbol{q}$, we search the category that corresponds to t_j, so as to obtain the pseudo term with the closest specificity value to t_j. Algorithm 3 formulates the heuristic search, where the function **binarySearch** returns the pseudo term that has the closest specificity value to t_j. In Algorithm 3, Lines 2 and 3 are offline executed in advance. As a result, the time complexity of Algorithm 3 is significantly reduced to $O(|\mathcal{T}(\boldsymbol{q})| \cdot \log_2(|\mathcal{T}|))$.

Now, with the help of Algorithms 1–3, we present an example to illustrate the working process of our approach. Below, we use the categories presented in Table 2 as the topic space $\mathcal{P}$, and use "cougar puma" as the user query $\boldsymbol{q}$. First, Algorithm 1 maps the search terms of $\boldsymbol{q}$ into a set $\mathcal{P}(\boldsymbol{q})$ of relevant topics, and then, from $\mathcal{P}(\boldsymbol{q})$, finds out the topic 'Animals' that has the greatest relevance to $\boldsymbol{q}$, thereby, outputting the user intention behind $\boldsymbol{q}$, i.e., $\mathcal{P}(\boldsymbol{q})$={'Animals'}. Second, Algorithm 2 searches the topic space $\mathcal{P}$, and as a result obtains the topic 'Tools' that has the greatest innocuousness to $\boldsymbol{q}$, i.e., outputting $\mathcal{P}'(\boldsymbol{q})$={'Tools'}. Finally, based on the characteristics of search terms of $\boldsymbol{q}$, Algorithm 3 searches all the terms that belong to 'Tools', and as a result obtains

two terms 'tape' and 'ruler' that have high plausibility with those of $\boldsymbol{q}$, thereby, generating a pseudo query "tape ruler". It is obvious that the approach presents a pseudo query of good quality.

In addition, from above, we see that for a user query, Algorithm 3 only constructs one pseudo query. Actually, we can repeat Algorithms 2 and 3 several times to construct a group of pseudo queries, and the number of pseudo queries constructed for each user query is an input parameter of the approach, which would be adjusted dynamically in subsequent experiments.

4. Experiment

In this section, we present experimental evaluations for the proposed approach in two parts. The first part focuses on the effectiveness of the pseudo queries to mask the true intention of a user query. The second part quantifies the extra performance overheads caused by our approach.

4.1. Experimental setup

Before the experimental evaluation, we briefly describe the experimental setup, i.e., the reference dataset, search queries, system resource configuration and algorithm candidates.

(1) **Dataset pre-process**: From Section 3, we know that the implementations of all the algorithms depend on the reference knowledge derived from an external human repository. Specifically, the reference dataset used in the experiments, which was built based on Wikipedia, consists of 7,512,630 terms (article titles), 3,304,175 concepts (articles) and 509,407 categories. The reference dataset was created based on the enwiki dump published on 2011-10-07. First, we executed the SQL script[2] (which defines the schema of the Wikipedia database) to initialize an empty MySQL database. Second, we imported data into the MySQL database by executing the scripts[3] on relevant tables, such as Page (which is used to store the basic information of concepts), Category (which is used to store categories), Redirect (which is used to store redirect pages, and thus is often used to process synonymous concepts), Pagelinks (which are used to store the connections within concepts) and Categorylinks (which are used to store the connections within categories or between concepts and categories). Finally, with the help of the database, we extracted and rebuilt all kinds of knowledge information, e.g., terms, concepts and categories, as well as connections within concepts, within categories or between concepts and categories, consequently, generating a memory-based reference dataset. Moreover, empirically, we simply chose 30 generalized categories as the topic space $\mathcal{P}$, which are shown in Table 2. As a result, by using the reference dataset, all the knowledge information (e.g., $\mathcal{T}$, $\mathcal{E}$, $\mathcal{P}$, $\mathcal{T}(e_j)$, $\mathcal{E}(e_j)$, $\mathcal{E}(p_i)$, $depth(e_j, p_i)$, $Re(p_i, p_j)$, $F_p(t_j)$, $F_y(t_j)$, $F_o(t_j)$ etc.) mentioned in Section 3 can be obtained efficiently.

(2) **Workload**: We construct user queries by selecting search terms from the term space randomly, and the number of search terms contained in each user query is an experiment parameter and can be adjusted in our experiments. The user queries are processed systematically according to Algorithms 1–3, before being submitted to the search engine.

(3) **System configuration**: In our approach, all the algorithms were implemented by using the Java programming language. Besides, we simply set the threshold λ_u of Definition 2 to be 0 in our experiments. The experiments were performed on a Java Virtual Machine (version 1.7.0_07) with an Intel Core 2 Duo 3 GHz CPU and 2 GB of maximum working memory.

(4) **Algorithms**: We benchmark the proposed approach against the random approach (in which the pseudo terms are randomly chosen from Wikipedia). Here, we do not compare against other algorithms that are mentioned in the related work section, since these algorithms are proposed under different system models or privacy models, and are not comparable to the proposed approach. Instead, we analyzed the advantages and disadvantages of these approaches in the related work section.

4.2. Effectiveness evaluation

The effectiveness of the pseudo queries to mask the user intention is evaluated from two aspects, i.e., plausibility and innocuousness, whose evaluation metrics are designed based on Definition 4 and 6, respectively:

- **Plausibility**, measured by the characteristic similarity of a user query $\boldsymbol{q}$ to its pseudo queries $\mathcal{G}$, i.e., $\min_{\boldsymbol{q}' \in \mathcal{G}} \{Pla(\boldsymbol{q}', \boldsymbol{q})\}$. A higher value is better, because it means that the pseudo queries have more similar characteristic distribution as the genuine user query, thereby, making them difficult to be ruled out by an adversary.
- **Innocuousness**, measured by the topic irrelevance of pseudo queries $\mathcal{G}$ to their related user query $\boldsymbol{q}$. Let $\mathcal{P}(\boldsymbol{q}')$ be a group of topics that are relevant to a pseudo query $q' \in \mathcal{G}$. The metric is defined as $\min_{\boldsymbol{q}' \in \mathcal{G}} \{\min_{p'_i \in \mathcal{P}(\boldsymbol{q}')} \{Inn(p'_i, \boldsymbol{q})\}\}$. A higher value is better, because it means that the pseudo queries are more innocuous to the user intention.

Moreover, in order to establish the benchmark, for each user query, we also used a random approach to construct a group of additional pseudo queries which had the same size to the user query, but each search term was randomly chosen from the term space. Below, for convenience, we denote our proposed approach as **Wiki** and the benchmark approach as **Random**.

[2] http://en.wikipedia.org/wiki/wikipedia:database_download.
[3] http://dumps.wikimedia.org/enwiki/20111007/.

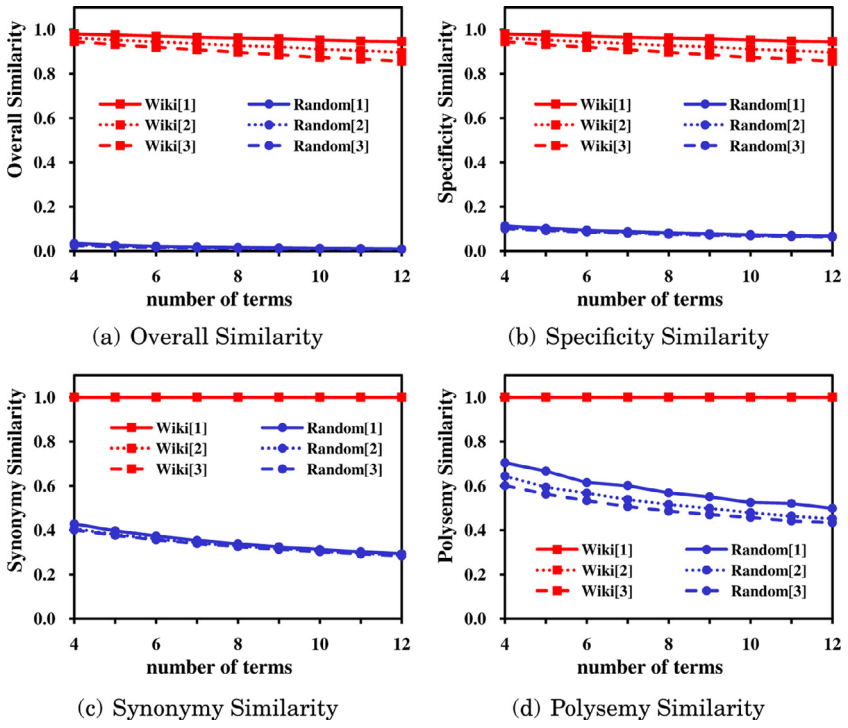

Fig. 3. Plausibility evaluation.

In the first group of experiments, we judged the plausibility of the pseudo queries. First, for each user query $\boldsymbol{q}$, we fixed the number of relevant topics at 1 (i.e., $|\mathcal{P}(\boldsymbol{q})| = 1$), and varied the number of search terms from 4 to 12 (i.e., $|\mathcal{T}(\boldsymbol{q})| = 4, 5, \ldots, 12$). Second, we used **Wiki** or **Random** to construct a group $\mathcal{G}$ of pseudo queries for $\boldsymbol{q}$. Finally, we measured the plausibility between the user query $\boldsymbol{q}$ and the pseudo queries $\mathcal{G}$. The evaluation results are shown in Fig. 3, where "Wiki[n]" (n is varied from 1 to 3) means the number of pseudo queries constructed using **Wiki** for each user query; and "Random[n]" means the number of pseudo queries constructed using **Random**.

As expected, the pseudo queries constructed using **Wiki** exhibit a much better plausibility, compared with those using **Random**. Specifically, the similarity of the pseudo queries from **Wiki** to the user queries is close to 1.0, i.e., both have highly similar characteristic distribution; and the similarity almost remains unchanged, while the number of pseudo queries or search terms is changed (this is benefited from the large term space of Wikipedia). As a result, such high plausibility makes the pseudo queries difficult to be distinguished from the genuine user queries.

In the second group of experiments, we evaluated the innocuousness of the pseudo queries to the user intention. In the experiments, we not only varied the number of search terms from 4 to 12, but also varied the number of relevant topics from 1 to 4. Fig. 4 presents the evaluation results, where the integer value in the caption of each graph indicates the number of relevant topics of each user query. It can be seen that the pseudo queries from **Wiki** are more innocuous to the user intention, compared with those from **Random**. As seen from each graph in Fig. 4, however, the irrelevance of pseudo topics to the user queries would decrease with the increasing of the number of search terms; this is caused by the increase of the relevance between the user topics and the user queries (see Eq. (1) in Definition 4). Besides, it can be seen that the innocuousness of **Wiki** would be slowly approaching **Random**, with the increasing of the number of relevant topics and search terms; this is caused by the limited size of the topic space in the experiments (we only chose 30 topics).

Based on the above experimental analysis, we conclude that our approach can generate good-quality pseudo queries for a user query, which not only have better plausibility to the user query, but also have better innocuousness to the user intention, i.e., the approach has good effectiveness.

4.3. Performance evaluation

The extra running time overheads are mainly caused by the following three parts: (1) generating pseudo queries by the client; (2) executing pseudo queries by the search engine; and (3) the network traffic volume between the client and the search engine. According to the analysis of time complexity of 1,–3, we know that the overhead from the first part is not dominant (it is generally less than 1 ms), so we will focus on the time overhead from the last two parts. In this group of experiments, we first used our approach to construct a group of pseudo queries for each user query; then, issued the pseudo queries together with the user query to Google; and finally, examined the returned results to compute related time performance metrics. Here, the time

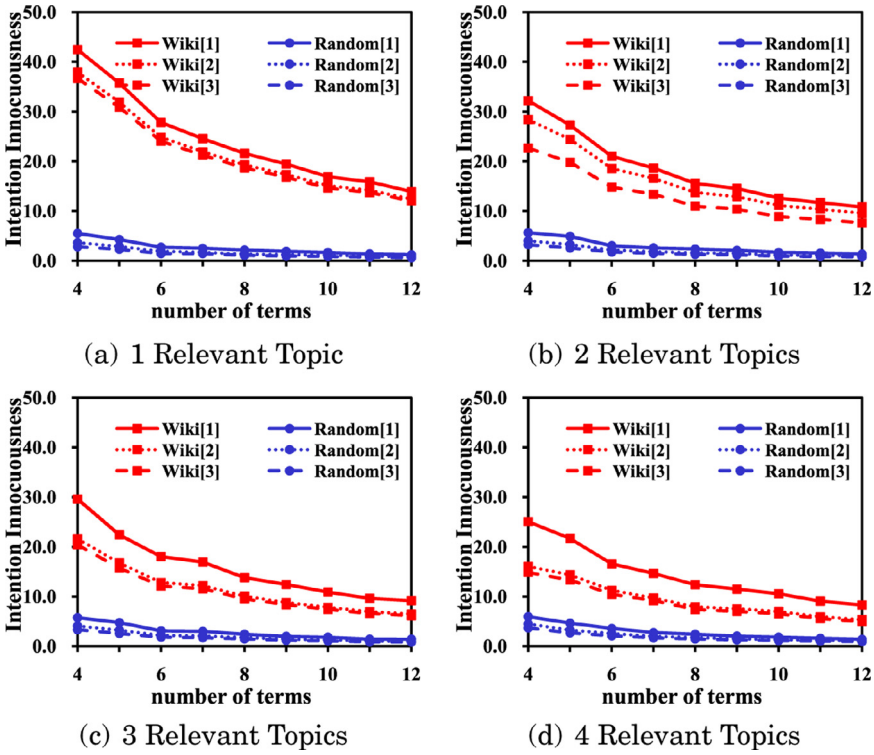

Fig. 4. Innocuousness evaluation.

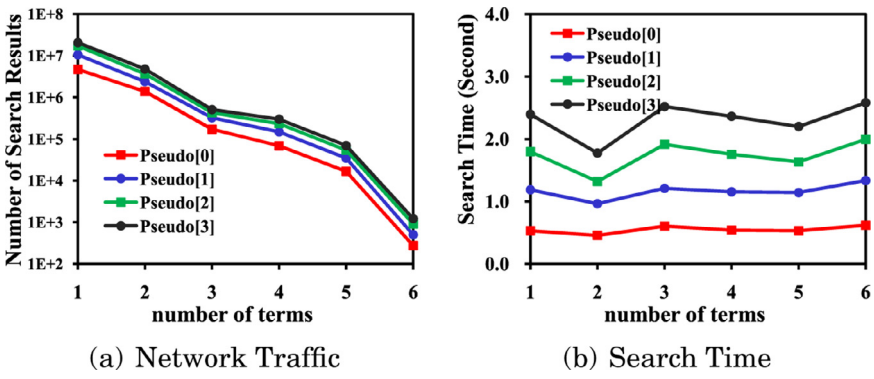

Fig. 5. Performance evaluation.

performance metrics include the search time, and the number of returned items (because the number of the returned items is positively correlated with the network traffic, we used it as the metric of network traffic).

Fig. 5 presents the experiment results, where "Pseudo[n]" denotes the number of pseudo queries issued together with each user query. As observed from the figures, it is inevitable that both the search time and the network traffic would grow with the increasing of the number of pseudo queries; and the growth is approximately in direct proportion to the number of pseudo queries. Besides, Fig. 5 also shows that the number of the search items returned from a pseudo query is roughly identical to that from a genuine user query, to a large extent. This is due to the high specificity plausibility of each pseudo query to its corresponding user query.

Besides time performance, another extra overhead is caused by the space allocation for the reference dataset from Wikipedia. In the dataset, there are more than 7,000,000 terms. In the experiments, to load all the terms and their additional information, we used about 780 MB main memory space. Thus, it is necessary for a client to assign at least 1 GB working memory space to the Java Virtual Machine. We think that such a space overhead should be acceptable under present hardware resource configuration.

Based on the above experimental analysis, we conclude that our approach would not cause serious extra performance overheads on either running time or memory space, i.e., the approach has good practicality.

5. Conclusion

In this paper, we proposed an approach to construct plausible innocuous pseudo queries to protect the query intention for users when using search engines. The client-based model we used makes the approach require not only no change to existing search engines, but also no compromise to precision-recall performance. Moreover, we conducted extensive experimental evaluations, and the experimental results demonstrated the effectiveness and practicality of the approach: (1) it can generate good-quality pseudo queries for a user query, which not only have better plausibility to the user query, but also are more innocuous to the user intention; and (2) it does not cause serious performance overheads, whether running time overheads or memory space overheads. Thus, we conclude that the user query intention can be protected effectively using our approach.

In the future, we plan to further study how to protect the user intention across a sequence of user queries, instead of single user query. In addition, our approach is proposed based on the reference knowledge from Wikipedia. Wikipedia is an open repository built collaboratively by volunteers, so sometimes its contents are not reliable. Thus, we also plan to integrate other repositories into the approach.

Acknowledgments

We thank anonymous reviewers and Prof. S. Tang for their valuable comments. The work is supported by the Zhejiang Provincial Natural Science Foundation of China (nos. LY15F020020, LQ13F020009 and LY13F010005), the Hubei Provincial Natural Science Foundation of China (no. 2013CFB415), the State Key Laboratory of Biogeology and Environmental Geology, China University of Geosciences (no. GBL31505), the China Postdoctoral Science Foundation funded projects (nos. 2012M521251 and 2013T60623) and the National Natural Science Foundation of China (nos. 61202171, 61303113, 61300227 and 61402337).

References

[1] M. Murugesan, C. Clifton, Providing privacy through plausibly deniable search, in: Proceedings of the SIAM International Conference on Data Mining, 2009, pp. 768–779.
[2] H. Pang, X. Xiao, J. Shen, Obfuscating the topical intention in enterprise text search, in: Proceedings of the IEEE International Conference on Data Engineering, 2012, pp. 1168–1179.
[3] D. Sánchez, J. Castellà-Roca, A. Viejo, Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines, Inf. Sci. 218 (2013) 17–30.
[4] M. Murugesan, J. Wei, C. Clifton, L. Si, J. Vaidya, Efficient privacy-preserving similar document detection, VLDB J. 19 (4) (2010) 257–275.
[5] M. Batet, A. Erola, D. Sánchez, J. Castellà-Roca, Utility preserving query log anonymization via semantic microaggregation, Inf. Sci. 242 (2013) 49–63.
[6] R. Dingledine, N. Mathewson, Tor: the second-generation onion router, in: Proceedings of the USENIX Security Symposium, 2004, pp. 303–320.
[7] M. Barbaro, T. Zeller, S. Hansell, A face is exposed for AOL searcher no. 4417749, New York Times 9 (2006).
[8] R. Jones, R. Kumar, B. Pang, A. Tomkins, "I know what you did last summer" – query logs and user privacy, in: Proceedings of the ACM Conference on Information and Knowledge Management, 2007, pp. 909–914.
[9] G. Ateniese, A.D. Santis, A.L. Ferrara, B. Masucci, Provably-secure time-bound hierarchical key assignment schemes, J. Cryptol. 25 (2) (2012) 243–270.
[10] J. Bethencourt, D. Song, B. Waters, New constructions and practical applications for private stream searching, in: Proceedings of the IEEE Symposium on Security and Privacy, 2006, pp. 132–139.
[11] D. Boneh, G.D. Crescenzo, R. Ostrovsky, G. Persiano, Public key encryption with keyword search, Lect. Notes Comput. Sci. 49 (16) (2004) 506–522.
[12] M.J. Freedman, Y. Ishai, B. Pinkas, O. Reingold, Keyword search and oblivious pseudorandom functions, Lect. Notes Comput. Sci. (2005) 303–324.
[13] A. Baumeler, A. Brodbent, Quantum private information retrieval has linear communication complexity, J. Cryptol. 28 (1) (2013) 161–175.
[14] B. Chor, E. Kushilevitz, O. Goldreich, M. Sudan, Private information retrieval, J. ACM 45 (6) (1998) 965–981.
[15] D.C. Howe, H. Nissenbaum, Trackmenot: resisting surveillance in web search, Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society, 2009, pp. 417–436.
[16] H. Pang, K. Mouratidis, Authenticating the query results of text search engines, Proc. VLDB Endowment 1 (1) (2008) 126–137.
[17] H. Hacigümüş, B. Iyer, C. Li, S. Mehrotra, Executing SQL over encrypted data in the database-service-provider model, in: Proceedings of the ACM SIGMOD, 2002, pp. 216–227.
[18] Z. Wu, G. Xu, Z. Yu, X. Yi, E. Chen, Y. Zhang, Executing SQL queries over encrypted character strings in the database-as-service model, Knowl. Based Syst. 35 (2012) 332–348.
[19] H. Xu, S. Guo, K. Chen, Building confidential and efficient query services in the cloud with rasp data perturbation, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 322–335.
[20] H. Pang, J. Shen, R. Krishnan, Privacy-preserving similarity-based text retrieval, ACM Trans. Internet Technol. 10 (1) (2010) 4.
[21] R. Paulet, M.G. Koasar, X. Yi, E. Bertino, Privacy-preserving and content-protecting location based queries, IEEE Trans. Knowl. Data Eng. 26 (5) (2014) 1200–1210.
[22] X. Shen, B. Tan, C. Zhai, Privacy protection in personalized search, ACM SIGIR Forum 41 (1) (2007) 4–17.
[23] J. Wei, M. Murugesan, C. Clifton, L. Si, Similar document detection with limited information disclosure, in: Proceedings of the IEEE International Conference on Data Engineering, 2008, pp. 735–743.
[24] P. Husbands, H. Simon, C. Ding, On the use of the singular value decomposition for text retrieval, SIAM Comput. Inf. Retr. (2001) 145–156.
[25] H. Pang, X. Ding, X. Xiao, Embellishing text search queries to protect user privacy, Proc. VLDB Endowment 3 (1–2) (2010) 598–607.
[26] J. Hu, G. Wang, F. Lochovsky, J. Sun, Z. Chen, Understanding user's query intent with Wikipedia, in: Proceedings of the International World Wide Web Conference, 2009, pp. 471–480.
[27] Y. Jiang, X. Zhang, Y. Tang, R. Nie, Feature-based approaches to semantic similarity assessment of concepts using Wikipedia, Inf. Process. Manag. 51 (3) (2015) 215–234.
[28] G. Xu, Z. Wu, G. Li, E. Chen, Improving contextual advertising matching by using Wikipedia thesaurus knowledge, Knowl. Inf. Syst. 43 (3) (2015) 599–631.
[29] P. Wang, J. Hu, H. Zeng, Z. Chen, Using Wikipedia knowledge to improve text classification, Knowl. Inf. Syst. 19 (3) (2009) 265–281.