

# Anonymising Queries by Semantic Decomposition

Danushka Bollegala  
University of Liverpool, Liverpool,  
L693BX, UK.  
danushka@liverpool.ac.uk

Tomoya Machide  
National Institute of Informatics, 2-1-2  
Hitotsubashi, Chiyoda-ku, Tokyo,  
101-8430, Japan.  
machide@nii.ac.jp

Ken-ichi Kawarabayashi  
National Institute of Informatics, 2-1-2  
Hitotsubashi, Chiyoda-ku, Tokyo,  
101-8430, Japan.  
k\_keniti@nii.ac.jp

## ABSTRACT

Protecting the privacy of search engine users is an important requirement in many information retrieval scenarios. A user might not want a search engine to guess his or her information need despite requesting relevant results. We propose a method to protect the privacy of search engine users by decomposing the queries using semantically *related* and unrelated *distractor* terms. Instead of a single query, the search engine receives multiple decomposed query terms. Next, we reconstruct the search results relevant to the original query term by aggregating the search results retrieved for the decomposed query terms. We show that the word embeddings learnt using a distributed representation learning method can be used to find semantically related and distractor query terms. We derive the relationship between the *anonymity* achieved through the proposed query anonymisation method and the *reconstructability* of the original search results using the decomposed queries. We analytically study the risk of discovering the search engine users' information intents under the proposed query anonymisation method, and empirically evaluate its robustness against clustering-based attacks. Our experimental results show that the proposed method can accurately reconstruct the search results for user queries, without compromising the privacy of the search engine users.

## KEYWORDS

Query Anonymisation | Information Retrieval | Word Embeddings | Anonymity | Reconstructability

### ACM Reference Format:

Danushka Bollegala, Tomoya Machide, and Ken-ichi Kawarabayashi. . Anonymising Queries by Semantic Decomposition. In . ACM, New York, NY, USA, 11 pages.

## 1 INTRODUCTION

Information retrieval systems have become essential tools in our day-to-day activities. We constantly search information on the Web using search engines by issuing keywords that describe our information needs. However, we might not always want the search engine to discover our intent through the keywords we use in a search session. For example, a patient with a particular disease might want to use a web search engine to find information about his/her disease, but at the same time might not want to disclose his/her health conditions.

As web search engine users, we are left with two options regarding our privacy. First, we can trust the search engine not to disclose the keywords that we use in a search session to third parties, or even to use for any other purpose other than providing search results to the users who issued the queries. However, the user agreements in

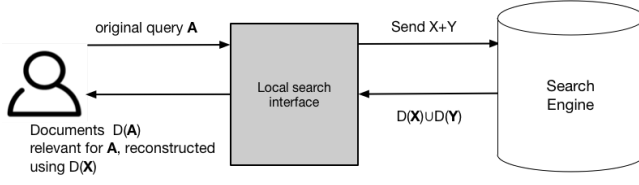
most web search engines do not allow such user rights. Although search engines pledge to protect the privacy of their users by encrypting queries and search results<sup>1</sup>, the encryption is between the user and the search engine – the original non-encrypted queries are still available to the search engine. The keywords issued by the users are a vital source of information for improving the relevancy of the search engine and displaying relevant adverts to the users. For example, in learning to rank [9], keywords issued by a user and the documents clicked by that user are recorded by the search engine to learn the optimal dynamic ranking of the search results. Query logs have been used extensively to learn the user interests and extract attributes related to frequently searched entities [19, 20, 24, 25, 27]. Considering the fact that placing advertisements for the highly bid keywords is one of the main revenue sources for search engines, there are obvious commercial motivations for the search engines to exploit the user queries beyond simply providing relevant search results to their users. For example, it has been reported that advertisements contribute to 96% of Google's revenue<sup>2</sup>. Therefore, it would be unwise to assume that the user queries will not be exploited in a manner unintended by the users

As an alternative approach that does not rely on the goodwill of the search engine companies, we propose a method (shown in Figure 1), where we disguise the queries that are sent to a search engine such that it is difficult for the search engine to guess the real information need of the user by looking at the keywords, yet it is somehow possible for the users to *reconstruct* the search results relevant for them from what is returned by the search engine. The proposed method does not require any encryption or blindly trusting the search engine companies or any third-party mediators. However, this is a non-trivial task because a search engine must be able to recognise the information need of a user in order to provide relevant results in the first place. Therefore, query anonymisation and relevance of search results are at a direct trade-off.

Specifically, given a user query  $A$ , our proposed method first finds a set of  $n$  noisy relevant terms  $X_1, X_2, \dots, X_n$  and  $m$  distractor terms  $Y_1, Y_2, \dots, Y_m$  for  $A$ . We use pre-trained word embeddings for identifying the noisy-relevant and distractor terms. We add Gaussian noise to the relevant terms such that it becomes difficult for the search engine to discover  $A$  using  $X_1, X_2, \dots, X_n$ . However,  $X_1, X_2, \dots, X_n$  are derived using  $A$ , so there is a risk that the search engine will perform some form of de-noising to unveil  $A$  from  $X_1, X_2, \dots, X_n$ . Therefore, using  $X_1, X_2, \dots, X_n$  alone as the keywords does not guarantee anonymity. To mitigate this risk, we generate a set of distractor terms  $Y_1, Y_2, \dots, Y_m$  separately for each user query. We then issue  $X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_m$  in random order to the search engine

<sup>1</sup><https://goo.gl/JSBvpK>

<sup>2</sup><https://www.wordstream.com/articles/google-earnings>



**Figure 1: Overview of the proposed method. The original query  $A$  is decomposed into a set of relevant ( $X$ ) and distractor ( $Y$ ) terms at the user-end. The search engine returns documents relevant for both  $X$  and  $Y$ , denoted by  $D(X) \cup D(Y)$ . We will ignore  $D(Y)$  and reconstruct the search results for  $A$  using  $D(X)$ .**

to retrieve the corresponding search results. We then reconstruct the search results for  $A$  using the search results we retrieve from the noisy-relevant terms and discard the search results retrieved from the distractor terms. It is noteworthy that during any stage of the proposed method, we *do not* issue  $A$  as a standalone query nor in conjunction with any other terms to the search engine. Moreover, we do not require access to the search index, which is typically not shared by the search engine companies with the outside world.

Our contributions in this paper can be summarised as follows:

- We propose a method to anonymise user queries sent to a search engine by semantic decomposition to protect the privacy of the search engine users. Our proposed method uses pretrained word embeddings.
- We introduce the concepts of *anonymity* (i.e., how difficult it is to guess the original user query by looking at the auxiliary queries sent to the search engine?), and *reconstructability* (i.e. how easy it is to reconstruct the search results for the original query from the search results for the auxiliary queries?), and propose methods to estimate their values.
- We theoretically derive the relationship between anonymity and reconstructability using known properties of distributed word representations.
- We evaluate the robustness of the proposed query anonymisation method against clustering-based attacks, where a search engine would cluster the keywords it receives within a single session to filter our distractors and predict the original query from the induced clusters. Our experimental results show that by selecting appropriate distractor terms, it is possible to guarantee query anonymity, while reconstructing the relevant search results.

## 2 QUERY ANONYMISATION VIA SEMANTIC DECOMPOSITION

### 2.1 Retrieval Model

Modern search engines use a complex retrieval mechanism that involves search result ranking, sessions, personalisation, etc. Moreover, the exact implementations of those mechanisms differ from one search engine to another and not publicly disclosed. Therefore, to simplify the theoretical analysis and empirical validation, we resort to a classical inverted index-based information retrieval, where we return *all* documents that contain all words in the query, without performing any relevance ranking.

### 2.2 Finding Noisy-Related Terms

Expanding a user query using related terms is a popular technique in information retrieval [2], and is particularly useful when the number of results for the original query is small or zero. For example, consider the scenario that a user wants to obtain search results for *Bill Gates*. In a typical search engine, we would search using the query *Bill Gates* and retrieve documents that contain the phrase *Bill Gates*. However, assuming that we did not find any documents containing *Bill Gates*, we could automatically expand the original query using its related terms to overcome the zero results problem. For example, we could expand *Bill Gates* using the related term *Microsoft*, which is a company founded by *Bill Gates*.

Although query expansion using related terms is motivated as a technique for improving the recall in a search engine, we take a different perspective in this paper – we consider query expansion as a method for anonymising the search intent of a user. Numerous methods have been proposed in prior work on query expansion to find good candidate terms for expanding a given user query such as using pre-compiled thesauri containing related terms and query logs [2]. We note that any method that can find related terms for a given user query  $A$  can be used for our purpose given that the following requirements are satisfied:

- (1) The user query  $A$  must never be sent to the search engine when retrieving related terms for  $A$  because this would obviously compromise the anonymisation goal.
- (2) Repeated queries to the search engine must be minimised in order to reduce the burden on the search engine. We assume that the query anonymisation process to take place outside of the search engine using a publicly available search API. Although modern Web search engines would gracefully scale with the number of users/queries, anonymisation methods that send excessively large numbers of queries are likely to be banned by the search engines because of the processing overhead. Therefore, it is important that we limit the search queries that we issue to the search engine when computing the related terms.
- (3) No information regarding the distribution of documents nor the search index must be required by the related term identification method. If we had access to the index of the search engine, then we could easily find the terms that are co-occurring with the user query, thereby identifying related terms. However, we assume that the query anonymisation process happens outside of the search engine. None of the major commercial web search engines such as Google, Bing or Baidu provide direct access to their search indices due to security concerns. Therefore, it is realistic to assume that we will not have access to the search index during anytime of the anonymisation process, including the step where we find related terms to a given user query.
- (4) The related terms must not be too similar to the original user query  $A$  because that would enable the search engine to guess  $A$  via the related terms it receives. For this purpose, we would add noise to the user query  $A$  and find *noisy related neighbours* that are less similar to  $A$ .

We propose a method that use pre-trained word embeddings to find related terms for a user query that satisfy all of the above-mentioned requirements. Context-independent word embedding methods such as word2vec [15] and GloVe [21] can represent the meanings of words using low dimensional dense vectors. Using word embeddings is also computationally attractive because they are low dimensional (typically 100 – 600 dimensions are sufficient), consuming less memory and faster when computing similarity scores. Although we focus on single word queries for ease of discussion, we note that by using context-sensitive phrase embeddings such as Elmo [22] and BERT [5] we can obtain vectors representing multi-word queries, which we defer to future work.

We denote the pretrained word embedding of a term  $A$  by  $v(A)$ . To perturbate word embeddings, we add a vector,  $\theta \in \mathbb{R}^d$ , sampled independently for each  $A$  from the  $d$ -dimensional Gaussian with a zero mean and a unit variance, and measure the cosine similarity between  $v(A) + \theta$  and each of the words  $X_i \in \mathcal{V}$  in a predefined and fixed vocabulary  $\mathcal{V}$ , using their word embeddings  $v(X_i)$ . We then select the top most similar words  $X_1, X_2, \dots, X_n$  as the noisy related terms of  $A$ .

Let us denote the set of documents retrieved using a query  $A$  by  $\mathcal{D}(A)$ . If we use a sufficiently large number of related terms  $X_i$  to  $A$ , we will be able to retrieve  $\mathcal{D}(A)$  exactly using

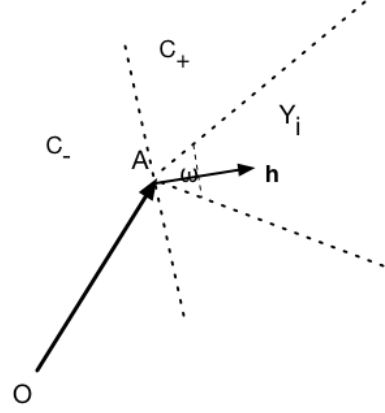
$$\mathcal{D}'(A) = \bigcup_{i=1}^n \mathcal{D}(X_i). \quad (1)$$

However, in practice we are limited to using a truncated list of  $n$  related terms  $X_1, X_2, \dots, X_n$  because of computational efficiency and to limit the number of queries sent to the search engine. Therefore, in practice  $\mathcal{D}'(A)$  will not be exactly equal to  $\mathcal{D}(A)$ . Nonetheless, we assume the equality to hold in (1), and later analyse the approximation error. We do not consider the problem of ranking the search results in this work, and focus only on reconstructing the set of search results. If the number of relevant search result is large and we would like to rank the most relevant search results at the top, we can still use static or dynamic ranking information provided by the search engine [9].

### 2.3 Anonymisation via Distractor Terms

Searching using noisy related terms  $X_i$  alone of a user query  $A$ , does not guarantee the anonymity of the users. The probability of predicting the original user query increases with the number of related terms used. Therefore, we require further mechanisms to ensure that it will be difficult for the search engine to predict  $A$  from the queries it has seen. For this purpose, we select a set of unrelated terms  $\{Y_1, Y_2, \dots, Y_m\}$ , which we refer to as the *distractor* terms.

Several techniques can be used to find the distractor terms for a given query  $A$ . For example, we can randomly select terms from the vocabulary  $\mathcal{V}$  as the distractor terms. However, such randomly selected distractor terms are unlikely to be coherent, and could be easily singled-out from the related terms by the search engine. If we know the semantic category of  $A$  (e.g.  $A$  is a *person* or a *location* etc.), then we can limit the distractor terms to the same semantic category as  $A$ . This will guarantee that both related terms as well as distractor terms are semantically related in the sense that they both represent the same category. Therefore, it will be difficult for the



**Figure 2: Selecting distractor terms for a given query  $A$ .** We first compute the noise ( $\theta$ ) added vector  $A'$  for  $A$ , and then search for terms  $Y_i$  that are located inside a cone that forms an angle  $\omega$  with  $A'$ . This would ensure that distractor terms are sufficiently similar to the noise component, therefore difficult to distinguish from  $A$ .

search engine to discriminate between related terms and distractor terms. Information about the semantic categories of terms can be obtained through different ways such as Wikipedia category pages, taxonomies such as the WordNet [16] or by named entity recognition (NER) tools.

We propose a method to find distractor terms  $Y_i$  for each query  $A$  using pretrained word embeddings as illustrated in Figure 2. Let us consider a set of candidate terms  $C$  from which we must select the distractor terms. For example,  $C$  could be a randomly selected subset from the vocabulary of the corpus used to train word embeddings. First, we select a random hyperplane (represented by the normal vector  $h \in \mathbb{R}^d$  to the hyperplane) in the embedding space that passes through the point corresponding to  $A$ . Next, we split  $C$  into two mutually exclusive sets  $C_+ = \{x : x \in C, x^T h \geq 0\}$  and  $C_- = \{x : x \in C, x^T h < 0\}$  depending on which side of the hyperplane the word is located. Let us define  $C_{\max}$  and  $C_{\min}$  to be respectively the larger and smaller of the two sets  $C_+$  and  $C_-$  (i.e.  $C_{\max} = \operatorname{argmax}_{S \in \{C_+, C_-\}} |S|$  and  $C_{\min} = \operatorname{argmin}_{S \in \{C_+, C_-\}} |S|$ ). Next, we remove the top 10% of the similar words in  $C_{\max}$  to the original query  $A$ . We then use this reduced  $C_{\max}$  as  $C$  (i.e.  $C \leftarrow C_{\max}$ ) and repeat this process until we are left with the desired number of distractor terms in  $C$ . Intuitively, we are partitioning the candidate set into two groups in each iteration considering some attribute (dimension) of the word embedding of the query (possibly representing some latent meaning of the query), and removing similar terms in that subspace.

### 2.4 Reconstructing Search Results

For a query,  $A$ , once we have identified a set of noisy related terms,  $\{X_1, X_2, \dots, X_n\}$ , and a set of distractor terms,  $\{Y_1, Y_2, \dots, Y_m\}$ , we will issue those terms as queries to the search engine and retrieve the relevant search results for each individual term. We issue related terms and distractor terms in a random sequence, and ignore the

results returned by the search engine for the distractor terms. Finally, we can reconstruct the search results for  $A$  using (1).

### 3 ANONYMITY VS. RECONSTRUCTABILITY

Our proposed query decomposition method strikes a fine balance between two factors (a) the difficulty for the search engine to guess the original user query  $A$ , from the set of terms that it receives  $Q(A) = \{X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_m\}$ , and (b) the difficulty to reconstruct the search results,  $\mathcal{D}(A)$ , for the original user query,  $A$ , using the search results for the noisy related terms following (1). We refer to (a) as the *anonymity*, and (b) as the *reconstructability* of the proposed query decomposition process.

#### 3.1 Anonymity

We define *anonymity*,  $\alpha$ , as the ease (or alternatively difficulty) to guess the user query  $A$ , from the terms issued to the search engine and compute it as follows:

$$\alpha = 1 - \frac{1}{|Q(A)|} \sum_{q \in Q(A)} \text{sim}(v(A), v(q)) \quad (2)$$

Specifically, we measure the average cosine similarity between the word embedding,  $v(A)$ , for the original user query  $A$ , and the word embeddings  $v(q)$  for each of  $q \in Q(A)$  search terms. If the similarity is higher, then it becomes easier for the search engine to guess  $A$  from the search terms. The difference between this average similarity and 1 (i.e. the maximum value for the average similarity) is considered as a measure of anonymity we can guarantee through the proposed query decomposition process.

#### 3.2 Reconstructability

We reconstruct the search results for  $A$  using the search results for the queries  $X_1, X_2, \dots, X_n$  following (1). We define *reconstructability*,  $\rho$  as a measure of the accuracy of this reconstruction process and is defined as follows:

$$\rho = \frac{|\mathcal{D}(A) \cap \mathcal{D}'(A)|}{|\mathcal{D}(A)|} \quad (3)$$

A document retrieved by only a single noisy related term might not be relevant to the original user query  $A$ . A more robust reconstruction procedure would be to consider a document as relevant if it has been retrieved by at least  $l$  different noisy related terms. If a user query  $A$  can be represented by a set of documents where, each document is retrieved by at least  $l < n$  different noisy related terms, then we say  $A$  to be *l-reconstructable*. In fact, the reconstruction process defined in (1) corresponds to the special case where  $l = 1$ . Increasing the value of  $l$  would decrease the number of relevant documents retrieved for the original user query  $A$ , but it is likely to increase the relevance of the retrieval process.

### 4 RELATIONSHIP BETWEEN ANONYMITY AND RECONSTRUCTABILITY

In this section, we derive the relationship between anonymity and reconstructability. Because anonymity can be increased arbitrarily by increasing the distractor terms, in this analysis, we ignore distractor terms. This can be seen as a lower-bound for the anonymity that you can obtain, without using any distractor terms. We first discuss the

case where we have only one related term (i.e.  $n = l = 1$ ) and then consider  $l > 1$  reconstructability case.

#### 4.1 $n = l = 1$ case

Let us consider the case where  $n = 1$ . Here, for a given query  $A$ , we have only a single related term  $X = X_1$ . In this case,  $l = 1$ , and we consider all documents retrieved using  $X$  as relevant for  $A$ . We first note that reconstructability,  $\rho$ , can be written as,

$$\rho = \frac{|\mathcal{D}(A) \cap \mathcal{D}'(A)|}{|\mathcal{D}(A)|} \quad (4)$$

from the definition of reconstructability.

Because we have a single noisy related term  $X$ , we have  $\mathcal{D}'(A) = \mathcal{D}(X)$ . By substituting this in (4), we get

$$\rho = \frac{|\mathcal{D}(A) \cap \mathcal{D}(X)|}{|\mathcal{D}(A)|}. \quad (5)$$

If we consider the co-occurrence context of two terms to be the document in which they co-occur, then (5) can be written as a conditional probability given by (6).

$$\rho = \frac{p(A, X)}{p(A)} = p(X|A) \quad (6)$$

Theorem 2.2 in [1] provides a useful connection between the probability of a word (or the joint probability of two words) and their word representations, which we summarise below.

$$\log p(A, X) = \frac{\|v(A) + v(X)\|_2^2}{2d} - 2 \log Z \pm \epsilon \quad (7)$$

$$\log p(A) = \frac{\|v(A)\|_2^2}{2d} - \log Z \pm \epsilon \quad (8)$$

Here,  $Z$  is the partition function and  $\epsilon$  is the approximation error.

By taking the logarithm of both sides in (6) we obtain,

$$\begin{aligned} \log \rho &= \log p(A, X) - \log p(A) \\ &= \frac{\|v(X)\|_2^2 + 2v(X)^\top v(A)}{2d} - \log Z \end{aligned} \quad (9)$$

Anonymity for a single query term  $X$  can be computed using cosine similarity as follows:

$$\alpha = 1 - \frac{v(A)^\top v(X)}{\|v(A)\|_2 \|v(X)\|_2} \quad (10)$$

By substituting (10) in (9) we get,

$$\log \rho = \frac{\|v(X)\|_2^2}{2d} + \frac{(1 - \alpha) \|v(A)\|_2 \|v(X)\|_2}{d} - \log Z. \quad (11)$$

Because  $A$  is a given query,  $v(A)$  is a constant. Moreover, if we assume that different related terms  $X_i$  have similar norms, (from (8) it follows that such related terms must have similar frequencies of occurrence in the corpus), then from (11) we see that there exists a linear inverse relationship between  $\log \rho$  and  $\alpha$ . Because logarithm function is monotonically increasing (11) implies an inverse relationship between  $\rho$  and  $\alpha$ .

#### 4.2 $n = l > 1$ case

Let us now extend the relationship given by (11) to the case where we consider a document to be relevant if it can be retrieved from all of the  $n$  related terms. In other words, we have  $l = n$  reconstructability in this case. Because each search result is retrieved by all  $l$  terms, we have

$$\mathcal{D}'(A) = \cap_{i=1}^l \mathcal{D}(X_i). \quad (12)$$

Reconstructability can be computed in this case as follows:

$$\rho = \frac{p(A, X_1, X_2, \dots, X_l)}{p(A)} = p(X_1, X_2, \dots, X_l | A) \approx \prod_{i=1}^l p(X_i | A) \quad (13)$$

In (13) we have assumed that the related terms are mutually independent given the query  $A$ .

Let us take the logarithm on both sides of (13), and use (7) and (8) in the same manner as we did in Section 4.1 to derive the relationship given by (14).

$$\log \rho = \frac{1}{2d} \sum_{i=1}^l \|v(X_i)\|_2^2 + \frac{1}{d} \sum_{i=1}^l v(A)^\top v(X_i) - \log Z \quad (14)$$

In the  $n = l$  case, anonymity can be computed as follows:

$$\alpha = 1 - \frac{1}{l} \sum_{i=1}^l \frac{v(A)^\top v(X_i)}{\|v(A)\|_2 \|v(X_i)\|_2} \quad (15)$$

Let us further assume that all related terms  $X_1, X_2, \dots, X_l$  occur approximately the same number of times in the corpus. From (8) it then follows that  $\|v(X_i)\|_2 = c$  for  $i = 1, 2, \dots, l$  for some  $c \in \mathbb{R}$ . By plugging (15) in (14), and using the approximation  $\|v(X_i)\|_2 = c$  we arrive at the relationship between  $\rho$ ,  $\alpha$ , and  $l$  given by (16).

$$\log \rho = \frac{cl}{2d} (c + 2(1 - \alpha) \|v(A)\|_2) - \log Z \quad (16)$$

□

In the general case of  $l$ -reconstructability, we will have a subset of  $l \leq n$  related terms retrieving each document. The reconstructability given by (16) must be considered as a lower-bound for this general case because we will still be able to reconstruct the search results using  $\binom{n}{l}$  subsets of  $l$  related terms selected from a set of  $n$  related terms.

## 5 EXPERIMENTS AND RESULTS

To evaluate the proposed method we create a dataset where we select 50 popular queries from Wikipedia query logs and associate them with the relevant Wikipedia articles. The 50 query terms used in our experiments are as follows: *airfield, alex, anthropology, antoine, antony, autonomous, belfast, ben, benares, benet, benz, biodiversity, broadway, carol, commercial, consciousness, crown, custer, earths, elena, gallery, haddad, haig, helmut, hughes, hugo, irit, judith, kahn, katarina, leith, marshal, masaaki, memorial, negro, oakley, outlaw, product, rings, runaway, sammy, santa, sine, stawell, steve, toole, tube, wait, wilkerson, angel*.

We use December 2015 dump of English Wikipedia for this purpose and build a keyword-based inverted search index. We use 300 dimensional pretrained GloVe [21] embeddings trained from a 42

billion token Web crawled corpus<sup>3</sup> as the word embeddings for computing relevant terms. Figures 3-5 show the anonymity and logarithm of the reconstructability values for the 50 queries in our dataset at three different levels of noise and different numbers of distractor terms. Specifically, we add Gaussian noise with zero mean and standard deviations of 0.6 and 1.0 respectively to stimulate medium and high levels of noise, whereas the no-noise case corresponds to not perturbing the word embeddings.

We see a negative correlation between anonymity and reconstructability in all plots as predicted by (16). Moreover, the absolute value of the negative correlation increases with the noise level in all cases with different numbers of distractor terms. Addition of noise affects the selection of related terms. It does not affect the selection of distractor terms. However, related terms influence both anonymity as well as reconstructability. Because the Gaussian noise is added to the word embedding of the original query, and the nearest neighbours to this noise added embedding are selected as the related terms, this process would help to increase anonymity. On the other hand, the search results obtained using noisy related terms will be less relevant to the original user query. Therefore, reconstructing the search results for the original user query using the search results for the noisy related terms will become more difficult, resulting in decreasing the reconstructability. The overall effect of increasing anonymity and decreasing reconstructability is shown by the increased negative gradient of the line of best fit in the figures.

### 5.1 Robustness against Attacks

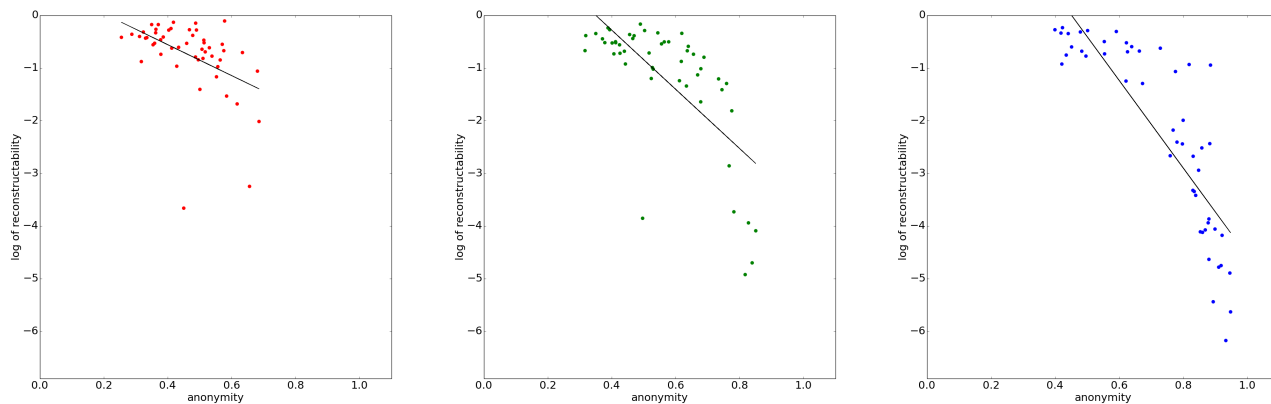
An important aspect of a query anonymisation method is how robust it will be against attacks. Given that the proposed method sends two groups of terms (relevant and distractor) to a search engine, a natural line of attack will be for the search engine to cluster the received terms to filter out distractor terms and then guess the user query from the relevant terms. We call such attacks as *clustering attacks* in this paper.

As a concrete example, we simulate a clustering attacker who applies  $k$ -means clustering to the received terms. The similarity between terms for the purpose of clustering is computed using the cosine similarity between the corresponding word embeddings. Any clustering algorithm can be used for this purpose. We use  $k$ -means clustering because of its simplicity. Next, the attacker must identify a single cluster that is likely to contain the relevant terms. For this purpose, we measure the *coherence*,  $\mu(C)$ , of a cluster  $C$  given by (17).

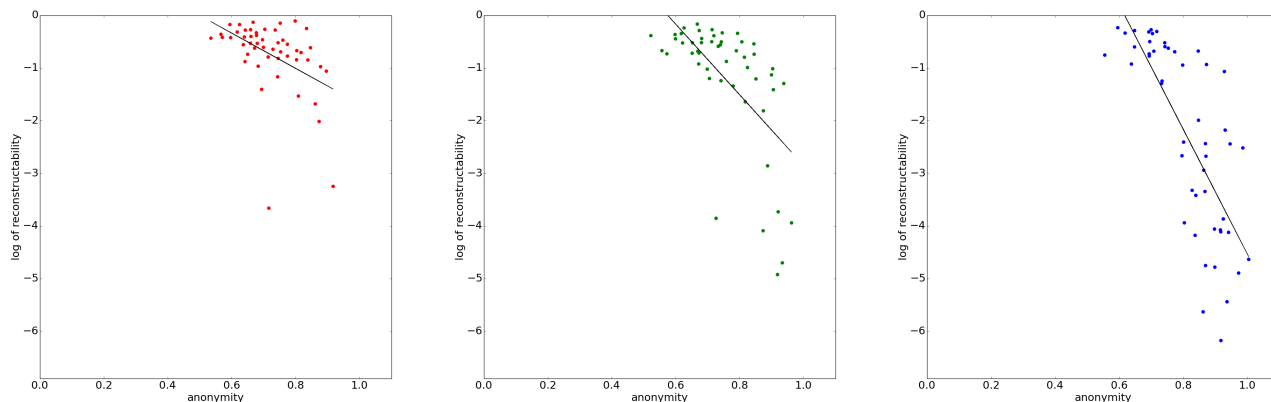
$$\mu(C) = \frac{2}{|C|(|C| - 1)} \sum_{u, v \in C, u \neq v} \text{sim}(u, v) \quad (17)$$

Here,  $u, v \in C$  are two distinct terms in  $C$ . Because a cluster containing relevant terms will be more coherent than a cluster containing distractor terms, the attacker selects the cluster with the highest coherence as the relevant cluster. Finally, we find the term from the entire vocabulary that is closest to the centroid of the cluster as the guess  $\hat{A}$  of the original user query  $A$ . We define *hit rate* to be the proportion of the queries that we disclose via the clustering attack. Figure 6 shows the hit rates for the clustering attacks under different numbers of distractor terms.

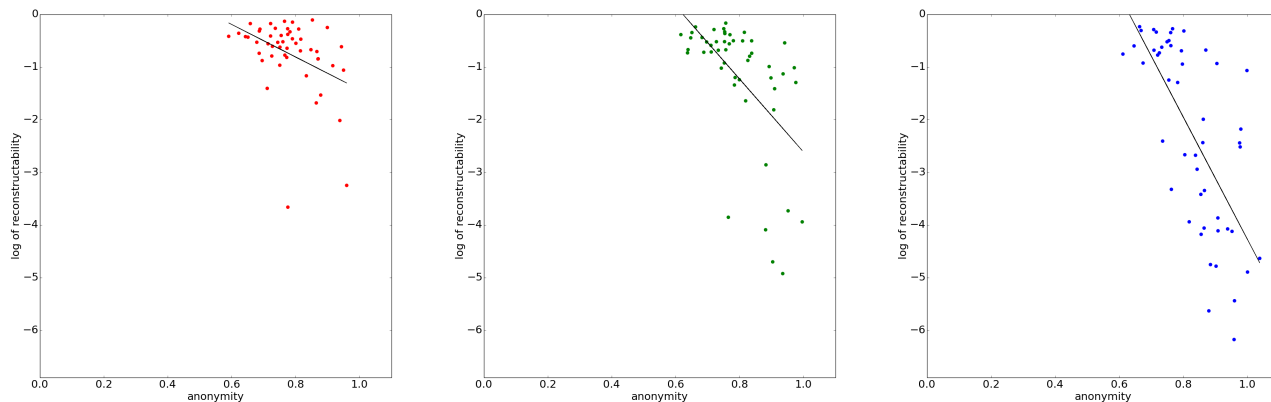
<sup>3</sup><https://nlp.stanford.edu/projects/glove/>



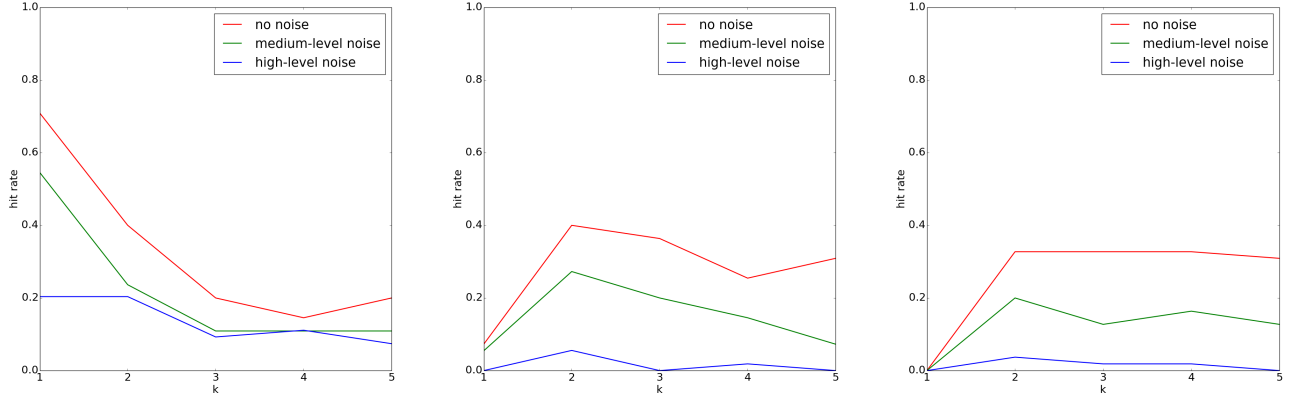
**Figure 3: Relationship between anonymity and reconstructability under different levels of added noise and no distractor terms (left: no-noise, middle: medium-level of noise, and right: high-level of noise).**



**Figure 4: Relationship between anonymity and reconstructability under different levels of added noise and with 20 distractor terms (left: no-noise, middle: medium-level of noise, and right: high-level of noise).**



**Figure 5: Relationship between anonymity and reconstructability under different levels of added noise and with 40 distractor terms (left: no-noise, middle: medium-level of noise, and right: high-level of noise).**



**Figure 6: Hit rates for the  $k$ -means clustering attacks for increasing number of clusters ( $k$ ) and distractor terms. (left: no distractors, middle: 20 distractors, and right: 40 distractors). In each figure, we show results for three levels of added noise.**

From Figure 6 left we see that the hit rate is high when we do not use any distractor terms. In this case, the set of candidate terms consists purely of related terms  $X_i$ . We see that if we cluster all the related terms into one cluster ( $k = 1$ ) we can easily pick the original query  $A$  by measuring the similarity to the centroid of the cluster. The hit rate drops when we add noise to the word embeddings, but even with the highest level of noise, we see that it is possible to discover the original query in 19% of the time. However, the hit rate drops significantly for all levels of noise when we add distractor terms as shown in the middle and right plots in Figure 6. Further results are presented in the SI. These results show the importance of using distractor terms.

Hit rate is maximum when we set  $k = 2$ , which is an ideal choice for the number of clusters considering the fact that we have two groups of terms (related terms and distractors) among the candidates. Increasing  $k$  also increases the possibility of further splitting the related terms into multiple clusters thereby decreasing the probability of discovering the original query from a single cluster. We see that hit rates under no or medium levels of noise drops when we increase the number of distractor terms from 20 to 40, but the effect on high-level noise added candidates is less prominent. This result suggests that we could increase the number of distractor terms while keeping the level of noise to a minimum.

We show the related and distractor terms for two example queries, *Hitler*, in Table 1 and *mass murder*, in Table 2. We see terms that are related to the original queries can be accurately identified from the word embeddings. Moreover, by adding a high-level of noise to the embeddings, we can generate distractor terms that are sufficiently further from the original queries. Consequently, we see that both anonymity and reconstructability is relatively high for the examples. Interestingly, the clustering attack is unable to discover the original queries, irrespective of the number of clusters produced.

## 5.2 Trade-off between Reconstructability and the Hit Rate in Clustering Attacks

If the keywords (related and distractor terms) sent to the search engine are related to the original user query then the search engine

Query	Hitler
noise	high-level
related terms	nazi, führer, gun, wehrmacht, guns, nra, pistol, bullets
distractors	schenectady, fairfield, columbia, hanover, lafayette, bronx, evansville, youngstown, tallahassee, alexandria, northampton
anonymity	0.867
reconstructability	0.831
Clustering Attack	Revealed Query
k=1	motgomery
k=2	albany, george
k=3	smith, albany
k=4	smith, fresno
k=5	rifle, albany

**Table 1: Relevant and distractor terms for the query *Hitler*. Both anonymity and reconstructability is high for this query with even with a small number of distractor terms. Clustering attack with different number of clusters ( $k$ ) does not reveal the original query.**

Query	mass murder
noise	high-level
related terms	terrorism, killed, wrath, full-grown
distractors	roselle, morristown, rockville, schenectady, utica, albany, ashland, hartford, salem, columbus
anonymity	0.789
reconstructability	0.747
Clustering Attack	Revealed Query
k=1	richmond
k=2	fremont, death
k=4	pasadena, words
k=4	pasadena, words
k=5	pasadena, anderson

**Table 2: Relevant and distractor terms for the query *mass murder* with 10 distractor terms. We see that the query or its two tokens are not revealed by the clustering attacks with different  $k$  values.**

will be able to return relevant search results that we can use to reconstruct the search results for the original user query. However, this will also increase the risk that the search engine can guess the original user query using some attacking method such as  $k$ -means clustering described in the paper. Hit rate was defined as the ratio of the user queries correctly predicted by the clustering attack and is a measure of the robustness of the proposed query anonymisation method against  $k$ -means clustering attacks. Therefore, a natural question is *what is the relationship between the reconstructability and the hit rate*.

To empirically study the relationship between reconstructability and hit rate, we conduct the following experiment. We randomly select 109 user-queries and add Gaussian noise with zero-mean and standard deviations 0 (no noise), 0.6, 1.0, 1.4 and 1.8. In each case, we vary the number of distractor terms 0-120. We then apply  $k$ -means clustering attacks with  $k$  values of 1, 2, 3, 4 and 5. In total, for a fixed  $k$ -value and the number of distractor terms, we have 545 clustering attacks. To make the evaluation more conservative, in this section we consider the terms in the vocabulary closest to the respective centroids in all clusters and not only the most coherent one as specified in Section 5.1. If the original query matches any of those  $k$  terms, we consider it to be a hit. We randomly sample data points from even intervals of reconstructability values and plot in Figure 7.

We see a positive relationship between the reconstructability and the hit rate in all figures. This indicates a trade-off between the reconstructability and the hit rate, which shows that if we try to increase the reconstructability by selecting more relevant keywords to the original user-query, then it simultaneously increases the risk of the search engine discovering the original user-query via a clustering attack. Moreover, we see that when we increase the number of distractor terms the hit rate drops for the same value of reconstructability. This result shows that in order to overcome the trade off between the reconstructability and the hit rate we can simply increase the number of distractor terms, thereby making the query anonymisation method more robust against clustering attacks. Moreover, the drop due to distractor terms is more prominent for the  $k = 1$  attacks when we have distractor terms compared to that when we do not have distractor terms. This is because both related and distractor terms will be contained in this single cluster from which it is difficult to guess the original user-query. Therefore, multiple clusters are required for a successful  $k$ -means clustering attack.

Overall, the hit rate drops in the order of  $k = 5$ ,  $k = 3$  and  $k = 2$  when we increase the number of distractor terms. This result suggests that if one wants to increase the hit rate, then an effective strategy is to increase the number of clusters because we consider it to be a hit if the user-query is found via any of the clusters. Intuitively, if we form more clusters and pick all terms from the vocabulary closest to any one of the centroids, then the likelihood of predicting the original user-query increases with the number of clusters formed. However, in practice, we will need to further select one term from all the clusters. Nevertheless, we can consider the hit rate obtained in this manner to be a more conservative estimate, whereas in reality it will be less and therefore be more robust against attacks.

## 6 HUMAN EVALUATION

The goal of our work was to anonymise queries sent to search engines such that the search engine cannot guess the user's information intent. However, it is an interesting question whether a human, not a search engine, could guess the original query from the set of related and distractor terms suggested by the proposed method. This can be seen as an upper baseline for anonymisation. To empirically evaluate the difficulty for humans to predict the original query, we devise a *query prediction game*, where a group of human subjects are required to predict the original query from the related and distractor terms suggested by the proposed method.

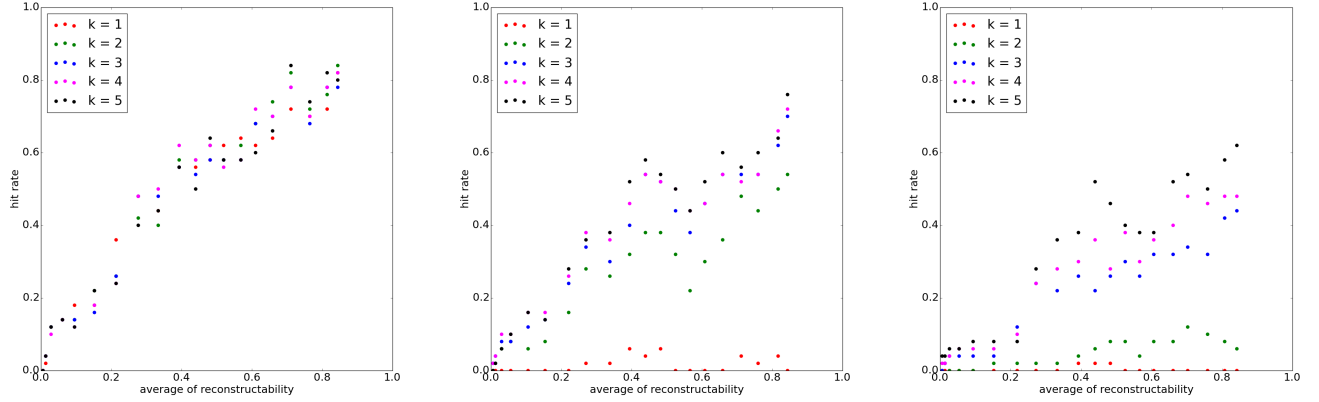
The query prediction game is conducted in two stages. In the first stage, we randomly shuffle the related and distractor term sets extracted by the proposed method for a hidden query. The human subject is unaware of which of the terms are related to the original user-query and which are distractors. A human subject has a single guess to predict the user-query and wins only if the original query is correctly predicted. If the human subject fails at this first step, then we remove all distractor terms and display only the related terms to the human subject. The human subject then has a second chance to predict the original query from the related set of terms. If the human subject correctly predicts the original query in the second stage, we consider it to be a winning case. Otherwise, the current round of the game is terminated and the next set of terms are shown to the human subject. Winning rate is defined as the number of games won by the human subjects, where the original user query was correctly predicted.

Figures 8 and 9 show the winning rates for the first and second stages of the query prediction game against the anonymity of the queries. All queries selected for the prediction game have reconstructability scores greater than 0.3, which indicates that the search results for the original query can be accurately reconstructed from the related and distractor terms shown to the human subjects. We see that the winning rate for the first stage is lower than the second stage, indicating that it is significantly easier for human subjects to guess the original query when we have removed the distractor terms. This result suggests that the distractor terms found by the proposed method can distract not only search engines but also humans. From Figure 9 we see that there is a gradual negative correlation between hit rate and anonymity. This implies that more anonymous the terms are, it becomes difficult for the human subjects to predict the original query, which is a desirable property for a query anonymisation method.

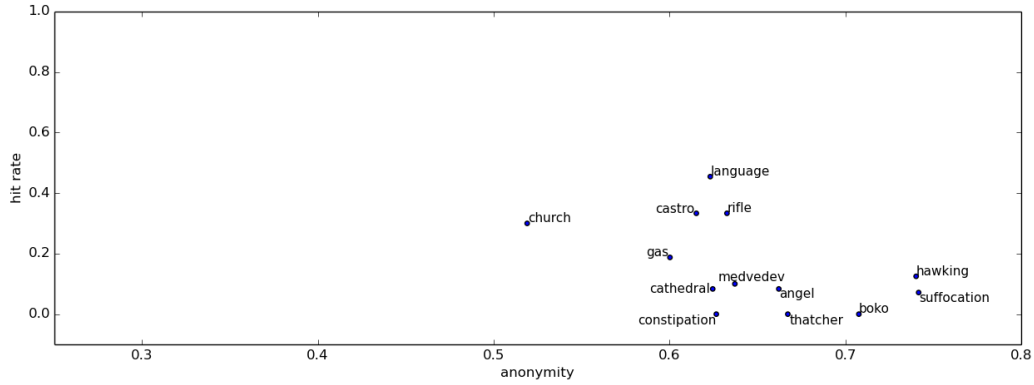
## 7 RELATED WORK

Our work is closely related to several different research fields such as query anonymisation, user profile unlinking, user unidentifiability and Private Information Retrieval (PIR). Traditionally, information retrieval systems such as Web search engines have been primarily text-based interfaces where users enter keywords that describe their information need and the search engine returns relevant documents to the users as the search results. The queries entered by the Web search engine users often reveal intimate private information about the users that they would not like to be known to the general public. One of the early incidents of query logs leaking private information in the public domain is the AOL's release of query log data in

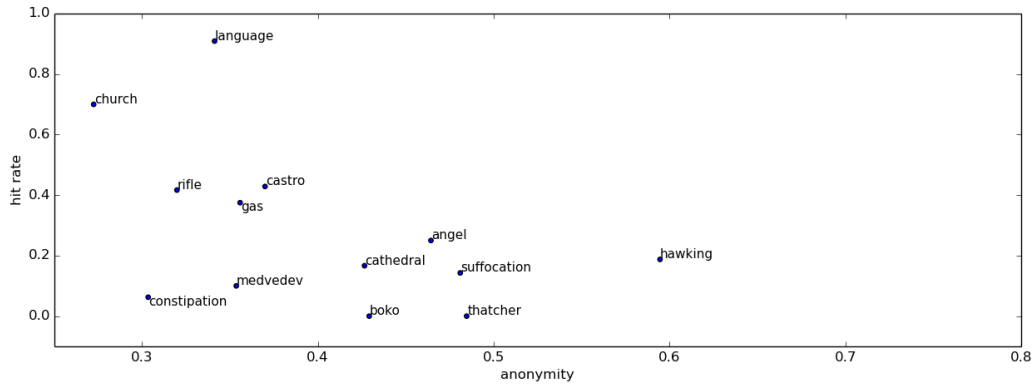




**Figure 7: Hit-rate shown against reconstructability for  $k$ -means attacks with left no distractor terms, middle 60 distractor terms and right 120 distractor terms.**



**Figure 8: Winning rate vs. anonymity for the first-stage of the query prediction game**



**Figure 9: Winning rate vs. anonymity for the second-stage of the query prediction game**

2006.<sup>4</sup> AOL released 20 million search queries from over 600K users, representing about 1.5% of AOL's search data from March, April and May of 2006. The data contained the query, session id, anonymised user id, and the rank and domain of the clicked result.

<sup>4</sup><https://tinyurl.com/y9qx9ufz>

Despite the user ids being anonymised, various private information about the users could be easily triangulated from the released queries, which resulted after nine days AOL to issue an apology, remove the website and terminate a number of employees responsible for the decision, including the CTO. Following this incident various methods have been proposed to anonymise user queries such as

token-based hashing [12] and query-log bundling [11]. However, in these approaches anonymisation happens only at the Web search engine's side without any intervention by the users, and the users must trust the good intentions of the search engine with respect to the user privacy. Moreover, [12] showed that hashing alone *does not* guarantee user privacy.

Accessing Web search engines via an anonymised proxy server such as the onion routing [7], TOR [6], Dissent [4] or RAC [17] is a popular strategy employed by common users. The goal is to prevent the search engine link the queries issues by a user to his or her user profile. Unfortunately, hiding the IP address of a user alone does not guarantee privacy as evident from the AOL incident in 2006, which already had user IPs replaced by random ids in the released query logs. In order to completely unlink their profiles, users must continuously change the proxy servers used and clean caches in the form of cookies and embedded javascript, which is a cumbersome process.

A complementary approach for ensuring the unidentifiability of users by the search engines is to issue a mixture of noisy or unrelated keywords alongside the keywords that describe the information need of the users. Several browser add-ons that automatically append unrelated fake terms have been developed such as TrackMetNot [10], OptimiseGoogle, Google Privacy and Private Web Search tool [26]. Although this approach is similar to our proposal to append user queries with distractor terms previously proposed methods have relied on pre-compiled ontologies [23] such as the WordNet or queries issued by other users shared via a peer network. Such approaches have scalability issues because most named entities that appear in search queries do not appear in the WordNet and it is unlikely that users would openly share their keywords to be used by their peers.

The goal in Private Information Retrieval [29] is to retrieve data from a database without revealing the query but only some encrypted or obfuscated version of it [3, 18]. For example, in homomorphic encryption-based methods the user (client) submits encrypted keywords and the search engine (server) performs a blinded lookup and returns the results again in an encrypted form, which can then be decrypted by the user. Embellishing queries with decoy terms further protects the privacy of the users. However, unlike our proposed method, PIR methods assume search engines to accommodate the client side encryption methods, which is a critical limitation because modern commercial Web search engines do not allow this.

Although we considered text-based queries, the proposed framework is not limited to text-based information retrieval, but can be easily extended to other types of search platforms. For example, in the case of voice-based information retrieval [13], we can use the spectrum of the voice input and add some noise to it such as white noise to find the distractors. Likewise, in image-based information retrieval, we can add noise to the image embedding produced by, for example, a convolutional neural network-based filter [8, 14, 28]. We plan to extend the proposed method to other types of information retrieval tasks in the future.

## 8 CONCLUSION

We proposed a method to anonymise queries sent to a Web search engine by decomposing the query into a set of related terms and a set of distractor terms. We then reconstruct the search results for

the original query using the search results we obtain for the related terms, discarding the search results for the distractor terms. We theoretically studied the relationship between the anonymity and the reconstructability obtained using the proposed method under different noise levels. We empirically showed that the proposed anonymisation method is robust against a  $k$ -means clustering attack. Moreover, a human evaluation task, implemented as a query prediction game, showed that it is even difficult for humans to predict the original query from the anonymisation produced by the proposed method.

## REFERENCES

- [1] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A Latent Variable Model Approach to PMI-based Word Embeddings. *Transactions of the Association for Computational Linguistics* 4 (2016), 385–399.
- [2] Claudio Carpineto and Giovanni Romano. 2012. A Survey of Automatic Query Expansion in Information Retrieval. *Journal of ACL Computing Surveys* 44, 1 (2012), 1 – 50.
- [3] B Chor, N. Gilboa, and M. Naor. 1997. *Private information retrieval by keywords*. Technical Report. Department of Computer Science, Technion, Israel Institute of Technology.
- [4] H. Corrigan-Gibbs and B. Ford. 2010. Dissent: accountable anonymous group messaging. In *Proc. of CCS*.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL-HLT*.
- [6] R. Dingledine, N. Mathewson, and P. Syverson. 2004. TOR: The second generation onion router. In *Proc. of the Usenix Security Symposium*.
- [7] D. Goldschlag, M. Reed, and P. Syverson. 1999. Onion routing. *Commun. ACM* 42, 2 (1999).
- [8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proc. of International Conference on Learning Representations (ICLR)*.
- [9] Chuan He, Cong Wang, Yi-Xin Zhong, and Rui-fan Li. 2008. A Survey on Learning to Rank. In *Proc. of the 7th Intl. Conf. on Machine Learning and Cybernetics*. 1734 – 1739.
- [10] D. C. Howe and H. Nissenbaum. 2009. TrackMeNot: Resisting surveillance in web search. Lessons from the Identity Train: Anonymity, Privacy and Identity in a Networked Society.
- [11] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. 2008. Vanity Fair: Privacy in Querylog Bundles. In *Proc. of CIKM*. 853–862.
- [12] Ravi Kumar, Jasmine Novak, Bo Pang, and Andrew Tomkins. 2007. On Anonymizing Query Logs via Token-based Hashing. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. ACM, New York, NY, USA, 629–638. <https://doi.org/10.1145/1242572.1242657>
- [13] Lin-shan Lee and Yi-cheng Pan. 2009. Voice-based Information Retrieval – how far are we from the text-based information retrieval?. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. 26–43.
- [14] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. 2017. NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles. (07 2017). arXiv:1707.03501
- [15] Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representation in vector space, In *Proc. of International Conference on Learning Representations. CoRR*.
- [16] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (November 1995), 39 – 41.
- [17] S. Ben Mokhtar, G. Berthou, A. Diarra, V. Quéma, and A. Shoker. 2013. RAC: A freerider-resilient scalable, anonymous communication protocol. In *Proc. of ICDCS*.
- [18] R Ostrovsky and W. I. Skeith. 2007. A survey of single-database PIR: techniques and applications. In *Proc. of Public Key Cryptography (PKC)*, Vol. 4450. 393–411.
- [19] Marius Pasca. 2007. Organizing and searching the world wide web of facts-step two: Harnessing the Wisdom of the Crowds. In *WWW 2007*. 101–110.
- [20] Marius Pasca. 2014. Queries as a Source of Lexicalized Commonsense Knowledge. In *Proc. of EMNLP*. 1081–1091.
- [21] Jeffery Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: global vectors for word representation. In *Proc. of EMNLP*. 1532–1543.
- [22] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL-HLT*. arXiv:arXiv:1802.05365
- [23] Albin Petit, Sonia Ben Mokhtar, Lionel Brunie, and Harald Kosch. 2014. Towards efficient and accurate privacy preserving web search. *Proceedings of the 9th Workshop on Middleware for Next Generation Internet Computing - MWANG '14*

- (2014). <https://doi.org/10.1145/2676733.2676734>
- [24] Matthew Richardson. 2008. Learning about the world through long term query logs. *ACM Transactions on the Web* 2, 4 (2008).
  - [25] Eldar Sadikov, Jayant Madhavan, Lu Wang, and Alon Halevy. 2010. Clustering Query Refinements by User Intent. In *WWW 2010*. 841–850.
  - [26] F. Saint-Jean, A. Johnson, D. Boneh, and J. Feigenbaum. 2007. Private web search. In *Proc. of ACM Workshop on Privacy in Electronic Society*. 84–90.
  - [27] Rodrygo L. T. Santos, Craig Macdonald, and Idah Ounis. 2010. Exploiting Query Reformulations for Web Search Result Diversification. In *WWW 2010*. 881–890.
  - [28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. *arXiv* (2014).
  - [29] Sergey Yekhanin. 2010. Private Information Retrieval. *Commun. ACM* 53, 4 (April 2010), 68–73. <https://doi.org/10.1145/1721654.1721674>