k − RDF-Neighbourhood Anonymity: Combining Structural and Attribute-Based Anonymisation for Linked Data

Benjamin Heitmann[1,2], Felix Hermsen[1], and Stefan Decker[1,2]

[1] Informatik 5 – Information Systems
RWTH Aachen University, Ahornstr. 55, 52056 Aachen, Germany
lastname@dbis.rwth-aachen.de
[2] Fraunhofer Institute for Applied Information Technology FIT
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
firstname.lastname@fit.fraunhofer.de

Abstract. We provide a new way for anonymising a heterogeneous graph containing personal identifiable information. The anonymisation algorithm is called k − RDF-neighbourhood anonymity, because it changes the one hoop neighbourhood of at least k persons inside an RDF graph so that they cannot be distinguished. This enhances the privacy of persons represented in the graph. Our approach allows us to control the loss of information in different parts of the graph to adjust the trade-off between full privacy and data utility. In particular, we can control the weighting of subgraphs induced by individual properties as well as the weighting of attributes represented by literals. To the best of our knowledge, our approach is the first one which considers all subgraphs of an RDF graph at the same time during the anonymisation, instead of projecting the graph into its subgraphs, anonymising each subgraph separately, and them merging the anonymised subgraphs again. In addition, our approach allows partial anonymisation of RDF graphs, for use cases in which only specific entity types need to be protected. We conducted an experiment, which shows that the overall loss of information after anonymising the graph is smaller, if the anonymisation takes all parts of the graph into account, instead of focusing only on either the structure or only on the attributes of the graph.

1 Introduction

The Resource Description Framework as part of Linked Data can be used to build an open accessible heterogeneous RDF graph. Therefore, if this graph contains personally identifiable information it is sometimes desirable to anonymize only a part of the graph. Hence, we aim to enable both complete anonymisation of an RDF graph as well as partial anonymisation.

For example, imagine a heterogeneous RDF graph, which describes people and their political views. This graph could be of public benefit, because it could be used to automate the process of calculating statistics. For instance, what percentage of people in a certain area agrees with gun ownership or gay marriage.

However, this is sensitive information about an individual and should not be public knowledge. As a consequence individuals in such a data set should be anonymized so that their privacy is preserved. However, such a heterogeneous graph may also contain data about persons of public interest, for instance public figures, celebrities or politicians. Especially the political views of politicians should be of public knowledge so that regular people can look them up and make decisions in the context of elections, e.g. based on shared values and political views. Thus, in such use cases it is desirable to only anonymize persons who are not persons of public interest, but to not anonymise politicians and celebrities.

In addition, the anonymization of heterogeneous graphs is more challenging than the anonymization of homogeneous graphs, because heterogeneous graphs have more than one edge and vertex type.

Our goal is to develop and implement an anonymization approach to anonymize heterogeneous `RDF` graphs containing personal identifiable information which allows controlling the trade-off between data utility and full privacy. This trade-off will be measured through information loss when comparing the original graph and the anonymised graph.

We evaluate our approach experimentally by anonymising data with our algorithm in two different ways: first we set the weights in order to prioritise preservation of the social network structure within the data set, which follows the state-of-the-art approaches for anonymising RDF data in the same way as social networking data. Then we repeat the anonymisation step, by setting the weights of our algorithm to prioritise all subgraphs as well as all literal properties equally. We then compare the anonymisation results by measuring the loss of information for each part of the graph as well as on overall. This allows us to measure if our algorithm can be tuned in regards to the loss of utility both for parts of the RDF graph as well as for the overall loss of utility.

In this paper, we present abbreviated versions of the algorithms which are described in full in [1].

The remainder of this paper is structured as follows: In Section 2 we provide an overview of the background in regards to anonymisation by summarising the state of the art for anonymisation of tabular data and heterogeneous graphs such as RDF and Linked Data. Then in Section 3 we describe our approach for anonymising heterogeneous graphs by combining structural and attribute-based anonymisation, which we call $k - $ `RDF`-neighbourhood anonymity. Next, we describe the evaluation of our approach in Section 4. We explain the experiment design, the data generation, and our results from measuring the loss of information. Then we discuss our approach and our evaluation results in Section 5, and conclude the paper in Section 6 with a summary and a list of future work.

2 Background

In this section, we provide an overview of the background in regards to anonymisation by summarising the state of the art for anonymisation of tabular data and heterogeneous graphs such as RDF and Linked Data.

Overall related research stresses the importance and the growing demand for privacy. However, it also lays out that sharing datasets containing personal information can be of benefit. Therefore, it is important to protect privacy through anonymization.

In particular research in the area of anonymizing homogeneous social network graphs shows the feasibility of anonymization techniques and that there are several ways how to do it.

Researchers concerned about the privacy in the Semantic Web community noticed that the benefits of Linked Data and Semantic Web technologies, such as portability and linkability, also make de-anonymisation of individuals easier. The reason for this is that de-anonymization attacks match and link data, which is one of the core principles of Linked Data. Nevertheless, research in the field of anonymizing heterogeneous `RDF` graphs suggest that ideas used for the anonymization of homogeneous social network graphs like k-anonymity can be transferred to `RDF` graphs.

2.1 k-anonymity

Anonymization emerged as relational datasets had to be shared and the privacy of entities inside those datasets had to to be protected [2]. The most well-known approach to protect against the identifiability threat is the idea of k-anonymity [2]. The input is a single table, where a row represents an entity and a column an attribute. For instance, a relational health care dataset could be composed out of 4 attributes: the name, the age, the living place and the disease of a person.

A dataset is said to be k-anonymous, if and only if each row is indistinguishable from at least $k-1$ other rows. For this, a row is divided into 3 parts, into its identifiable attributes (IA), quasi identifiable attributes (QIA) and sensitive attributes (SA) [2].

IAs are suppressed (removed), since they can identify the entity directly. IAs are for example the name or the social security number of a person.

QIAs are generalized. Generalization is the process to blur QIAs of a row such that the data has still some utility left. This is done because a QIA does not directly identify an entity. However, QIAs can identify an entity when they are combined.

For instance, a study conducted in the United States of America found that 87% of the population is uniquely identified by the combination of three attributes [2]. These attributes are the gender, the date of birth and the 5-digit zip-code. That is why it is important to generalize QIAs and it is not sufficient to just suppress identifiable attributes.

Approaches which implement k-anonymity leave sensitive attributes unmodified during the anonymisation process, because these attributes are important for the dataset to be released in the first place. For example, if a medical institution wants to conduct research on disease patterns in certain areas the attribute "disease" is very important. However, the identity of a person is not important. Therefore, hospitals or other institutions releasing such information to the public for research purposes, will leave the sensitive attributes unchanged.

2.2 Anonymization of Homogeneous Graphs

[3] gives a broad overview about available research and divides research regarding the topic of this section into two categories. The first category is called clustering based approaches and the second one is called graph modification approaches.

The idea of **clustering based approaches** is to summarize vertices to one (clustered) super vertex. These approaches generally will shrink the graph but also remove all structure between them [3]. Zheleva et al. [4] propose two clustering based anonymization approaches. However, the goal of these approaches is not to prevent re-identification but to protect sensitive relationships. The corresponding threats are called link-ability and disclosure of information [5]. A combination has been proposed by Campan and Truta [6].

In contrast, **graph modification approaches** rely on the deletion and addition of nodes or edges to change the structure of a graph to satisfy different conditions. [3] divides graph modification approaches into randomized graph construction approaches, such as k-degree anonymity [7], and greedy graph modification approaches, such as k-neighborhood anonymity [8].

Liu et al. [7] proposed k-degree anonymity. This approach shows one possible way of transfering k-anonymity from tabular data to graph data. It is a randomized graph construction approach, which aims to modify the graph, such that every vertex has the same degree as at least $k-1$ other vertices. This is achieved by computing the smallest number of edges which have to be added so that k vertices have the same degree. Afterwards that number of edges is randomly added to satisfy the condition. This approach minimizes the loss of information.

Zhou et al. [8] proposed k-neighborhood anonymity, which is satisfied, when at least k nodes have the same one-hop neighborhood. Their approach is categorized as a greedy graph modification approach, since they greedily add edges to similar neighborhoods to make them the same. In addition to this, they search for similar neighborhoods to reduce information loss.

2.3 Anonymization of Heterogeneous Graphs

The research paper from Radulovic et al. [9] proposes an anonymization framework for the Resource Description framework. They state that since an increasing amount of `RDF` data is shared, privacy issues are expected. Moreover, there are already existing privacy concerns. For example, the paper "Privacy Concerns of FOAF-Based Linked Data" by Decker et al. [10] discusses that spam e-mails could be generated out of an FOAF based dataset.

Radulovic et al. [9] proposed an approach called k-`RDF`anonymity. The approach is based on k-anonymity. The idea is that a resource can not be distinguished from at least $k-1$ other resources. The work does not state any pseudo code and does not refer to any implementations. However, they describe different anonymization operations, which can be used to implement k-`RDF`anonymity [9]. The approach targets a subset of resources. They refer to them as entities of interest (EOI). They state that anonymizing heterogeneous `RDF` graphs are more complex than homogeneous graphs, because information regarding the EOI can

occur in the different places and forms [9]. They can occur in the subject URI, in the data type value, subject URI, object property value and more complex scenarios [9]. Furthermore, they describe one information loss metric similar to the one used for k-neighbourhood anonymity [8].

A different approach for anonymising heterogeneous graph data, called k-neighbourhood anonymity, is proposed by Zhuyan Lin [11]. We will describe it in more detail in section 3.

Rachapali et al. [12] extended SPARQL with a new query form called SANITIZE, which consists of a set of sanitization operations and used to sanitize an RDF graph. The provided language helps in implementing privacy features for RDF data during SPARQL queries.

3 $k-$ RDF-neighbourhood anonymity

As we combine and extend ideas from both Zhou et al. [8] and Radulovic et al. [9], we call our approach k-RDF-Neighborhood Anonymity. The idea of our approach is that the one-hop neighbourhood of a resource which is going to be anonymized is indistinguishable from the one-hop neighbourhood of at least $k-1$ other resources.

On a conceptual level, our goal is to develop a greedy heterogeneous graph modification approach. We choose a graph modification approach instead of a graph clustering approach, because a graph clustering approaches remove all information between clustered nodes [3]. However, we aim to preserve as much structure of the graph after the anonymisation as possible, as our objective is to release a graph instead of just graph metrics.

We identified that most anonymization approaches try to reduce the loss of information in only one structural part. However, in general social network users also have attributes. We call this the attribute part. One part of research ignores that user inside a social network have attributes. The other part of research generalizes them based on the previously computed structural anonymization.

In order to reduce the computational complexity and thus the total run-time of our algorithm, we restrict the anonymization to the one-hop neighbourhood of resources with type `foaf:Person`. This idea was proposed by Zhou et al. [8], who also introduced the idea of anonymization weighting parameters.

However, their approach is not designed for heterogeneous graphs and modifications are required. To make these modifications, we include ideas presented by Radulovic et al. [9]. They proposed k-RDF-Anonymity and presented different anonymization operations, which can be used to implement the idea of k-anonymity for heterogeneous graphs.

3.1 Overview of algorithm

The steps of our $k-$ RDF-neighbourhood anonymity algorithm are as follows:

1. Compute a minimal string for each EOI representing its one hoop neighbourhood.

2. Search in a greedy way for k similar strings under the objective of reducing loss of information.
   (a) Take the one-hop neighborhood of the un-anonymised vertex with the highest degree.
   (b) Compute a similarity value between this neighbourhood and all other one hoop neighborhoods which are not marked as anonymized.
   (c) Select the k-1 most similar neighbourhoods.
3. Modify the graph by changing all k identified one-hop neighbourhoods so they are indistinguishable within the graph.
4. Repeat 2 and 3 until all EOIs are marked as anonymized.

In the following we provide abbreviated descriptions of the anonymisation criterion, of the neighbourhood code computation and similarity, and of the graph modification. The full details can be found in [1].

3.2 Anonymisation criterion

We say a heterogeneous `RDF` graph $G_{he} = (V, E, f_v, f_e)$ is anonymized, if all entities of interest (`EOI`) are anonymized. We define an `EOI` to be a vertex associated with the type `FOAF:person`.

In addition to that, we say an `EOI` is anonymized, if there are at least $k-1$ other `EOIs` having the same one-hop neighbourhood, where $k \in \mathbb{N}$. We call the set of `EOIs` having the same one-hop neighbourhood an equivalence class.

The one-hop neighbourhood of a vertex $v \in V$ is defined as follows:

Definition 1 *The heterogeneous graph $G_{1hop}(v) = (V_{1hop}, E_{1hop}, f_v, f_e)$ is said to be the one-hop neighbourhood of a vertex $v \in V$ of $G_{he} = (V, E, f_v, f_e)$, if $G_{1hop}(v)$ is a subgraph of G_{he}. Specifically, let V_{1hop} be the set of all vertices $v_o \in V$ that are directly connected with v including v itself. Moreover, let E_{1hop} be the set of edges connecting v with $v_o \in V_{1hop}$. In addition to that, E_{1hop} also contains all edges, which connect vertices in V_{1hop} among one another.*

3.3 Neighbourhood code computation

Since our approach frequently compares one-hop neighborhoods, it would be inefficient to frequently conduct isomorphism tests. The reason for that is that there is no known polynomial time algorithm for the general graph isomorphism problem [8]. Therefore, we generate a string we call the full neighborhood code ($FNHC$), which we use to compare neighborhoods in a more efficient way.

To compute the $FNHC$ we divide the input graph $G_{1hop}(v) = (V_{1hop}, E_{1hop}, f_v, f_e)$ into three neighborhoods: (1) the attribute neighbourhood NHC_a; (2) the human collaboration neighbourhood NHC_{hc}; and (3) the social neighbourhood NHC_s. For each neighborhood we generate a neighborhood code (NHC). We put them together in a defined order to obtain the $FNHC$. For the attribute and collaboration neighbourhood the NHC is the result of lexicographically ordering all strings representing entities in the neighbourhood of the person and then concatenating them.

For the NHC representing the social neighbourhood, we use the approach developed by Xifeng et al. [13]. Their idea is to calculate all possible *depth first search trees* (DFS-trees) for each subgraph G_{s_i} and vertex $v \in V$. After that they are encoded and the minimum DFS code is searched. The ordered concatenation of the minimum depth-first-search-tree codes of each subgraph results in the string NHC_s. Two social network neighborhoods are isomorphic, if their NHC_s codes are the same according to [13] . This also applies to the $FNHC$, which is the concatenation of NHC_a,NHC_{hc} and NHC_s.

3.4 Similarity of neighbourhood codes

The similarity value between two one-hop neighborhood graphs is the sum of the similarity of each substring of the full neighborhood code NHC.
$sim(FNHC(G_{1hop}(v)), FNHC(G_{1hop}(u)))$ is the weighted sum of four different similarity values.

The similarity value is composed of four parts. The **age similarity**, the **based_near similarity**, **the project similarity** and the **knows similarity**. Each individual similarity function takes the intrinsic properties of the strings representing that specific neighbourhood into account.

The combined similarity value for two one-hop neighborhoods is computed by the following function:

$$sim(FNHC(G_{1hop}(v)), FNHC(G_{1hop}(u))) = \alpha \cdot sim_{age} + \beta \cdot sim_{based\_near} + \gamma \cdot sim_{project} + \delta \cdot sim_{knows}$$

3.5 Modification of one-hop neighbourhood

The graph modification algorithm generalizes n vertices by changing the heterogeneous graph $G_{he} = (V, E, f_v, f_e)$ so that vertices $T = \{v_1, ..., v_n\} \subseteq V$ form an equivalence class. The graph modification algorithm starts by changing G_{he} so that v_1 and v_2 are generalized and create an equivalence class $EQ = \{v_1, v_2\}$. After this the equivalence class is expanded by one additional vertex $v \in T$, where $v \notin EQ$. Next we change G_{he} so that v_1 and v are generalized. This process generally changes $G_{1hop}(v_1)$. To adjust to the new changes we have to update all other one-hop neighborhoods of vertices $v_i \in EQ$, where $v_i \neq v_1$, such that v_1 and v_i are generalized. We expand the equivalence class until each vertex of T has been added, such that $EQ = \{v_1, ..., v_n\}$. If $n \geq k$ we can call each vertex in EQ anonymized.

We say $v, u \in T$ with $v \neq u$ are generalized, if the `age` attributes, the `based_near` attributes, the `current_project` structures and the `knows` structures are generalized so that $FNHC(G_{1hop}(v)) = FNHC(G_{1hop}(u))$.

Generalization of RDF graphs should not add false information to the graph. Hence, we delete edges and not add edges as it is proposed by [8]. Moreover, this idea is in compliance with the open world assumption. The open world assumption is that a missing statement can also be true, if it is not contained in the dataset.

We assume the deleted edge labeled with `FOAF:knows` connects vertices $a, b \in V$. Moreover, let $G_{1hop}(a)$ be the one-hop neighborhood of a and $G_{1hop}(b)$ the one-hop neighborhood of b. Then we have to update all full neighborhood codes of $V(G_{1hop}(a)) \cup V(G_{1hop}(b))$.

3.6 Pseudo-code for whole algorithm

In this section we describe how the previously explained steps are combined in order to anonymize a heterogeneous `RDF` graph $G_{he} = (V, E, f_v, f_e)$. The inputs for this algorithm are: an equivalence class size parameter $k \in \mathbb{N}$, weighting coefficients $\alpha, \beta, \gamma, \delta \in \mathbb{R}_+$, anonymization percentage parameter $p \in [0, 1]$ and a heterogeneous `RDF` graph G_{he}.

Algorithm 1 Pseudo-code for the Anonymization algorithm: k-`RDF`-Anonymity

procedure $k\_RDF\_Anonymity(G_{he}, k, p, \alpha, \beta, \gamma, \delta)$
  $EOIs \leftarrow$ select all vertices with type `FOAF:persons` of G_{he} and randomly select p percent.
  for all $v \in EOIs$ **do**
    compute $FNHC(G_{1hop}(v))$
    remove identifiable attributes
    change the label of v such that the name in it is removed.
  end for
  while EOIs.$size > 2(k-1)$ **do**
    target $\leftarrow v \in EOIs$ with $G_{1hop}(v)$ such that there is no $G_{1hop}(v')$ with higher degree
    EOIs.$remove(v)$
    sim_list $\leftarrow$ empty list
    for all $u \in EOI's$ **do**
      sim_list $\leftarrow$ add $sim(FNHC(G_{1hop}(target)), FNHC(G_{1hop}(u)))$
    end for
    best_fitting_vertices $\leftarrow$ empty list
    for all $i \in \{1, ..., (k-1)\}$ **do**
      u $\leftarrow$ find minimum in sim_list and save associated vertex u
      best_fitting_vertices $\leftarrow u$
      sim_list.$remove(u)$
    end for
    $graph\_modification(best\_fitting\_vertices \cup target)$
    EOIs.$remove(best\_fitting\_vertices)$
  end while
  $graph\_modification(EOI's)$
  $return\ G_{he}$
end procedure

The algorithm first selects all vertices representing persons of G_{he} and then selects a subset of them based on the input p. This simulates that there are some individuals, who want all their private information released. All other vertices

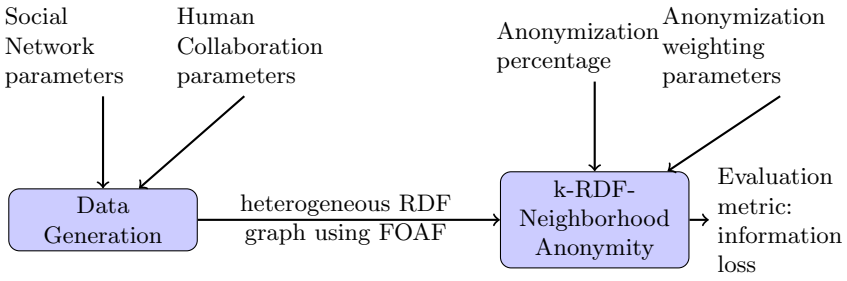

Fig. 1. Overview of evaluation

should be anonymized. We call these vertices entities of interest (`EOI`). The next step is to compute for each vertex v in `EOI` a full neighborhood code as explained in section 5.1. We also remove all identifiable attributes v has. Moreover, there might be some identifiable information in the resource `URI`. Therefore, we change the label of the vertex so that this kind of identifiable information is removed. After this we greedily select k vertices and anonymize them.

We select the vertex $target \in$ `EOIs`, which has the biggest one-hop neighborhood, because this is the neighborhood, which could introduce the highest amount of information loss. We compute a similarity value between $target$ and all other vertices in `EOI` and select $k-1$ vertices with the smallest similarity value, because they reduce the information loss. We save them in a list called sim_list. The corresponding decision problem is the substructure similarity search problem and proven to be NP-hard in [8]. The last step is to generalize the set of vertices $sim\_list \cup target$ with the graph_modification algorithm. We repeat this until each vertex in `EOI` is anonymized.

4 Evaluation

We evaluate our approach with an experiment using synthetic data. Then we anonymise the data set using our algorithm in two different ways. We then compare the anonymisation results by measuring the loss of information for each part of the graph as well as on overall. This allows us to measure if our algorithm can be tuned in regards to the loss of utility both for parts of the RDF graph as well as for the overall loss of utility. Figure 1 shows an overview of our evaluation.

For the evaluation, we will compare two different sets of weights for the algorithm. The first set of weights prioritises the preservation of the social network structure within the data set, which follows the state-of-the-art approaches for anonymising RDF data in the same way as social networking data. Then we repeat the anonymisation step, by setting the weights of our algorithm to prioritise all subgraphs equally as well as all literal properties.

4.1 Data generation

We evaluate our anonymization algorithm using a heterogeneous `RDF` graph using the Friend of a Friend (FOAF) vocabulary. We specifically use the FOAF

vocabulary, because a graph using FOAF links people and information together. It can be seen as a combination of social networks, representational networks and information networks [3]. In order to model a classical social networking graph, we use the property `foaf:knows` to represent edges in the social graph, and the class `foaf:Person` to represent vertices.

Unsuitability of BTC data: Our main objective is to develop an anonymization approach protecting the identity of a person in a heterogeneous `RDF` graph. Therefore, we looked for a graph containing personal identifiable information as well as social network connections between persons. The most promising dataset we found is the Billion Triples Challenge 2009 Dataset (BTC) [14]. The entire dataset contains at least 1 billion triples and is 17GB large. Due to restrictions on the available server infrastructure, we used the reduced version with a size of 2.3 GB.

However, during preparation of our experiment, we discovered that the BTC data set is unsuitable for our experiment. The BTC dataset contains approximately 290.000 resources defined as `foaf:Person` and 310.000 triples having the property `foaf:knows` as predicate. However, the majority existing instances of `foaf:Person` are not connected to each other. This results in a large number of disconnected graph components regarding the FOAF data contained in the BTC data. In addition, most of the `foaf:Person` instances on have either social connections or attributes, as contained in the BTC data. Taken together, both factors result in the BTC data being unsuitable for our experiment. In order to simulate anonymisation of social network data for a heterogeneous graph, we require one single, giant, connected component given by the social connections in addition attributes being present for a sufficient percentage of vertices in the connected component.

Therefore, as the BTC data set did not contain data fulfilling both of these requirements, we used synthetically generated data for our experiments.

Generation of synthetic data: For our evaluation, our approach is to generate a synthetic FOAF based dataset describing persons. In our dataset a person has a name, an age and a living place. The name serves as an identifiable attribute, while the other two serve as quasi identifiable attributes. Moreover, persons know each other. This forms a social network and is generated based on the social network parameters.

We use two different property types in our graph, which are `foaf:knows` and `foaf:currentProject`. This makes the RDF graph a heterogeneous graph in the formal sense.

In addition to that, persons are engaged in projects. A person can be engaged in multiple projects. This forms a bipartite graph between persons and projects. This graph is generated based on the human collaboration parameters.

The information network is a heterogeneous `RDF` graph using the geonames vocabulary representing city regions, districts and the federal states of Germany. This kind of information is used to give a person a living place and is important for the anonymization, since it represents a semantic hierarchy of places.

[3] http://xmlns.com/foaf/spec

Our data generation algorithm generates 2^n persons and assigns them a random real world name. Then we compute a social network among them, which consists of m edges. To achieve this, we use an algorithm called Recursive Matrix (R-MAT) [15]. In particular this algorithm uses four input variables, which can be adjusted to generate different types of networks. We adjusted them so that it computes a small world network having a power law vertex degree distribution. In addition to that, the R-Mat algorithm naturally calculates communities. In a community, which exactly consists of $2^{(n-rd)}$ persons, each person has the same living place and a random age based on a uniform distribution between 20 and 80. Furthermore, we generated a random bipartite graph between persons and projects. Each person is involved in at least min projects and at most max projects.

For our evaluation, we generated 10 different graphs for the anonymization phase. For each graph we generated $n = 2^8 = 256$ person resources and $m = 3 * 256 = 768$ `FOAF:knows` edges. The reason why these parameters are so small is high complexity the algorithm which computes the full neighbourhood code. In addition we generated $n_2 = 15$ different projects and associated a person with at least $min = 3$ projects and a maximum of $max = 7$ projects. We chose the recursion depth $rd = 5$, such that at least $2^3 = 8$ people are associated with the same living place. We saved each graph as a turtle file.

4.2 Parameters of our algorithm for the evaluation

As described in the listing of the algorithm in Section 3.6, our algorithm requires the following parameters: α as the weighting parameter for the age attribute similarity; β as the weighting parameter for location similarity; γ as the weighting parameter for the collaboration structure; and δ as the weighting parameter for the social network structure.

For the first evaluated version of our algorithm, we used the parameters ($\alpha = 0, \beta = 0, \gamma = 0, \delta = 1$). This parameterisation forces the algorithm to compute the anonymisation solely based on the social network structure, which in turn prioritises preservation of the social network structure, while changing as much within the collaboration structure and the attributes as required.

In contrast, the second evaluated version of our algorithm uses the parameters ($\alpha = 1, \beta = 1, \gamma = 1, \delta = 1$). This parameterisation requires the algorithm to give equal priority to all parts of the graph. So the social network structure is preserved equally well as the collaboration structure and the attributes.

Anonymisation percentage: We chose the parameter p to be 0.95 for all the anonymizations. The result is that $0.95 * 256 = 243$ individuals are anonymized and $0.05 * 256 = 13$ not. In addition to that, for each anonymization associated with the same graph we chose the same 243 resources with type `FOAF:person`.

4.3 Evaluation metric: Information loss

The loss of information is calculated by comparing the output of the anonymization algorithm $G_{he_{out}}$ with the input graph $G_{he_{in}}$. Let $G_{1hop}(v)$ be the one-

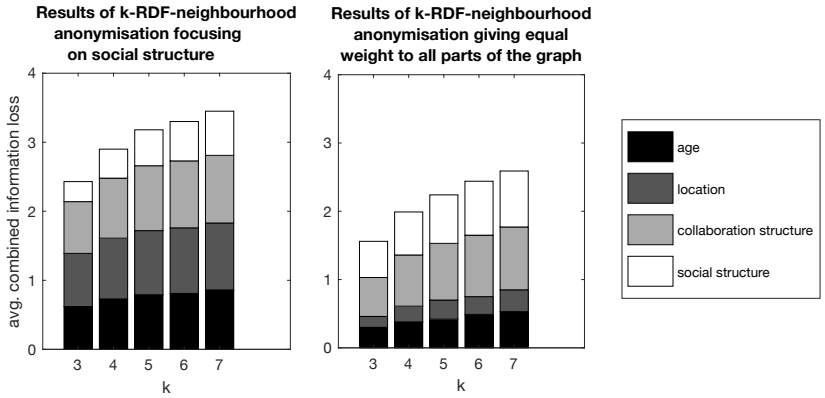

Fig. 2. Results of the evaluation: we compare two parameterisations of our $k-$ `RDF`-neighbourhood anonymity algorithm using information loss as a metric.

hop neighborhood of $v \in V(G_{he_{in}})$ and $G_{1hop}(u)$ the one-hop neighborhood of $u \in V(G_{he_{out}})$, where $l_v(v) = l_v(u)$. Moreover, the type of v and u is `FOAF:Person` and u has been anonymized.

Then the **information loss** is the similarity value between the $G_{1hop}(v)$ and $G_{1hop}(u)$. We sum all similarity values and divide the output by the number of vertices having type `FOAF:persons` in one graph.

However, we normalize the similarity value of the project similarity and the knows similarity so that they are not the number of removed edges. The project similarity is divided by the number of vertices in the project neighborhood of $G_{1hop}(v)$. The knows similarity is divided by the number of edges in the knows neighborhood of $G_{1hop}(v)$. This means the project similarity and the knows similarity are values between 0 and 1.

The final output is the sum of the average loss of information for each of the four parts of the graph, which results in a value between 0 and 4, where smaller is better.

4.4 Results of evaluation

Figure 2 presents the results of our evaluation. We compare two parameterisations of our $k-$ `RDF`-neighbourhood anonymity algorithm using information loss as a metric. On the right are the results of focusing on the preservation of the social network structure of the RDF graph. On the left are the results when all subgraphs and literal properties are weighted equally in regards to their preservation after anonymisation.

The x-axis shows values of k between 3 and 7, where k is the size of equivalence classes generated for the one-hop neighbourhoods in the equivalence class. All neighbourhoods within an equivalence class are indistinguishable from each other. The parameter k is an indicator of how strong the anonymization is. In the

context of our evaluation we expect larger values of k to perform worse than smaller values.

The y-axis shows the total combined average information loss over all parts of the graph. As there are four parts and as the maximum information loss per part is 1, the maximum combined information loss is 4. In addition, the individual information loss for each part is also show, as indicated by the colour for each part listed in the legend on the right. If the information loss value is 1, this means all information is lost, whereas 0 means no information is lost. Therefore smaller information loss is better in the context of our evaluation.

The results clearly show that the overall loss of information is smaller for the parameterisation of the algorithm which weights all parts of the graph equally for the purpose of data preservation during the anonymisation process. This can be seen by the generally lower combined average information loss on the right hand side of the diagram.

In addition, we can observe for both parameterisations of the algorithm, that the loss of information increases in a linear way to 1 with rising k. The loss of information is always smaller for the parameterisation with equal weight for all parts for the same value of k.

We stated previously that we expect, if the anonymization is concentrated solely on one part, the loss of information should be small in that part and high in every other part. This is clearly visible when looking at the results on the left hand side, which shows that the parameterisation of the algorithm which gives preference to the social structure generally reduces the information loss on that structure. The white parts at the top of the bars are much shorter than all other three parts of the bars.

Therefore, to have good data utility we propose that anonymization should not be concentrated solely on one part of the graph. To the best of our knowledge we are the first ones to make this statement and present experiment results supporting this statement.

Summary of evaluation: The evaluation of that data resulted in the knowledge that an anonymization, which solely concentrates on one part of the network, achieves low loss of information in that part but high loss of information in every other part. This decreases the data utility, because the overall loss of information is high. However, we observed that the overall loss of information is always smaller, if the anonymization is concentrated on multiple parts. This observation answers the research question, if anonymization should be based on solely one part or multiple parts.

5 Discussion

We will now discuss the differences of our anonymisation algorithm with regards to the algorithm on which it is based. In addition, we will discuss our preliminary results of evaluating our anonymisation algorithm by de-anonymising the results in order to measure how good the anonymisation was.

5.1 Extensions of Zhou et al.s approach

We introduced k-RDF-neighborhood anonymity, which is based on the approach proposed by Zhou et al. [8]. They proposed $k-$neighborhood anonymity. Moreover, our approach additionally uses ideas of research presented in the related work section.

The main extension of our approach in comparison to Zhou et al., is that our approach is designed to anonymize a *heterogeneous graph*. The approach of Zhou et al. is designed to anonymize a *homogeneous graph*. While we only show how to anonymise a heterogeneous RDF graph with two link types, our approach is inherently extensible to an unlimited number of link types. However, this will also increase the computational complexity and run time of executing the algorithm. Further differences are:

- Our approach assumes a directed graph, whereas Zhou et al. assume an undirected graph.
- Our approach is designed to do partial and full anonymization. Their approach is designed to do full anonymization.
- We delete edges and they add edges to satisfy the anonymization criteria. We note that deleting edges is more complicated, since the associated neighborhoods could be split. We do this because the idea of generalization is to not introduce false information. Therefore, to be consistent in that respect we delete edges. In addition it is in compliance with the open world assumption of RDF semantics.
- We assume that each vertex is uniquely identified by its vertex label. They assume that the vertex label is the quasi identifiable attribute. Their assumption reduces the complexity.

5.2 Preliminary results of de-anonymisation experiment

We performed an additional experiment in which we implemented a de-anonymisation algorithm in order to have an additional way to measure the performance of different parameterisations of our anonymisation algorithm.

De-anonymisation algorithms require so-called "background data" in order to re-identify the persons in an anonymised data set. In our case, we took the original input data for one iteration of our experiment, and removed a number of random edges and vertices. This modified data set then was used as the background data for the de-anonymisation algorithm to re-identify the persons in the partially anonymised output of our k-RDF-neighborhood anonymity evaluation.

In general, there are two families of de-anonymisation algorithms, which are (1) *guessing*-based and (2) *matching*-based de-anonymisation [16]. The later is in generally more powerful and provides better accuracy during the re-identification however it also requires solving more complex implementation issues in order to implement it [16] [17]. Therefore we used a guessing based de-anonymisation approach during our preliminary experiment. However, we only achieved unsatisfying re-identification rates using our guessing based de-anonymisation algorithm, and will revisit the topic in future work.

6 Conclusion and future work

In this paper we proposed an anonymization algorithm for heterogeneous RDF graphs containing personal identifiable information. It protects the identity of persons by generalizing the graph. More specifically, our greedy anonymization algorithm finds k similar one-hop neighborhoods of persons and changes them so that they cannot be distinguished. This step is repeated until each person, which wants to stay private, is anonymized. Our anonymization process results in the loss of information. The loss of information is measured by comparing the input graph with its anonymized version.

The difference between a heterogeneous graph and a homogeneous graph is that a heterogeneous graph consists of multiple edge types and vertex types. Therefore, the graph has multiple parts. For example a heterogeneous graph can contain a social network and a human collaboration network. Moreover, attributes of persons can be included in the same heterogeneous graph as well.

Our anonymization algorithm can concentrate solely on one part or multiple parts of the input graph. On which part or parts the algorithm is concentrated on we specify with weighting parameters. The result is that the loss of information drops in the parts that have been targeted for preservation.

We presented the results of an evaluation, which showed that our algorithm can be tuned to preserve the information of one or all parts of a heterogeneous RDF graph. Moreover, we observed in the experiment data that anonymizations, which are concentrated on solely one part of the input graph, achieve low loss of information in that part but high loss of information in every other part. This results in an overall high amount of loss of information.

In addition, we observed that the overall loss of information is lowered by weighting the preservation of multiple parts of the input graph highly. Therefore, we hypothesize that the anonymization of a heterogeneous graph should always be concentrated on multiple parts to achieve a good trade-off between data utility and loss of information. However, the priority could be to reduce the information loss in a specific part of a heterogeneous graph as much as possible with disregard to the overall loss of information. Therefore, an anonymization algorithm should have the ability to be adjusted for such different settings. Our anonymization algorithm allows for prioritisation of parts of the graph through setting its weighting parameters.

In addition, we showed that our algorithm is able to provide partial anonymisation of a graph as well.

For future work, we will investigate improvements to both our de-anonymisation algorithm implementation and our experiment design. In particular we are planning to implement a matching-based de-anonymisation algorithm in the future.

References

1. Hermsen, F.: Anonymization and De-Anonymization of Heterogeneous Graphs Containing Personal Identifiable Information. B.Sc. Thesis, to be published, RWTH Aachen University, Germany (2017)

2. Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05) (2002) 557–570
3. Zhou, B., Pei, J., Luk, W.: A Brief Survey on Anonymization Techniques for Privacy Preserving Publishing of Social Network Data. SIGKDD Explor. Newsl. **10**(2) (2008) 12–22
4. Zheleva, E., Getoor, L.: Preserving the Privacy of Sensitive Relationships in Graph Data. In Bonchi, F., Ferrari, E., Malin, B., Saygin, Y., eds.: Privacy, Security, and Trust in KDD: First ACM SIGKDD International Workshop, PinKDD 2007, Springer (2008) 153–171
5. Deng, M., Wuyts, K., Scandariato, E., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. In: Requirements Eng (2011) 16: 3. doi:10.1007/s00766-010-0115-7. (2011)
6. Campan, A., Truta, T.M.: A clustering approach for data and structural anonymity in social networks. In: Proceedings of the 2nd ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD'08), in Conjunction with KDD'08, Las Vegas, Nevada, USA. (2008)
7. Liu, K., Terzi, E.: Towards Identity Anonymization on Graphs. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. SIGMOD '08, New York, NY, USA, ACM (2008) 93–106
8. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE'08). (2008) 506–515
9. Radulovic, F., García-Castro, R., Gómez-Pérez, A.: Towards the Anonymisation of RDF Data. Technical report, KSI Research (2015)
10. Nasirifard, P., Hausenblas, M., Decker, S.: Privacy concerns of FOAF-based linked data. In: Trust and Privacy on the Social and Semantic Web Workshop (SPOT 09) at ESWC09, Heraklion, Greece. (2009)
11. Li, Z.: From Isomorphism-Based Security for Graphs to Semantics-Preservubg Security for the Resource Description Framework RDF. Master's thesis, University of Waterloo, Canada (2016)
12. Rachapalli, J., Khadilkar, V., Kantarcioglu, M., Thuraisingham, B.: Towards Fine Grained RDF Access Control. In: Proceedings of the 19th ACM Symposium on Access Control Models and Technologies. SACMAT '14, New York, NY, USA, ACM (2014) 165–176
13. Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: 2002 IEEE International Conference on Data Mining, 2002. Proceedings. (2002) 721–724
14. Harth, A.: Billion Triples Challenge Data Set, downloaded from http://km.aifb.kit.edu/projects/btc-2009/ (2009)
15. Chakrabarti, D., Zhan, Y., Faloutsos, C.: R-mat: A recursive model for graph mining. In: Proceedings of the SIAM International Conference on Data Mining. (2004) 442–446
16. Ding, X., Zhang, L., Wan, Z., Gu, M.: A Brief Survey on De-anonymization Attacks in Online Social Networks. In: International Conference on Computational Aspects of Social Networks. (2010) 611–615
17. Al-Azizy, D., Millard, D., Symeonidis, I., O'Hara, K., Shadbolt, N.: A literature survey and classifications on data deanonymisation. In: International Conference on Risks and Security of Internet and Systems, Springer (2016) 36–51