RiAiR: A Framework for Sensitive RDF Protection

Irvin Dongo[1,2] and Richard Chbeir[3]

[1]*Univ. Bordeaux, ESTIA, Bidart, France*
[2]*Universidad Católica San Pablo, Arequipa, Peru*
[3]*Univ Pau & Pays Adour, LIUPPA, EA3000, 64600, Anglet, France*
E-mail: i.dongoescalante@estia.fr; richard.chbeir@univ-pau.fr

Received 04 January 2019;
Accepted 06 March 2019

Abstract

The Semantic Web and the Linked Open Data (LOD) initiatives promote the integration and combination of RDF data on the Web. In some cases, data need to be analyzed and protected before publication in order to avoid the disclosure of sensitive information. However, existing RDF techniques do not ensure that sensitive information cannot be discovered since all RDF resources are linked in the Semantic Web and the combination of different datasets could produce or disclose unexpected sensitive information. In this context, we propose a framework, called *RiAiR*, which reduces the complexity of the RDF structure in order to decrease the interaction of the expert user for the classification of RDF data into identifiers, quasi-identifiers, etc. An intersection process suggests disclosure sources that can compromise the data. Moreover, by a generalization method, we decrease the connections among resources to comply with the main objectives of integration and combination of the Semantic Web. Results show a viability and high performance for a scenario where heterogeneous and linked datasets are present.

Keywords: RDF protection, Sensitive information, Semantic Web, Disclosure source.

Journal of Web Engineering, Vol. 18_1-3, 43–96.
doi: 10.13052/jwe1540-9589.18132

1 Introduction

With the advance of the Semantic Web and the Linked Open Data initiatives, more and more RDF documents are available on the Web. RDF describes resources as triples: $\langle \texttt{subject}, \texttt{predicate}, \texttt{object} \rangle$, where `subjects`, `predicates`, and `objects` are all resources identified by their IRIs. Objects can also be literals (e.g., a number, a string), which can be annotated with optional type information, called datatype. Since the last decade, RDF is attracting more and more people, and data is gathered and published by different sources (e.g., companies, governments) for many purposes such as statistics, testing, and research proposals. For instance, according to [21], more governments are becoming *e-governments*, since they are part of the LOD initiatives, providing their data to have a more flexible data integration, increasing the data quality, and offering new services. However, as more data is available, sensitive information (e.g., diseases, salaries, or bank accounts) could be sometimes provided or inferred leading to compromise the privacy of related entities (e.g., patients, users, companies).

Data can be analyzed and protected before being published on the Web [24, 41], or limited in access for queries over controlled scenarios [35, 48]. In this work, we only focus on the protection of RDF data, expressed as documents, by the analysis of the data before publication. A privacy protection of the RDF data is tricky, since the use of different published heterogeneous datasets could break some protection. For instance, the combination of well-known datasets as DBpedia and Enipedia[1] produces sensitive information of places of interest (e.g., schools, hospitals, production factories), regarding their proximity to nuclear power plants (high contamination resource).

According to [41], anonymization is one common and widely adopted technique for sensitive data protection that has been

[1] *Enipedia* is a dataset containing data related to the production of energy and its applications. The information available on Enipedia is provided by governments, which support the LOD. `http://enipedia.tudelft.nl`

successfully applied in practice. It consists on protecting the entities of interest by removing or modifying identifiable information to make them anonymous before publication, while keeping the utility of the data. This latter is modified according to certain criteria of the existing values (e.g., taxonomies, ranges) to satisfy some conditions of anonymity (e.g., k-anonymity[2], l-diversity[3]). To apply anonymization, it is necessary to identify and classify the data (see D in Figure 1) into: (i) *main entities*, which are the entities of interest, and (ii) *related data* that is directly or indirectly associated to the main entities and can compromise their privacy. The related data can also be classified as [6]: (i) *Identifiers*, data that directly identify a main entity (e.g., security social number); (ii) *Quasi-identifiers*, data that can be used to link with other data to identify a main entity (e.g., birthday, postal code, gender); (iii) *Sensitive information*, which is the data that compromise a main entity (e.g., diseases); and (iv) *Unsensitive information* that does not have a particular role or impact.

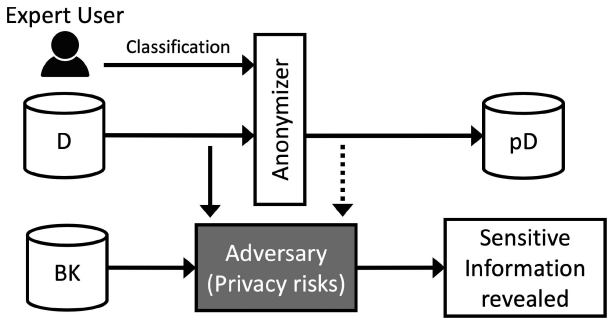

Figure 1 Anonymization framework inspired from [29]; D is the data to be published, BK is the Background Knowledge; and pD the protected data obtained by the anonymization process, considering the classification made by the expert user.

[2]k-anonymity is one of the most used common condition, that consists on making entities undistinguished from at least $k-1$ other entities, because they have similar information [43].

[3]l-diversity is an extension of the k-anonymity model that protects the corresponding sensitive values within a homogeneous group.

A classification, which is performed by an expert user (see Expert User in Figure 1) who knows previously the data and is responsible of protecting model, is based on predefined assumptions about how an adversary can take advantage over these data. These assumptions are called *Background Knowledge*. The background knowledge (see BK in Figure 1) is the information related to the published data, which can be used by adversaries to discover sensitive information of the main entities. Due to the huge complexity of the RDF structure, a classification requires a high interaction of the expert user. Moreover, all RDF's elements can be considered as main entities, and they can also be classified into identifiers, quasi-identifiers, sensitive information, etc., making the RDF protection complex.

Works on RDF anonymization are limited [24, 41]. They mainly apply generalization and suppression operations over taxonomies (each RDF's element has a defined taxonomy) to anonymize the RDF document. Defined areas (neighborhood) are also provided [24], where anonymization properties as k-anonymity are satisfied. Various anonymous RDF documents are generated by the combination of all values from the taxonomies and a measure is required to choose the best option. However, the exhaustive method to select the best anonymous RDF document makes these approaches unsuitable for complex cases, since a greater quantity of values to take into account, needs a more elaborate anonymization process (more possible solutions).

Since RDF forms a directed, labeled graph structure with data, where the edges (`predicates`) represent the named link between two resources, represented by the graph nodes (`subjects` and `objects`) [36], databases and graphs anonymization techniques could be applied, but they are limited and inappropriate for privacy protection in the Semantic Web, as we detail in Section 3.

Thus, in the context of RDF data, the following limitations are identified:

1. RDF anonymization techniques are limited and designed for a particular and ideal scenario, which is inappropriate when having several linked heterogeneous datasets [4, 24, 41, 48];
2. The non-consideration of IRIs as external and reachable resources makes the current RDF solutions unsuitable for protection on the

Web, since other available resources could link or infer sensitive information;
3. The presence and consideration of resources (IRIs and Blank nodes), which are a fundamental part of the RDF data, makes the database oriented methods [26, 30, 31, 33, 44] unsustainable for a large quantity of resources due to the number of JOIN functions needed to satisfy the existing normalized models;
4. Graph anonymization techniques assume simple, undirected and unlabeled graphs [5, 7, 8, 22, 27, 28, 52, 53, 56]; thus, the reduction of complexity of the RDF structure to a simple graph is necessary for the application of graph solutions, but inappropriate for the Semantic Web, since properties and semantic relations among resources would be ignored;
5. The complexity of the RDF structure requires a high interaction of the expert user to identify and select the RDF's elements to be protected (main entities), and the ones related to the main entities (identifiers, quasi-identifiers, sensitive information, and unsensitive information); and
6. Approaches based on conceptual RDF representations are needed in order to provide more general solutions that can be serialized later on different formats (e.g., RDF/XML, Turtle, N3, JsonLD).

To overcome these limitations, we propose a framework, called *RiAiR* (Reduction, Intersection, and Anonymization in RDF) , which is independent of the serialization formats and providers. The proposal is designed for RDF documents, considering their elements (IRIs, blank nodes, literals) and the scenario, where a huge quantity of information is available. The complexity of the RDF structure is reduced to make possible the task of classification and to suggest potential disclosure sources to the expert user, decreasing his interaction. Moreover, by a generalization method, we reduce the connections among datasets, preserving the main objectives of the Semantic Web (integration and combination), and protecting the sensitive information at the same time.

We validated our anonymization approach through several experiments. We evaluated the viability and the performance of the proposal with respect to the related work. Results show a real viability of our

approach for linked heterogeneous datasets and a high performance of the anonymization process of quadratic order with respect to the triples of the the data to be published (n) and the ones from the background knowledge (m) ($i.e., O(n^2 + m^2)$).

The paper is organized as follows. Section 2 presents a motivating scenario to illustrate the disclosure of sensitive information on the Web. Section 3 surveys the related literature. Terminologies and concepts are presented in Section 4. Section 5 describes our approach. Section 6 shows the experiments to evaluate the viability and performance of our approach. Finally, we present our conclusions in Section 7.

2 Motivating Scenario

The goal of the Semantic Web is to publish datasets, mainly as RDF, describing and combining resources on the Web for an open access. The datasets are usually treated and protected before being published; however, sensitive information could be deduced using related information available from other datasets. To illustrate this, let's consider a scenario in which a data manager *X* works for a government to publish a *dataset A*, related to energy production and its applications, on the Web[4].

An extract of the *dataset A* to be published is shown in Table 1.

Figure 2 shows the schema of the *dataset A* to be published. Note that the properties `prop:Latitude`, `prop:Longitude`, `rdfs:label`, and `cat:radioactive` define values, while the properties `prop:City`, `prop:Country`, and `cat:Fuel_type` define resources.

Table 1 An example of the data extracted from *Enipedia* dataset

Nº.	cat:Fuel_-type	cat:radio-active	rdfs:label	prop:City (rdfs:label)	prop:Country (rdfs:label)	prop:lat.	prop:long.
1	art:Nuclear	true	Hartlepool	Hartlepool Cleveland	United Kingdom	54.6824	–1.2166
2	art:Nuclear	true	Limerick	Pottstown	United States	40.2257	–75.5866
3	art:Nuclear	true	Neckar	Neckarwestheim	Germany	49.0411	9.1780
4	art:Nuclear	true	Beaver Valley	Shippingport	United States	40.6219	–80.4336

[4]The example provided uses an extract from *Enipedia* dataset.

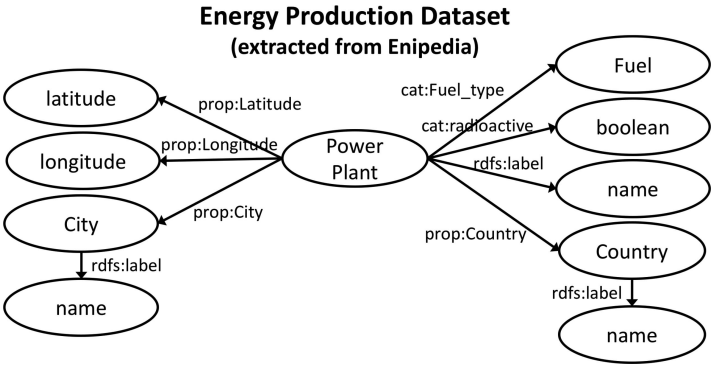

Figure 2 Structure of the data extracted of the enipedia dataset.

Table 2 Some places of interest available in the *DBpedia* dataset

Nº.	rdf:type	rdfs:label	prop:lat.	prop:long.
1	dbo:School	Hartlepool College of Further Education	54.6839	-1.2109
2	dbo:School	English Martyrs School and Sixth Form College	54.6754	-1.2362
3	dbo:School	Coventry Christian Schools	40.2505	-75.5930
4	dbo:School	HÃ¶lderlin-Gymnasium Lauffen am Neckar	49.0704	9.1394
5	dbo:School	Pennsylvania Cyber Charter School	40.6385	-80.4549

As a data manager, X should pay attention about the side effect of publishing the *dataset A* on the Web, since it can produce sensitive information for entities already published. For instance, *DBpedia*[5], which is a linked open dataset extracted from Wikipedia, can be used as background knowledge in order to discover sensitive information related to places of interest. This dataset can be easily connected by the use of properties, such as `prop:Latitude` and `prop:Longitude` present in the *dataset A* as well. Table 2 shows some places of interest available in the *DBpedia* dataset.

By the intersection among coordinates (`prop:Latitude` and `prop:Longitude`) of nuclear power plants (*dataset A*) and the ones of places of interest (*dataset DBpedia*), one can easily identify their proximity in a defined `Region`. A `Region` is an area obtained by the maximum distance between a nuclear power plant and a place of interest. The following SPARQL query produces the intersection

[5]DBpedia does not contain sensitive information, since all data correspond mainly to well-known entities (e.g., places, governments, actors, singers).

between the *dataset A* to be published and the *dataset DBpedia*. Note that a `Region` of 100 km was used to obtain the results.

```
SELECT DISTINCT
?Place ?g bif:st_distance(?g,bif:st_point(".$long.",".$lat."))
AS ?distance
FROM
<http://dbpedia.org> WHERE {?p rdfs:label ?Place ;
geo:geometry ?g ; rdf:type dbo:School .
FILTER
(bif:st_intersects (?g, bif:st_point (".$long.", ".$lat."), 100)
&& (lang(?Place) = \"en\"))}
ORDER BY ASC(?distance)
```

Table 3 is the result of the intersection between the *dataset A* and *dataset DBpedia*. It shows in row 1 that a *school* is less than 500 meters distance from a power nuclear plant in United Kingdom. It also shows which hospitals, universities, and any other crowded places are close to power nuclear plants in a defined area. One can even identify which are the dirtiest power nuclear plants (`prop:Carbonemissions`) and the places next to them. If this combined information is available on the Web, it can be misused against the nuclear power plants to stop their production and management, and even against the places of interest near to them.

Figure 3 illustrates graphically the intersection between *dataset DBpedia* and the *dataset A*. The resource `Region` links `School`, `University`, `Hospital` and `Power Plant` resources.

To protect the *dataset A* to be published, *X* needs to identify and classify the data, according to the assumptions of how an adversary can obtain or produce sensitive information, using the background knowledge, as follows. *The information of a* `Power Plant` *resource*

Table 3 Some places of interest next to Nuclear Power Plants

Nuclear PowerPlant	City	Country	School	Distance (Km)
Hartlepool	Hartlepool Cleveland	United Kingdom	Hartlepool College of Further Education	0.40244
Hartlepool	Hartlepool Cleveland	United Kingdom	English Martyrs School and Sixth Form College	1.48812
Beaver Valley	Shippingport	United States	Pennsylvania Cyber Charter School	2.5761
Limerick	Pottstown	United States	Coventry Christian Schools	2.81988
Neckar	Neckarwestheim	Germany	Hölderlin-Gymnasium Lauffen am Neckar	4.2998

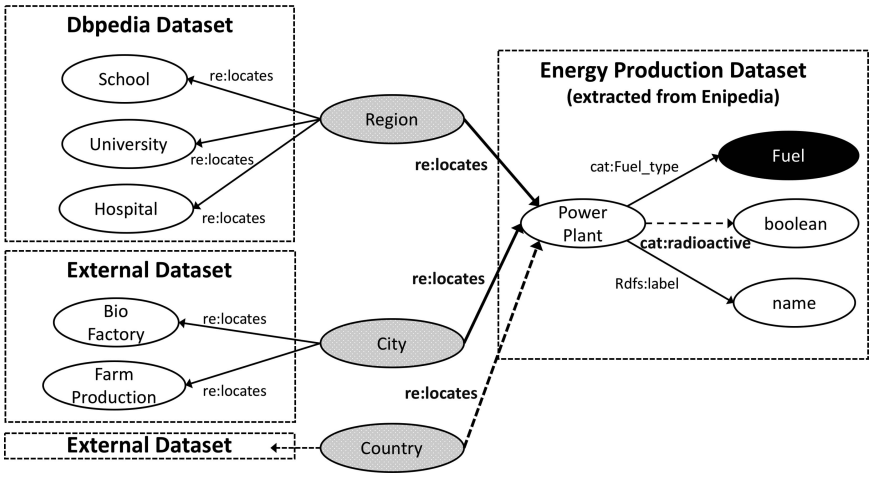

Figure 3 Intersection between *energy production* dataset and other datasets.

of type nuclear (`art:Nuclear`*) is sensitive, if there is at least a place of interest (e.g.,* `School`*) in a defined* `Region`[6].

- *Keys*: *(Identifiers/Quasi-Identifiers)*: Properties `prop:Longitude` and `prop:Latitude` are keys since both values indicate the position of a `Power Plant`, which belongs to a defined `Region`.
- *Sensitive Information*: A resource `dbo:School` and its properties are sensitive information, since they define the places of interest.
- *Unsensitive Information*: Other values and properties, which are not considered in the previous types, are unsensitive information.

Once X has established the classification, a protection technique based on this classification, should be used to protect the disclosure of sensitive information. Thus, the following challenges are defined in this study.

- Provide an easy classification of the RDF data (keys, sensitive information and unsensitive information);
- A similarity measure able to evaluate the intersection between the data to be published and the background knowledge, to suggest disclosure sources; and

[6]Considering only *DBpedia* dataset as external related information (background knowledge).

- Select the most appropriate protection taking into account the complexity of the RDF data and the objectives of the Semantic Web.

Our contribution in this study is as follows:

- A general framework designed for RDF documents, independent of the serialization formats, in a scenario where linked and heterogeneous resources are presented; i.e., the Web;
    1. A method to reduce the complexity of the RDF structure of the data to be published, simplifying the task of analysis, performed by the expert user;
    2. A method to suggest disclosure sources to the expert user, based on node similarity, reducing the task of data classification; and
    3. An anonymization operation, based on a generalization method, to decrease the relations among resources from different datasets, to preserve the main objectives of integration and combination of the Semantic Web.

The following section presents the related work of RDF anonymization.

3 Related Work

In this work, we focus on anonymization techniques as a solution to protect the sensitive information since it has been widely adopted for sensitive data protection [41]. To the best of our knowledge, works on RDF document anonymization are limited [24, 37–41, 48]; however, due to the particularity of the RDF data, other domains where anonymization has been extensively studied could be applied, such as: databases [16, 26, 30, 31, 44, 51] and graphs [5, 7, 8, 22, 27, 52, 56]. To evaluate and classify the existing works, we identified the following criteria of comparison according to the challenges and objectives of this work:

1. *The complexity of the data*, which should be aligned with the one of RDF structure, considering heterogeneous nodes and relations, increasing the expressibility and difficulty of the representation;

2. *The type of classification method* for identifiers, quasi-identifiers, sensitive and un-sensitive information due to the high quantity of entities, properties and values available on the Web, making difficult the task of the expert user; and
3. *The conditions of anonymity* that are proposed in the current proposals to identify the most appropriate ones for the Semantic Web.

Following sections describe the RDF, databases, and graph approaches in the context of anonymization.

3.1 RDF Document Anonymization

For RDF documents, the authors in [41] provide an overview of RDF's elements over the role in anonymization (e.g., explicit identifiers, quasi-identifiers, sensitive data). They propose a framework to anonymize RDF documents, which satisfies the k-anonymity condition. They consider the use of taxonomies for values and relations (each type of value and relation has its own taxonomy). Generalization and suppression operations are applied over these taxonomies to anonymize the RDF document. Once the operations are applied, several anonymous RDF documents are produced by the use of all value combinations from the taxonomies. A measure for anonymous solutions that satisfied the k-anonymity condition, is proposed to select the best option. In [24], the authors extend the previous work defining an area (neighborhood), where the k-anonymity condition is satisfied. The exhaustive method to select the best option makes these approaches unsuitable for complex cases, since a greater quantity of values to take into account, needs a more elaborate anonymization process (more possible solutions). Moreover, the authors assume a classification of the data provided by the model and they do not specify how this classification was performed.

Additionally, there are some works on the context of statistical queries [4, 48] based on grouping operators (e.g., SUM, AVG, MAX) and others based on expert-defined sanitization queries [37–40] to remove identifiers, but we only focus on the protection of RDF documents.

3.2 Database Anonymization

In some cases when one has small RDF data, a common practice can be to convert the RDF to a structured dataset as tables to reuse existing techniques. Anonymization in databases has been extensively studied and many works are available in the literature. One of the most used work is proposed in [42], the authors define a condition, called *k-anonymity*, where an entity cannot be identified, since there is at least $k-1$ other similar entities. However, the problem of satisfying the k-anonymity condition is NP-hard, producing different studies where the complexity and an efficient solution are addressed. For instance, to anonymize the data, the authors in [33] apply techniques based on neural networks, the authors in [2] apply genetic algorithms, while in [33] the authors use matching learning. Non-perturbative operations, such as generalization and suppression methods, where data is modified according to certain criteria of the existing values (e.g., taxonomies, ranges), are mainly used to satisfy the k-anonymity condition [3]. Other studies use *perturbative* operations, such as Micro-aggregation/clustering methods, where the entity values are replaced or modified by the centroid of the clusters, adding in some cases new entities to satisfy conditions of anonymization in each cluster [47, 55].

According to [30], k-anonymity condition does not protect the sensitive values, since k similar entities can have the same sensitive information, which is the one required by the adversary. For that, the authors in [30] extend the k-anonymity condition considering a diversity (l) of sensitive values for each set of similar entities (l-diversity). However, the disclosure is still possible due to the attribute distribution of the dataset. The authors in [26] propose a condition where the distribution of each sensitive attribute should be close/similar to the whole attribute distribution in the dataset (t-closeness). Other studies extend the previous mentioned conditions to address particular assumptions of the background knowledge. The authors in [31] propose a (k,T)-anonymization model over spatial and temporal dimensions. Other works apply the conditions of anonymity to different values as the authors [44] do, where l-diversity condition is satisfied by the sensitive information as well. The l-diversity condition is extended

in the clustering proposal work [51], defining a (k, l, θ)-diversity model, which takes into account the cluster size, the distinct sensitive attribute values, and the privacy preserving degree of the model. An improvement of certain conditions is made for special scenarios; for instance, the authors in [19] divide numerical sensitive values into several levels, getting a better protection for numerical values. Also, properties of the data such as utility, value distribution, etc., are considered to propose anonymization models. The work in [16] takes into account the association between quasi-identifiers and the sensitive information as a criterion to control the use of generalization hierarchy. Some semantic features are added in recent works. The authors in [34] provide a (l, d)-semantic diversity model based on a clustering method. They analyze the distance among sensitive values (d) to consider more actual diversity. According to [45], a value can be quasi-identifier and sensitive information at the same time, proposing a method that can treat "sensitive quasi-identifier" and satisfying the conditions of l-diversity and t-closeness.

Differential Privacy as k-anonymity is another well-used technique to provide privacy. The authors in [13] propose a perturbation method for true answer of a database query by the addition of a small amount of distributed random noise. This method is extended by other authors as in [23], where they improve the accuracy of a general class of histogram queries while satisfying differential privacy. The work in [32] is a non-interactive setting model, generalizing probabilistically the raw data and adding noise to guarantee differential privacy. Other studies are focused on the privacy of anonymized datasets, since a dataset, in the context of databases, can be affected by updating and removing operations, which can expose the sensitive information. The authors in [46] propose an architecture which protects the main entities for databases that require removing operations frequently. They apply generalization operations based on hierarchies (non-perturbative method). The model satisfies k-anonymity condition; however, the architecture needs to verify the anonymous data for each new deleting request in order to protect the privacy of the original datasets. A centralized scenario is required to apply this proposal.

Works on database anonymization approaches that satisfy k-anonymity and its variations, assume that the classification of data into identifiers, quasi-identifiers, sensitive and un-sensitive information is provided by a user expert, who knows the data, focusing mainly on the method to satisfy the conditions of anonymization. In the Semantic Web, it is unable to understand the detailed characteristics of external datasets, and assume all the background knowledge possessed by adversaries. Moreover, as more information is involve, more complex is the task of converting the RDF data to a structured normalized model, since a high granularity (many tables) is produced due to the use of IRIs, acting as foreign-keys.

Following section describes the works related to graph anonymization.

3.3 Graph Anonymization

RDF data can be represented as a graph structure, having labeled-nodes, and directed and labeled-edges. In the literature, there are several works in the context of social media, where the authors assume a simple network as undirected, node-unlabeled and edge-unlabeled structure [7, 27, 52] (see Group 9 in Table 4). These works focus on the privacy through the number of edges among nodes, since an adversary can have the information about the relations, which can be the only one with a particular number (k-degree condition). The work in [7] proposes a greedy algorithm to satisfy the k-degree by partitioning all nodes to n clusters. Each cluster becomes uniform with respect to the quasi-identifier attributes and the quasi-identifier relationship (generalization). To choose the best n values, two criteria are taken into account: (i) each cluster has to contain at least k nodes and (ii) minimize the information loss of the data. The authors in [27] propose an algorithm to satisfy the k-anonymity condition over the number of edges of each node. They also rename the k-anonymity as k-degree condition. The proposal consists in two steps: (i) Degree Anonymization, where a degree sequence of the graph (descending order) is generated to group similar nodes with the same degree and (ii) Graph construction, where an algorithm decides among which nodes a new edge is added according to satisfy the k-degree condition. In [52], the authors anonymize a graph by adding random edges. They provide an analysis on the spectrum

Table 4 Related Work Classification

G	Work	Requirements		
		Conditions of Anonymity	**Complexity of data**	**Classification Method**
1	[41]	k-anonymity	RDF	Manual (I, QI, SI, USI)
2	[24]	k-anonymity neighborhood	RDF	Manual (I, QI, SI, USI)
3	[48]	Differential privacy	RDF	Manual (SI)
4	[4]	Differential privacy	RDF	Manual (SI)
5	[33]	k-anonymity	Structured data	Manual (I, QI, SI, USI)
6	[26, 30, 31, 44]	k-anonymity and variations	Structured data	Manual (I, QI, SI, USI)
7	[13]	Differential privacy	Structured data	Manual SI
8	[23, 32, 46]	Differential privacy and variations	Structured data	Manual (SI)
9	[7, 27, 52]	k-degree	Undirected, node-unlabeled, edge-unlabeled	Manual (I, QI, SI, USI)
10	[5, 8, 22, 56]	k-degree	Undirected, node-labeled, edge-unlabeled	Manual (I, QI, SI, USI)
11	[28]	k-degree	Undirected, node-labeled, edge-labeled (weight)	Manual (I, QI, SI, USI)
12	[53]	k-degree l-diversity	Undirected, node-labeled, edge-unlabeled	Manual (I, QI, SI, USI)
13	[37, 38] [39, 40]	Sanitization	RDF	Manual (I, QI, Si, USI)
14	**Our proposal**	**Intersection**	**RDF**	**Automatic (I, QI)**

of the graph to measure the impact of the anonymization solution. The spectrum is directly related to the topological properties such as diameter, presence of cohesive clusters, long paths and bottlenecks, and randomness of the graph. Works in this group only take into account the number of relations as a condition of anonymity (k-degree), but in a scenario where a diversity of nodes is present, the number of operations to satisfy the k-degree condition increases exponentially. Moreover, diversity of edges values is not analyzed and the authors assume that the classification of the data is provided by the expert user.

Other works manage more complex graphs by assuming labeled-node structure as in [5, 8, 22, 56] (see Group 10 in Table 4). The authors in [5] demonstrate assuming several attacks that removing identifiers and renaming the nodes in an arbitrary manner, from a social graph, is an ineffective anonymization mechanism. Walk-based attacks are able to compromise the privacy for modest numbers of node (around 90%); thus, it has been proven for the authors that removing identifiers of the data is not a well protection. The authors in [8] assume that the adversary knows only degree-based information, which is the number of relations (edges) that has each node. To anonymize the graphs, they add new nodes instead of edges, since they affirm that *"introducing new nodes does not necessarily have an adverse effect. To the contrary, adding new nodes with similar properties could better preserve aggregate measures than will distorting the existing nodes"*. To satisfy the k-anonymity condition, an algorithm following four steps is provided: (i) Optimally partition degree sequence (descending order), (ii) Augment graph with new dummy nodes, (iii) Connect original graph nodes to new dummy nodes, and (iv) Insert inter-dummy-node edges to anonymize dummies. In [22], the authors propose an anonymization technique that protects against re-identification by generalizing the input graph. They generalize the graph by grouping nodes into partitions, and then publishing the number of nodes in each partition, along with the density of edges that exist within an across partitions. To preserve the privacy of individuals, which are represented as nodes in a social network, the authors in [56] assume that an adversary may have the background knowledge about the neighborhood of some target individuals. Two properties are taking into account: (i) node degree in power law distribution [14] and (ii) small-world phenomenon [50] to ensure a low loss of data. They greedily organize nodes into groups and anonymize the neighborhoods of nodes in the same group to satisfy the k-anonymity condition.

Works in this group have the same drawbacks as the previous one, which are related to the modeling of social graphs as a simple structure (even if the graph is node-labeled), and the assumption of the classification, which is provided by the expert user.

The authors in [28] work also on the context of social networks by ensuring the privacy of main entities, which are the nodes in the

graph (see Group 11 in Table 4). They consider a weight over edges, since it can represent affinity among two nodes, frequency among two persons, or similarity between two organizations. They propose a Gaussian Randomization Multiplication strategy due to its simple implementation in practice and responds to the dynamic-evolution nature of social networks, since it is very hard and costly to collect the information in advance in a huge and dynamic scenario. This work represents in a better way the scenario present in the Semantic Web. However, edges-labeled are reduced to values and they are not considered as reachable resources which can be used to disclosure the sensitive information. Also, this work assumes that the classification of the data is provided by user expert.

Another work is presented in [53] (see Group 12 in Table 4), the authors assume a more complex graph than the previous described groups. In fact, in addition of the node degree, they also assume the values of the nodes as sensitive data. They propose a framework, which satisfies k-anonymity and l-diversity conditions. They generate a sequence of 3-tuples (id, node-degree, and its respective sensitive value). A grouping algorithm is applied over the list to group similar triples, following certain criteria to satisfy the conditions of anonymization (k-anonymity and l-diversity). The sequence is called *KDLD sequence*, when all the defined conditions are satisfied. From the KDLD sequence, the graph is rebuilt. Then, they propose a graph construction technique adding nodes to preserve utilities of the original graph. Two key properties are considered: (i) Add as few noise edges as possible; (ii) Change the distance between nodes as less as possible.

In general, graph anonymization approaches assume a simple structure of the data as an undirected and unlabeled-edge social media graph. Also, k-degree is a one of the common conditions of anonymity used for the authors; however, considering a diversity of nodes as in RDF and using the existing solutions to satisfy the k-degree condition, the complexity increases considerably.

The following section summarizes and discusses the works related to anonymization.

3.4 Summary and Discussion

Existing techniques in the context of RDF document anonymization are really limited. In [24, 41], the authors reduce the complexity of RDF structure to micro-data, where a huge quantity of information such as heterogeneous nodes and relations is simplified and anonymized. However, in a scenario where thousands of heterogeneous resources are present, the current solutions are not appropriate due to the greedy algorithm to generate all possible solutions (anonymous RDF) and then, their measure to evaluate and select the most adequate one.

Since RDF data can be converted, in some cases, to a structured data as databases, database anonymization techniques could be also applied. Small RDF data can be managed by these solutions; however, reducing the complexity of big RDF data into structured models can produce a high semantic information loss (properties), and a huge granularity of the structured normalized-model. Moreover, solutions are proposed for simple cases where data satisfy conditions of anonymity, but when a diversity of values is present, the complexity of the solutions increases exponentially. As RDF data can be also represented as a graph, anonymization graph approaches have been explored in this work. The simplicity of the graph structure assumption makes the current approaches not adequate for the Semantic Web, where heterogeneous nodes and relations are present. Some criteria of anonymization, such as k-degree, can be adopted to the Semantic Web, but the solutions to satisfy these criteria have to be modified according to the complexity of the RDF structure.

Most of the works in RDF documents, databases and graphs anonymization assume that the classification of the data required to satisfy the conditions of anonymity, is provided by expert user. However, the scenario of the Semantic Web complicates the task of classification, since it is difficult to understand the detailed characteristics of external datasets, and assume all the background knowledge possessed by adversaries.

Table 4 shows our analysis in this regard. Note that none of the works on database and graph anonymization satisfies the criteria of complexity of data (heterogeneous nodes and relations). Moreover, the classification on the data is mainly provided by the proposals

and there is no information about how it was performed. We assume that the process to classify the data has been manual. Thus, a new anonymization approach able to cope all requirements is needed to provide an appropriate protection of sensitive information for the Semantic Web.

Before describing how our approach addresses these requirements, the following section introduces some common terminologies and definitions of anonymization in the context of RDF.

4 Terminologies and Definitions

For the Semantic Web, RDF is the *common format* to describe resources, which are abstractions of entities (documents, persons, companies, etc.) of the real world. RDF uses triples in the form of $\langle$`subject`, `predicate`, `object`$\rangle$ expressions also named statements, to provide relationships among resources. The following elements compose the RDF triples:

- An **IRI**, which is an extension of the Uniform Resource Identifier (URI) scheme to a much wider repertoire of characters from the Universal Character Set (Unicode/ISO 10646), including Chinese, Japanese, and Korean character sets [12].
- A **Blank Node**, representing a local identifier used in some concrete RDF syntaxes or RDF store implementations. A blank node can be associated with an identifier (`rdf:nodeID`) to be referenced in the local document, which is generated manually or automatically.
- A **Literal Node**, representing values as strings, numbers, and dates. According to the definition in [9], it consists of two or three parts:
  - A **lexical form**, being a Unicode string, which should be in Normal Form C[7] to assure that equivalent strings have a unique binary representation.
  - A datatype **IRI**, being an IRI identifying a **datatype** that determines how the lexical form maps to an object value.

[7] It is one of the four normalization forms, which consists on a Canonical Decomposition, followed by a Canonical Composition. http://www.unicode.org/reports/tr15/

Table 5 Description of sets

Set	Description
I	A set of IRIs is defined as: I = $\{i_1, i_2, ..., i_l\} \mid \forall i_i \in I$, i_i is an IRI.
L	A set of literal nodes is defined as: L= $\{l_1, l_2, ..., l_m\} \mid \forall l_i \in L$, l_i is a literal node.
BN	A set of blank nodes is defined as: $BN = \{bn_1, bn_2, ..., bn_n\} \mid \forall bn_i \in BN$, bn_i is a Blank Node.

- A **non-empty language tag** as defined by "Tags for Identifying Languages" [1], if and only if the datatype IRI is http://www.w3.org/1999/02/22-rdf-syntax-ns#langString.

Table 5 describes the sets of RDF's elements that we use in our approach description.

After the definition of sets of RDF'elements, we formally describe a triple in Definition 1.

Definition 1 Triple (*t*)*: A Triple, denoted as **t**, is defined as an atomic structure consisting of a 3-tuple with a Subject (**s**), a Predicate (**p**), and an Object (**o**), denoted as* $t :< s, p, o >$*, where:*

$-$ $s \in I \cup BN$ *represents the subject to be described, that can be an IRI or a blank node;*

$-$ $p \in I$ *is a predicate defined as an IRI in the form* `namespace_prefix:pre dicate_name`*, where* $namespace\_prefix$ *is a local identifier of the* IRI*, in which the predicate* $(predicate\_name)$ *is defined. The predicate (p) is also known as the **property** of the triple.*

$-$ $o \in I \cup BN \cup L$ *describes the object, that can be an IRI or a blank node.* ⧫

From our motivating scenario, we can observe several triples with different RDF resources, properties, and literals:

- t_1: `<genid:S1,rdf:type,dbo:School>`
- t_2: `<genid:S1,rdfs:label,"Hartlepool College of Further Education">`
- t_3: `<genid:S1,prop:latitude,1.4545>`
- t_4: `<genid:S1,prop:longitude,0.40244>`

A set of triples defines an RDF document, by encoding the triples, using a predefined serialization format complying with the RDF W3C standards, such as RDF/XML, Turtle, N3, etc. According to the

structure of triples, RDF document can be represented as an RDF Graph, since the structure allows node-edge-node relations. An RDF graph is defined in Definition 2.

Definition 2 ***RDF Graph (*G*)***: *An RDF graph of an RDF document is denoted as* $G_d(N, E)$*, where each triple* t_i *from d is represented as a node-edge-node link. Therefore, G nodes (N), denoted as* n_i*, represent subjects and objects, and G edges (E), denoted as* e_j*, represent corresponding predicates:* $n_i \in \bigcup_{t_i.s \cup t_i.o}$ *and* $e_j \in \bigcup_{t_i.p}$ [49]. ♦

The following subsection presents the formal concepts used in this work.

4.1 Problem Definition

As we show in the motivating scenario, there are cases in which sensitive information can be disclosed through the data published from different sources on the Web (due to data intersection). Thus, the data to be published, denoted as D, should be protected before, in order to avoid compromising the disclosure or production of sensitive information.

The available information on the Web is called background knowledge. It can be provided automatically or semi-automatically by the expert user and can contain simple or complex resources (e.g., one RDF resource, RDF graph, text files). The background knowledge is formally defined in Definition 3.

Definition 3 Background Knowledge (*BK*): *It is a set of IRIs, considered as nodes and denoted as BK:* $\{n_1, n_2, ..., n_i \mid \forall n_i, n_i$ *is a IRI*$\}$. ♦

In this work, we assume that the intersection between *D* and *BK* can disclose or produce sensitive information, hence identifiers and quasi-identifiers appear in D due to the connection among its subjects and objects. We rename both concepts to keys, defined in Definition 4, since they allow the disclosure of sensitive information.

Definition 4 Keys (*K*): *Keys are identifiers and quasi-identifiers, denoted as* $K : \{k_i \mid \forall k_i \in I \cup BN \cup L, k_i\ produces\ sensitive\ information\}$. ♦

We formally define our assumption concerning the intersection between D and BK datasets in Assumption 1.

Assumption 1 Key Detection (Intersection) (*IN*): *The intersection between a set of triples T and a set of IRIs I is defined as a set of nodes (subjects and objects of triples) that belong to the RDF graph of T (G_T), denoted as IN, where each node of IN has another similar one in I. The similarity among the two nodes is measured by a similarity function (simFunc), whose value is equal or greater than an established threshold.*

$$IN : T \sqcap I = \bigcup_{\{n_i \in G_T | sim(n_i \in T, n_j \in I, \alpha, \beta, \gamma) \geq threshold\}}$$

Where:

- $\sqcap$ *is an operator that defines the intersection between triples and IRIs;*
- n_i *is a subjects or object that belong to* T*;*
- n_j *is a IRI that belong to* I*;*
- sim *is the similarity function defined in Definition 5.*

The similarity function between two nodes is defined in Definition 5.

Definition 5 Similarity function (*simFunc*): *The similarity between two nodes is defined as a float value, denoted as simFunc that takes into account three different aspects of the nodes: (i) syntactic; (ii) semantic; and (iii) context analysis, such that:*

$$\begin{aligned} simFunc(n_i, n_j, \alpha, \beta, \gamma) = \; & \alpha \times syntactic\_similarity(n_i, n_j) \\ & + \beta \times semantic\_similarity(n_i, n_j) \\ & + \gamma \times context\_similarity(n_i, n_j)) \end{aligned}$$

Where:

- $n_i \in I \cup BN \cup L$ *and* $n_j \in I$*;*
- *Syntactic_similarity is a function which considers the syntactic aspect of the node, whose values are in [0, 1];*
- *Semantic_similarity is a function which considers the semantic aspect of the nodes, whose values are in [0, 1];*
- *Context_similarity is a function which considers the incoming and outgoing relations of the nodes, whose values are in [0, 1];*
- $\alpha + \beta + \gamma = 1$. ⧫

According to the type of nodes of BK (IRIs), different similarity functions should be provided to discover similar nodes. For instance, similarity in the context of RDF information retrieval has been widely studied and several work analyze queries (e.g., a node, graphs) with respect to RDF structure [17, 54]. Moreover, images, texts, and other multimedia files could be converted to RDF to facilitate the comparison of RDF nodes [15, 20, 25]. The nodes belonging to the intersection between D and BK (IN), are potential keys according to our assumption, then $K = IN$. *For example, according to our motivating scenario, the properties* `prop:Longitude` *and* `prop:Latitude` *from Enipedia dataset (D) are keys since the position identifies a particular Power Plant and have intersection with the ones from DBpedia dataset (BK).* The triples from D that contain at least a key are considered as disclosure sources, defined in 6, since the triples are connected to other resources.

Definition 6 Disclosure Sources (*DS*): *It is a set of triples, which contains at least a key from K, denoted as* $DS : \{ds_i \mid \forall ds_i \in D \wedge (ds_i.s \in K \vee ds_i.o \in K)$, ds_i *is a disclosure source that disclose or produce sensitive information*$\}$. ⧫

However, all triples in D that contain at least a key, cannot be considered as *disclosure sources*, since it depends of the scenario; thus, the interaction of the expert user is needed to identify only the ones that compromise the data to be published. *For example, the triple* $\langle ..., prop{:}lat, 54.6824\rangle$*, from Enipedia dataset (D), is considered as a potential disclosure source since it has a key (*`prop:lat`*) as predicate.* Definition 7 formally explains the result of the expert interaction.

Definition 7 Disclosure-Source Query (*EU*): *It is a selection/projection query applied over DS (*$\prod_{DS}$*), that returns triples considered as disclosure sources by the expert user according to the scenario. This set of triples is denoted as* $EU : \{eu_i \mid \forall eu_i \in DS$, eu_i *is considered as a disclosure source by the expert user*$\}$. ⧫

Using the classification of the expert user, anonymization methods can be applied on the selected triples in order to prevent the disclosure of sensitive information. Note that even the original set of triples (D)

could be protected, it should be re-protected considering the already published data (BK) and their intersections with the original one. A protection operation is formalized in Definition 8.

Definition 8 Protection Function (*ProtFunc*): *It is a function applied on a triple that returns another similar one, by modifying either the subject, the predicate, the object, or all the three RDF elements, to avoid the disclosure of sensitive information. It is denoted as* $ProtFunc(t \in D, op, par)$*, where* op *is a protection operation (e.g., generalization, suppression) and* pr *are the parameters of configuration (e.g., level of generalization).* ⧫

By the result of applying the protection process on the set of triples selected by the expert user, the protected data is obtained. This latter is formalized in Definition 9 and it does not allow the disclosure of sensitive information.

Definition 9 Protected data to be published (*pD*): *It is a set of triples denoted as pD, which is the result of applying any protection technique on the set of triples selected by the expert user (EU) of D; i.e., the data to be published are protected if their intersection with the BK does not produce the triples selected by the expert user, using the same threshold established during the intersection:*

$$pD = D \sqcap \{ProtFunc(eu_i) \mid eu_i \in EU\}$$

Where:

- D *is the data to be published;*
- $\sqcap$ *defines the replacement of the set* $EU \subset D$ *with the one obtained by applying a operation function over its elements;*
- EU *is the set of triples considered as disclosure sources by the expert (see Definition 7);*
- *ProtFunc is a function that applies a protection operation (e.g., generalization, suppression) on either the subject, predicate, object, or all three values.* ⧫

Following the previous example, let's protect the triple $\langle ..., prop{:}lat, 54.6824 \rangle$ *considered as a disclosure source, selected by*

*the expert user, by applying a generalization function over the predicate to reduce its similarity with the DBpedia dataset (**BK**):* $\langle ..., prop: coordinate, 54.6824 \rangle$.

The next section describes our protection process.

5 Protecting Process: Our Proposal

Our protection process mainly relies on a four phases approach (see Figure 4), called ***RiAiR***, where the input, a set of RDF documents in any serialization format (D), is converted into a graph representation, used by all modules: (i) *Reducing-Complexity phase* in which the graph is analyzed to reduce its complexity-structure to extract a compressed one; (ii) *Intersection phase*, where similar nodes between the input graph (reduced or not) from D and the one from the BK are identified as potential keys (IN); (iii) *Selecting phase* in which the expert user analyzes and selects the disclosure sources (EU), which contains at least one potential key; and (iv) *protection phase* that executes a protection process over the selected triples (EU).

A description of each phase is presented in the following sections.

5.1 Reducing-Complexity Phase

Since the expert user needs to classify thousands of triples available in D, a reduction step is needed in order to simplify the interaction and make easy the task of classification. As some triples are essential to describe concepts, they cannot be removed from the data and are considered as constraints. These latter are a set of triples, defined by the expert user, that have an important role over the data. The set of constraints is defined in Definition 10.

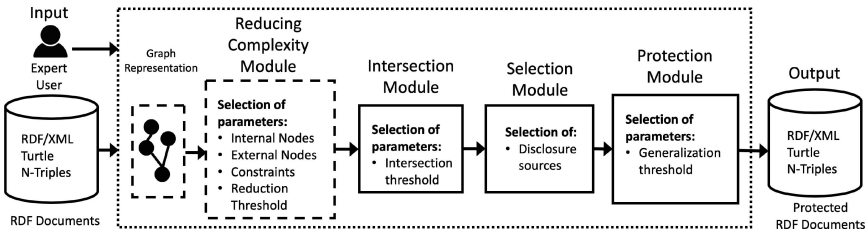

Figure 4 Framework of our RDF anonymization process.

Definition 10 Constraints (*C*): *It is a selection/projection query applied over D ($\prod_D$) that indicates the triples to be preserved. It is denoted as C:*$\{c_i{:}< s_i, p_i, o_i >| \forall c_i \in D, c_i$ *is a triple to be preserved*$\}$. ♦

For example, we define as a constraint the triples whose predicates are equal to the value $http://www.w3.org/1999/02/22\text{-}rdf\text{-}syntax\text{-}ns\#type$, since it describes the concept of a resource.

The set of triples $\mathrm{T} = \{\mathtt{t_i} : \langle \mathtt{s_i}, \mathtt{p_i}, \mathtt{o_i} \rangle\}$ of D is analyzed by the similarity function *simFunc* defined in Definition 5, considering the set of IRIs as a simple node (e.g., a resource). This similarity should take into account the context of the value (e.g., a similarity function based on the incoming and outgoing relations) instead of the analysis of the value itself in order to identify a more general resource. From two similar nodes, the one that subsumes the other is kept. A sorting step to organize the triples in a defined order is needed to return a unique output (e.g., Depth-Subject-Predicate-Object order). As sensitive information can be present in resources and literal values as well, we classify the nodes into two categories: *internal nodes*, which are the ones that are subjects and objects at the same time, and *external nodes* that are only objects in the set of triples (T).

We propose Algorithm 1 and Algorithm 2 to reduce the complexity of each category of nodes. The reducing-complexity algorithm applied on internal nodes, receives a set of triples $\mathrm{T} = \{\mathtt{t_i} : \langle \mathtt{s_i}, \mathtt{p_i}, \mathtt{o_i} \rangle\}$, a threshold th_1, a similarity function *simFunc*, and returns another set of triples $\mathrm{T}' = \{\mathtt{t'_i} : \langle \mathtt{s'_i}, \mathtt{p'_i}, \mathtt{o'_i} \rangle\}$. In Algorithm 1, each triple (t_i) in T is analyzed by the *simFunc* applied to its subject (node) with other subjects from T (lines 4–5 of Algorithm 1). If the *simFunc* is equal or greater than the defined threshold (*th*), the triple (t_i) is added to the list *processedListTriples* and the subject of t_i will be replaced by the one from t_j in all triples from T (lines 8, 9 of Algorithm 1). The replacing function is performed in line 11 of Algorithm 1 and the modified set of triples is returned in line 13.

The algorithm for external nodes receives a set of triples $\mathrm{T} = \{\mathtt{t_i} : \langle \mathtt{s_i}, \mathtt{p_i}, \mathtt{o_i} \rangle\}$, a threshold th_1, a similarity function *simFunc*, and returns another set of triples $\mathrm{T}' = \{\mathtt{t'_i} : \langle \mathtt{s'_i}, \mathtt{p'_i}, \mathtt{o'_i} \rangle\}$, according to the

Algorithm 1: Reducing complexity – Internal nodes

```
   Input: Set of triples T={t:⟨s,p,o⟩}, threshold th_1, Function simFunc
   Output: Set of triples T'
 1 processedListTriples = {};        //List of processed triples.
 2 replaceListNodes = {};        //List to replace nodes in the set of triples.
 3 T = T.sort(HSPO);          //Sorting by depth-subject-predicate-object order.
 4 foreach t_i in T do
 5     foreach t_j in T-{t_i} do
 6         if t_j ∉ processedListTriples then
 7             if simFunc(t_i.s, t_j.s)≥ th_1 then
 8                 processedListTriples.add(t_i);
 9                 replaceListNodes.add(Pair(t_i.s,t_j.s));
10                 break; //Since a similar node was found, the next t_i is
                     analyzed.
11 T' = T.replaceNodes(replaceListNodes);  // Nodes are replaced.
12 T' = T'.removeDuplicateTriples();        //Duplicate triples are removed.
13 return T';
```

threshold (*th*) provided by the expert user. A list, called *removeListTriples*, is used to store temporarily the triples to be removed in the last step of the algorithm (line 1 in Algorithm 2). As the previous algorithm, a sorting step is needed to return an unique output. Each subject (node) from triple t_i in T is compared with other subjects that belong to the triples in T, using the similarity function *simFunc* defined in Definition 5 for simple nodes. To verify if the triple has an external node, its depth[8] is calculated. If the depth of t_i is different than 0, then the object node is not external, and we move forward to the next triple in T (lines 4–5 of Algorithm 2). If the *simFunc* between t_i and t_j is equal or greater than the defined threshold and t_i does not belong to the set of constraints (C in Algorithm 2) defined by the expert user (see Definition 10), the triple t_i is added to the *removeListTriples* list (lines 8–10 in Algorithm 2). Finally, the triples are removed in line 11 in Algorithm 2).

[8]The depth of a triple is considered as the biggest path of its object to a terminal node.

Algorithm 2: Reducing complexity – External resource

```
Input: Set of triples T={t:⟨s,p,o⟩}, threshold th_1, Function simFunc
Output: Set of triples T'
1  removeListTriples = {};     //List to remove triples.
2  T = T.sort(HSPO);           //Sorting by depth-subject-predicate-object order.
3  foreach t_i in T do
4      if t_i ∈ removeListTriples ∨ depth of t_i ≠ 0 then
5          continue;           //Next triples is analyzed.
6      foreach t_j in T - {t_i} do
7          if t_j ∉ removeListTriples then
8              if simFunc(t_i.s, t_j.s)≥ th_1 and t_i ∉ C then
9                  removeListTriples.add(t_i);     //Adding triples to be
                     removed.
10                 break; //Since a similar node was found, the next t_i is
                     analyzed.
11 T' = T.removeTriples(removeListTriples);   //Triples of removeListTriples
     list are removed.
12 return T';
```

Note that Algorithm 1 and Algorithm 2 are independent and they can be used in any order.

The reducing-complexity algorithms are applied to the data to be published (D). Once the reductions are obtained, the intersection among this set and the BK can be performed. Following phase describes the intersection phase.

5.2 Intersection Phase

The previous phase reduces the complexity-structure of D; the number of triples of D to decrease the interaction of the expert user over the data. However, identifying the triples that are disclosure sources in the reduced set of D, is still a difficult task for the expert user. To identify the nodes of the reduced set D that belong to the intersection with the background knowledge (BK), we propose Algorithm 3, based on the *intersection among two datasets* assumption (see Assumption 1) and using the similarity function defined in Definition 5.

Algorithm 3: Intersection among two datasets

Input: Set of triples $T = \{t_i{:}\langle s,p,o\rangle\}$, I, threshold th_2, Function simFunc
Output: Set of nodes IN

```
1  IN = {};      //Set of nodes.
2  foreach t_i in T do
3      foreach i_j in I do
4          if simFunc(t_i.s, i_j) ≥ th_2 then
5              if t_i.s ∉ IN then
6                  IN.add(t_i.s);      //The subject of T is added.
7          if simFunc(t_i.o, i_j) ≥ th_2 then
8              if t_i.o ∉ IN then
9                  IN.add(t_i.o);      //The object of T is added.
10 return IN;
```

Algorithm 3 receives a set of triples $T = \{\mathtt{t_i} : \langle \mathtt{s_i}, \mathtt{p_i}, \mathtt{o_i}\rangle\}$, a set of IRIs I, a threshold th_2, a similarity function *simFunc*, and returns a set of nodes IN, according to the threshold defined by the expert user. Each subject and object from triple t_i in T is analyzed by using the similarity function (*simFunc*) with the IRI i_j in I. If *simFunc* is equal or greater than the defined threshold (*th*), the subject or object from triple t_i in T are added to the list IN (lines 4–9 in Algorithm 3). The set IN is returned in line 10.

The nodes of IN are considered as potential keys (see Definition 4), since they allow the connection of the data to be published with other datasets. Following section presents the selecting phase which is executed by the expert user.

5.3 Selecting Phase

According to Definition 6, triples that contain at least a key are disclosure sources and can disclose or produce sensitive information; however, not all triples that belong to this definition can reveal sensitive information; therefore, the interaction of the expert user is needed to select only the triples that compromise the data. The selection can be performed by a query or any other method.

To further simplify the expert user interaction, we propose the use of a Graphic User Interface (GUI) based on the set of potential disclosure sources (DS). By a visual interface, the expert user can analyze and select only the triples which are disclosure sources for the scenario. The set of triples obtained by the selection of the expert user, is the set EU (see Definition 7).

Following section describes the protecting phase applied over the set of triples EU.

5.4 Protection Phase

Once the disclosure sources are selected by the expert user, a protection process on these triples can be performed. We propose the use of generalization operations on the predicate of each triple, to only reduce the connections among datasets (D and BK), preserving the objectives of integration and combination of the Semantic Web. A taxonomy for each type of relation from the set of triples EU (see Definition 7), has to be provided by the expert user. Moreover, a measure to calculate the level of generalization, applied to the taxonomies (to choose a predicate form a set of values), is needed (e.g., hierarchical and taxonomy measures) in order to provide an appropriate, customized and measured protection according to different scenario. Algorithm 4 describes the protection process by applying a generalization operation on each selected triple of EU (see Definition 8).

Algorithm 4: Protection process

Input: Set of triples $T = \{t_i{:}\langle s,p,o\rangle\}$, Set of taxonomies TA, Level of generalization g
Output: Set of triples T'

```
1 T' = {};                                  // Set of triples.
2 foreach t_i in T do
3     Taxonomy ta = TA.getTaxonomy(t_i.p);  // Taxonomy of predicate
        t_i.p.
4     t_i.p = ta.getPredicate(g);           // Predicate from taxonomy ta.
5     T'.add(t);                            // The modified triple is added to T'.
6 return T';
```

Algorithm 4 receives a set of triples T $= \{\mathtt{t_i} : \langle \mathtt{s_i}, \mathtt{p_i}, \mathtt{o_i} \rangle\}$, a set of taxonomies TA, a level of generalization g, which is a value among [0, 1], and returns a set of modified triples T' $= \{\mathtt{t'_i} : \langle \mathtt{s'_i}, \mathtt{p'_i}, \mathtt{o'_i} \rangle\}$, according to the taxonomies and the level of generalization provided by the expert user. From the set of taxonomies provided by the expert user (TA), the taxonomy which corresponds to the predicate of t_i ($t_i.p$) is used to obtain another predicate that satisfy the level of generalization (g) (lines 3 and 4 in Algorithm 4). The modified triple is added to the list T' (line 5 in Algorithm 4) and the whole list is returned in line 6.

Note that to obtain the protected RDF data, the compressed triples selected by the expert user, have to be released to apply the protection process over their triples.

Our whole proposal overcomes the limitations identified in the context of RDF protection, such as the assumptions of simple data that is not similar to the one available on the Web, and the high interaction of expert user for the classification. The proposal is designed for RDF data, considering their elements (IRIs, blank nodes and literals) and the scenario, where linked and heterogeneous resources are available. The complexity of the RDF structure is reduced in order to decrease the interaction of the expert user and to make easy the task of classification. Potential keys are identified and disclosure sources are provided to the expert user. Moreover, by a generalization method, we reduce the connections among datasets, preserving the main objectives of the SW (integration and combination), and protecting the sensitive information at the same time.

The following section evaluates the complexity of our proposal.

5.5 Complexity Analysis of the Whole Anonymization Process

A complexity analysis of our anonymization approach indicates a quadratic order performance in terms of number of triples of the data to be published (n) and the ones from the background knowledge (m), i.e., $O(n^2 + m^2)$. A detailed complexity analysis was done on each phase of the process to get the complexity of the whole process:

- For the Reducing-complexity phase, each triple (n) is analyzed by searching another similar one in the set of triples, then their execution order is $O(n^2)$.
- The Intersection phase based on the reduced set of triples from D, has an execution order $O(n \times m)$, D and BK respectively, for the worst case where no triple was removed by reducing-complexity phase.
- The Configuration phase, which is made by the expert user, depends of the number of triples from D that contain potential keys, which are obtained by the intersection between D and BK. Thus, this phase has an execution order $O(n)$ where all triples are considered as disclosure sources.
- The anonymization phase, applied over the triples selected by the expert user, has an execution order of $O(n)$, if all triples from D are considered as disclosure sources.

As the four phases are executed sequentially, the whole protection approach exhibits a quadratic order complexity, i.e., $(O(n^2 + m^2 + n \times m + 2 \times n))$.

The following section evaluates the viability and demonstrate the quadratic order performance of our proposal.

6 Experimental Evaluation

To show the viability and performance of the approach for heterogenous datasets available on the Web, we performed an experimental evaluation.

6.1 Prototype and Implementation

To evaluate and validate our protection approach, a desktop prototype system, called *RiAiR*, was developed using Java. Figure 5 shows a visual interface of our prototype, which has several customizable options according to user-preferences. For example, the expert user can apply the reducing-complexity process to either internal, external nodes, or only one of them. The thresholds for the reduction, intersection, and

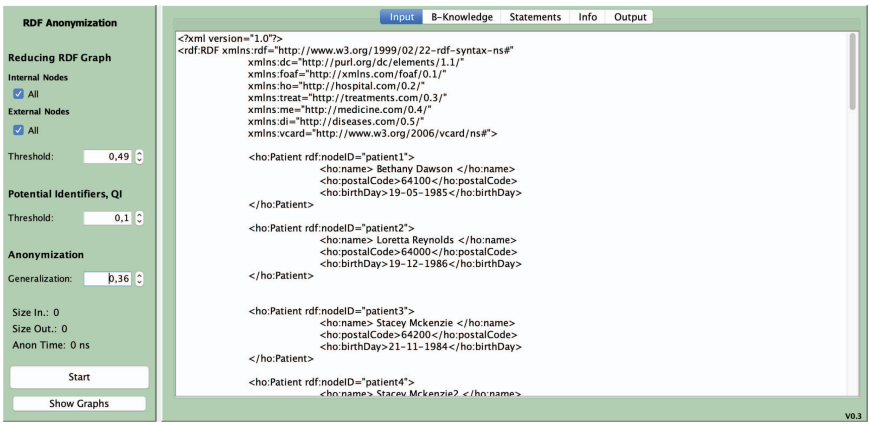

Figure 5 Visual interface of our protection approach.

protection processes can be also customized by the expert user, selecting a value among [0,1] in the left area of the visual interface.

For the reducing-complexity and intersection phases, we implemented the similarity function, called *simFunc* (Definition 5), considering only the context similarity to be independent of the domain and to address more heterogeneous datasets. The function is defined by using the incoming and outgoing properties (relations) from the nodes, since the behavior of a node can be determinate through its relations (context). We present the similarity function as follows.

$$
\begin{aligned}
simFunc(n_i, n_j, \alpha = 0, \beta = 0, \gamma = 1) &= \alpha \times syntactic\_similarity \\
&+ \beta \times semantic\_similarity \\
&+ \gamma \times (0.5 \times \frac{|incomingProperties(n_i) \cap incomingProperties(n_j)|}{|incomingProperties(n_i) \cup incomingProperties(n_j)|} \\
&\quad + 0.5 \times \frac{|outgoingProperties(n_i) \cap outgoingProperties(n_j)|}{|outgoingProperties(n_i) \cup outgoingProperties(n_j)|})
\end{aligned}
$$

Where:

- $incomingProperties$ is a function that returns the incoming relations of a node;
- $outgoingProperties$ is a function that returns the outgoing relations of a node.

Note that for the reducing-complexity phase, the intersection and union among properties is made by a syntactic string comparison; while for the intersection phase (see Definition 4), since the datasets are provided from different sources, the syntactic comparison is performed to only the *property name* of the incoming and outgoing properties (e.g., http://www.domain1.com/**nameProp** is equal to http://www.domain2.com/**nameProp**, since both property names are equals – nameProp).

For the anonymization phase, we implemented a generalization operation based on taxonomies provided by the expert user. The taxonomies are processed by the approach through the use of a simple document in XML format, presented as follows.

```
<taxonomies>
    <taxonomy_1>
        <taxonomy_1a>
        </taxonomy_1a>
        <taxonomy_1b>
        </taxonomy_1b>
    </taxonomy_1>
    <taxonomy_2>
        <taxonomy_2a>
        </taxonomy_2a>
    </taxonomy_2>
    ...
</taxonomies>
```

A taxonomy for each triple of the set EU (see Definition 7) is analyzed by applying a similarity measure that returns another similar relation (predicate) according to a defined threshold. We use the similarity measure of work [10], since it takes into account the deepness, the distance, and the children in common of the taxonomies.

6.2 Datasets and Environment

Our prototype was used to perform several experiments to evaluate the viability and the performance (execution time) of our approach

in comparison with the related work. To do so, we considered three datasets:

- **Data 1**: The *DBpedia person data*[9] with 16,842,176 triples (used to evaluate the reducing-complexity phase due to the huge number of triples);
- **Data 2 (BK)**: The *DBpedia geo coordinates*[10] with 151,205 triples; and
- **Data 3 (D)**: An extraction of Enipedia dataset (power plants), considering properties `art:Nuclear`, `cat:radioactive`, `prop:City`, `prop:Country`, `prop:lat`, `prop:long`, and `prop:year`, with 568 triples.

Using **Data 1**, **Data 2**, and **Data 3**, we evaluated the viability and performance of the reducing-complexity process, while for the intersection phase, we used **Data 2** and **Data 3**. The protection phase is applied over the reduced set of triples obtained by the reducing-complexity phase and the set of nodes of the intersection phase between **Data 3** and **Data 2**. Since in this particular case the BK is also a set of triples (a complex node), we applied the reducing-complexity process over the dataset as well. Experiments were undertaken on a MacBook Pro, 2.2 GHz Intel Core(TM) i7 with 16.00 GB, running a MacOS High Sierra and using a Sun JDK 1.7 programming environment.

6.3 Evaluation metrics

6.3.1 Accuracy in disclosure sources

In order to evaluate the accuracy of our approach when a set of triples are suggested as disclosure sources to the user expert, we calculated the F-score, based on the Recall (R) and Precision (PR). These criteria are commonly adopted in information retrieval and are calculated as follows:

---

[9]Information about persons extracted from the English and Germany Wikipedia, represented by the FOAF vocabulary – `http://wiki.dbpedia.org/Downloads2015-10`.

[10]Geographic coordinates extracted from Wikipedia – `https://wiki.dbpedia.org/downloads-2016-10`.

$$\textbf{PR} = \frac{A}{A+B} \in [0,1] \quad \textbf{R} = \frac{A}{A+C} \in [0,1] \quad \textbf{F-score} = \frac{2 \times PR \times R}{PR+R} \in [0,1]$$

where A is the number of correctly suggested triples; B is the number of wrongly suggested triples; and C is the number of triples not suggested by our approach but considered as disclosure sources.

According to our scenario, **Data 3** contains eight properties, from which only two properties (prop:lat and prop:long) are considered as disclosure sources. Thus, 142 triples need to be selected by the user expert, since 71 power plants are present. We describe the accuracy evaluation in subsection Configuration Phase.

6.3.2 Protection data verification

To consider a data as a protected one, it should not contain disclosure sources which compromise the data; thus, to verify the data, we propose a measure based on the sensitive triples returned by applying a query over the datasets. The verification is performed as the relation between the sensitive information produced by the original data with respect to the one produced by protected data; i.e.,

$$\textbf{AnonV(D, pD)} = \frac{N.\,of\ sensitive\ triples\ from\ D\ -\ N.\,of\ sensitive\ triples\ from\ pD}{N.\,of\ sensitive\ triples\ from\ D} \in [0,1].$$

where D is the data to be published and pD the protected one (see Definition 9).

For our evaluation, we use the query presented in our motivating scenario, considering any type of resources (e.g., `dbo:School`, `dbo:Hospital`). A total of 364 entities, represented by 1456 triples, are sensitive information.

```
SELECT DISTINCT
?Place ?g bif:st_distance(?g,bif:st_point(".$long.",".$lat.")) AS ?distance
FROM <http://dbpedia.org>
WHERE {?p rdfs:label ?Place ; geo:geometry ?g.
FILTER (bif:st_intersects (?g, bif:st_point (".$long.", ".$lat."), 100)
&& (lang(?Place) = \"en\"))}
ORDER BY ASC(?distance)
```

This metric evaluates the protected RDF data in the subsection Protection Phase. We describe and evaluate as follows each process to obtain a protected RDF data.

6.4 Reducing-Complexity Phase

We performed the reducing-complexity process over three real datasets available on the Web (**Data 1**, **Data 2**, and **Data 3**). We evaluated the Jena parsing-time (ms) and the size (bytes) of the input and output to compare the improvement of working over the output in terms of viability and performance.

6.4.1 Viability evaluation

Test 1: We chose randomly the value 0.44 as the threshold for the reducing-complexity process. We extracted 1,000 triples from each dataset and increased the number of triples by a step of 1,000 for the next iterations. Table 6 shows the results obtained for **Data 1**. This process reduced the complexity of more than 16 millions of triples to only 132 triples, since the values were extracted from *Wikipedia* following a schema with a finite number of properties. The Jena parsing-time of the input is reduced to 1.03 ms (132 triples) and its size to 9333 bytes. Note that applying the same threshold for different sets of triples extracted from **Data 1**, we obtain the same output for all the cases, showing that the general schema of the resources (finite number of properties) is returned by this process.

For **Data 2**, Table 7 shows the results of applying the reducing-complexity process. The dataset of 151,205 triples is reduced to only 4 triples, i.e., the 151,205 triples follow the schema represented by the 4 returned triples. The Jena parsing-time and the size of the input were

Table 6 **Test 1**: Reducing-Complexity process for **Data 1**, using a threshold 0.44

Data 1	**Input**			**Output**		
Threshold	**Triples**	**Jena Time (ms)**	**Size (bytes)**	**Triples**	**Jena Time (ms)**	**Size (bytes)**
0.44	1,000	7.99	68958	132	1.10	9333
0.44	2,000	16.89	138108	132	1.08	9333
0.44	3,000	23.95	207036	132	1.12	9333
0.44	4,000	30.41	276070	132	1.05	9333
0.44	5,000	36.50	345687	132	1.07	9333
0.44	6,000	42.75	414809	132	1.15	9333
0.44	7,000	48.23	484719	132	1.06	9333
0.44	8,000	53.11	553507	132	1.10	9333
0.44	9,000	56.93	622646	132	1.01	9333
0.44	10,000	61.12	666224	132	1.09	9333
0.44	**16,842,176**	–	–	132	1.03	9333

Table 7 **Test 1**: Reducing-Complexity process for **Data 2**, using a threshold 0.44

Data 2	**Input**			**Output**		
Threshold	**Triples**	**Jena Time (ms)**	**Size (bytes)**	**Triples**	**Jena Time (ms)**	**Size (bytes)**
0.44	1,000	9.45	77144	4	0.40	455
0.44	2,000	17.94	154729	4	0.35	455
0.44	3,000	25.37	232222	4	0.39	455
0.44	4,000	31.49	309952	4	0.44	455
0.44	5,000	38.63	387289	4	0.36	455
0.44	6,000	44.98	464888	4	0.41	455
0.44	7,000	51.81	543737	4	0.37	455
0.44	8,000	57.41	622768	4	0.36	455
0.44	9,000	62.74	700421	4	0.39	455
0.44	10,000	69.89	778651	4	0.42	455
0.44	**151,205**	–	–	4	0.40	455

Table 8 **Test 1**: Reducing-Complexity process for **Data 3**, using a threshold 0.44

Data 3	**Input**			**Output**		
Threshold	**Triples**	**Jena Time (ms)**	**Size (bytes)**	**Triples**	**Jena Time (ms)**	**Size (bytes)**
0.44	568	4.99	37645	8	0.68	769

reduced to 0.40 ms and 455 bytes, respectively. In **Data 3**, the output contains only 8 triples from 568 triples as we can observe in Table 8. The Jena parsing-time and the size of the dataset was reduced to 0.68 ms and 769 bytes, respectively. Similarly to the two previous data sets, the 8 returned triples represents the scheme of all triples in the set.

Test 2: In order to select the best threshold for the reducing-complexity process of each dataset, we evaluated the number of triples, Jena parsing-time, and the size of the output by using a threshold value between [0.01 – 1.00] with a step of 0.01. Table 9 shows the results obtained for **Data 1**. As we can observe, we obtained the best result for the thresholds from 0.01 to 0.29, where only nine properties are used in the whole database. The Jena parsing-time of the output was reduced to 0.49 ms, while the size was reduced to 834 bytes.

For **Data 2** and **Data 3** (see Tables 10 and 11), the best results were obtained for a wide range of thresholds [0.01–0.49]. By regarding the datasets, in **Data 2** and **Data 3**, all resources were described by the same properties (four and eight properties, respectively), while in **Data 1**, there are some resources described by only three or four properties from a total of nine, therefore in **Data 1**, the optimal threshold was obtained

Table 9 **Test 2**: Reducing-Complexity process for **Data 1** with a step 0.01

Data 1	**Input**			**Output**		
Threshold	**Triples**	**Jena Time (ms)**	**Size (bytes)**	**Triples**	**Jena Time (ms)**	**Size (bytes)**
[1.00 , 0.50]	10,000	63.62	666224	10,000	62.56	666224
[0.49 , 0.45]	10,000	61.54	666224	148	1.17	10420
0.44	10,000	62.21	666224	132	1.12	9333
0.43	10,000	65.32	666224	111	0.96	7934
0.43	10,000	62.59	666224	75	0.86	5423
[0.41 , 0.40]	10,000	61.98	666224	55	0.80	4040
0.39	10,000	60.81	666224	39	0.72	3069
0.38	10,000	62.44	666224	26	0.63	2174
[0.37 , 0.36]	10,000	62.86	666224	33	0.65	2617
[0.35 , 0.34]	10,000	61.12	666224	18	0.56	1523
[0.33 , 0.30]	10,000	63.29	666224	12	0.51	1047
[0.29 , 0.01]	10,000	63.58	666224	9	0.49	834
0.29	**16,842,176**	–	–	9	0.49	834

Table 10 **Test 2**: Reducing-Complexity process for **Data 2** with a step 0.01

Data 2	**Input**			**Output**		
Threshold	**Triples**	**Jena Time (ms)**	**Size (bytes)**	**Triples**	**Jena Time (ms)**	**Size (bytes)**
[1.00 , 0.50]	10,000	69.25	778651	10,000	69.42	778651
[0.49 , 0.01]	10,000	70.91	778651	4	0.39	455
0.49	**151,205**	–	–	4	0.39	455

Table 11 **Test 2**: Reducing-Complexity process for **Data 3** with a step 0.01

Data 3	**Input**			**Output**		
Threshold	**Triples**	**Jena Time (ms)**	**Size (bytes)**	**Triples**	**Jena Time (ms)**	**Size (bytes)**
[1.00 - 0.50]	568	4.92	37645	568	4.89	37645
[0.49 -0.01]	568	4.71	37645	8	0.39	769
0.49	**568**	–	–	8	0.39	769

in a smaller range [0.01–0.29], since for the range [0.30–0.49], some resources were not considered as similar to the general schema due to their less number of properties.

6.4.2 Performance evaluation

To evaluate the performance of of the reducing-complexity phase, we measured the average time of 10 executions for each test.

Test 3: We evaluated the time of the reducing-complexity process of 10,000 triples from **Data 1** by using several thresholds between [0.01–1.00] in order to observe the influence of the threshold over

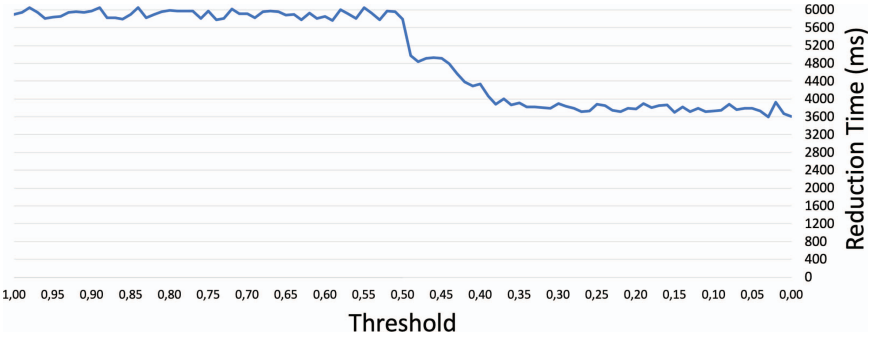

Figure 6 **Test 3:** Execution time of the reducing-complexity process using a threshold between 0.01 and 1.00.

the reduction time. Figure 6 shows that from a threshold 0.49, where the number of triples is reduced to only 148, the reduction time decreases to 4,977.91 ms until 3,668.54 ms for a threshold value of 0.01. As more triples are reduced during the reducing-complexity process, less comparisons are performed, since for each iteration less operations of similarity are needed to discover another similar node.

Test 4: In this test, we evaluated the impact of the number of triples, from **Data 1**, on the execution time of the reducing-complexity phase. We used a threshold value of 0.29, which was one of the thresholds that reduced more triples, and a step of 10,000 triples for the iterations. Figure 7 shows the execution time with respect to the number of triples. For 60,000 triples, the execution time is 302.65s. The result obtained confirms the quadratic performance of this process. The following section evaluates the intersection phase.

6.5 Intersection Phase

Using the reduced datasets of **Data 2** and **Data 3**, obtained by the reducing-complexity process (4 and 8 triples, respectively), we perform the intersection process considering **Data 3** as the data to be published (D), while **Data 2** as the background knowledge (BK).

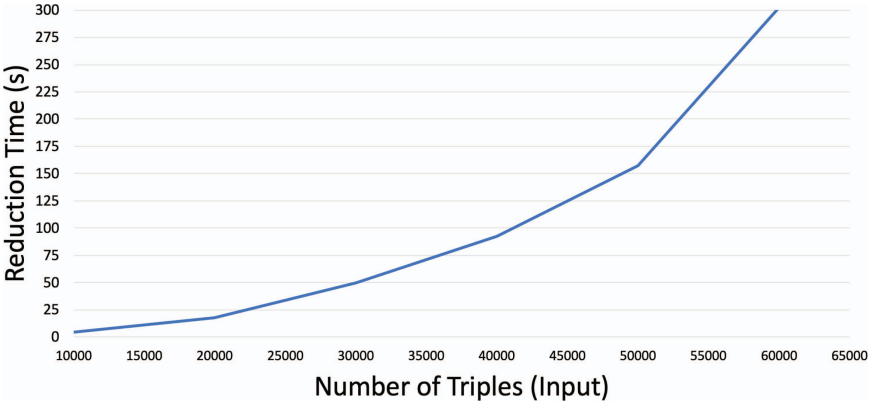

Figure 7 **Test 4:** Execution time of the reducing-complexity process using a threshold value of 0.29.

6.5.1 Viability evaluation

To evaluate the viability of applying this process over real scenarios, we chose randomly a threshold value (0.65) and later we analyzed the behavior of this process with respect to several threshold values.

Test 5: By using a threshold value of 0.65, the intersection process did not return any intersection node. Regarding the reduced datasets, the nodes that represent the latitude and longitude properties are terminal nodes, thus they do not have outgoing properties and its similarity is less than 0.50. Additionally, the similarity between the node which represents a power plant (**Data 3**) and the one which represents a place of interest (**Data 2**) is calculated based on two properties in common (intersection – latitude and longitude) from ten properties (union – eight properties in D and four properties in BK), thus their similarity value is 0.20.

Test 6: We evaluated the viability of this process using several thresholds from 0.01 to 1.00 with a step of 0.01 (see Table 12). From a threshold value between 1.00 and 0.50, no node was returned. For [0.49, 0.21], two nodes which represent the coordinates of the nuclear power plant resource in D, are returned as potential keys, which is what we expect. For [0.20, 0.01], three nodes are returned (coordinates and the node which represents the nuclear power plant).

Table 12 **Test 6**: Intersection process for **Data 3** with a step 0.01

Threshold	Number of Nodes
[1.00 - 0.50]	0
[0.49 - 0.21]	2
[0.20 - 0.01]	3

Table 13 **Test 9:** Accuracy evaluation for the set of triples suggested as disclosure sources to the Expert User

Intersec. Thres-hold	N. of potential keys	Triples suggested as disclosure sources (Expert User Interface)	Triples suggested as disclosure sources (Internal Mapping)	Valid	Not Valid	Not sugges-ted	Prec. (%)	Rec. (%)	F-s. (%)
[1.00 , 0.50]	0	**0**	0	0	0	142	0	0	**0**
[0.49 , 0.21]	2	**2**	142	142	0	0	100	100	**100**
[0.20 , 0.01]	3	**8**	568	142	426	0	25	100	**40**

6.5.2 Performance evaluation

Test 7: The time required to discover the nodes that can be potential keys, was measured. An average of 10 execution indicates a time of 0.24 ms for this process.

6.6 Selecting Phase

A GUI based on triples was built to reduce the effort of the expert user. The interface selects automatically the triples which contain at least one key, considered as potential disclosure sources.

Test 8: We measured the average of verifying the selected triples, which contain the nodes detected during the intersection process, of 10 people that have under- and post-graduate degrees in Computer Science. Since only eight triples are available in the reduced dataset of **Data 3**, the verifying average time was 8.23 s.

Test 9: We evaluated the accuracy of the set of triples suggested as disclosure sources by our approach, using the F-score measure. Table 13 shows that for a threshold between [0.49, 0.21] all triples which compromise the data to be published are suggested (Data 3), obtaining a F-score 100%. For a threshold between [0.20, 0.01] also the triples which compromise the data are suggested, but other triples were suggested as well. These thresholds have a F-score of 40%.

6.7 Protection Phase

The relations (properties) that belong to the triples considered as disclosure sources by the expert user, have to be protected in order to reduce the risk of disclosure of sensitive information. According to the configuration process, the eight triples from the reduced set of **Data 3** were pre-selected in the selecting interface, showing that they can be potentially used to disclose sensitive information. By the verification of the expert user, the anonymization process is performed. Since there are eight triples with different properties (predicates), eight taxonomies need to be provided by the expert user.

Test 10: We measured the average time of 10 executions, by using a random threshold of generalization (0.36). A time of 1.12 ms was required to perform this process.

Test 11: Additionally, we evaluated the protected data by using the *AnonV* function defined in subsection evaluation metrics. Table 14 shows that for a threshold less than 0.50 in the intersection phase, the protected data (pD) does not produce sensitive information, obtaining the maximum evaluation value (100%).

In these subsections, we evaluated the viability and performance of our approach by using datasets available on the Web. We demonstrated a huge reduction of the expert-user interaction suggesting disclosure sources. Also, a high performance was obtained for all the phases. Following subsection evaluates our approach with respect to related work.

Table 14 **Test 11:** Protection data evaluation according to the number of sensitive triples produced by the D and pD

Intersec. Threshold	Sensitive Triples in D	Sensitive Triples in pD	Protected Data Verification (%)
[1.00 , 0.50]	1456	1456	0
[0.49 , 0.21]	1456	0	100
[0.20 , 0.01]	1456	0	100

Table 15 **Test 12:** Related Work Comparison

Work	Complex. of data	Triples	Classification		Anonymization Time (s)			Total Time (s)
			Type	Time (s)				
[41]	RDF	D: 568 BK: 10,000	Manual (I, QI, SI, USI)	∼10,568(*)	>3,789.24(+)			>14,357.24 (>3.99 h)
[44]	Structured data	D: 528 BK: 10,000	Manual (I, QI, SI, USI)	∼10,568(*)	>3,632.67(+)			>14,200.67 (>3.94 h)
[53]	Graph	D:568 BK:10,000	Manual (I, QI, SI, USI)	∼10,568(*)	>3,721.34(+)			>14,289.34 (>3.97 h)
Our Approach	RDF	D: 568 BK: 10,000	Automatic (I, QI)	8.23 (Verification)	**Reduc.** D: 0.82 BK: 4.46	**Inter.** 0.00024	**Anon.** 0.00112	13.51 (0.00375 h)
(*) An estimation of 1 second for each triple. (+) The approach was stopped after an hour of execution.								

6.8 Related Work Comparison

In order to compare the viability and the performance of our approach with respect to the state of the art, we selected a work for each identified group of the related work section. For RDF data, we selected the work in [41], for structured data (database) the work in [44], while for graph data the work in [53]. Thresholds of 0.49, 0.10, and 0.36 were used for the reducing-complexity (D and BK), intersection, and generalization processes, respectively in our approach. The implementation of each work was done following the same development environment used for our approach, such as computer specifications and programming language.

Test 12: We evaluated the average time of 10 executions of the anonymization processes. From **Data 2**, 10,000 triples are considered as the background knowledge (BK) and the whole **Data 3** as the data to be published (D). Table 15 shows the results obtained for this comparison. The non-viability of the works in [41, 44, 53] for real scenarios, was clearly demonstrated in this evaluation, since the interaction of the expert user to classify the data, required a high effort (more than three hours), making this task almost impossible. Moreover, the execution time of the anonymization processes, without considering the classification, was greater than one hour for [41,44,53] (the executions were stopped after one hour of processing), while for our solution was only 5.28 s. Note that we considered the time of classification similar to the time of verification which was obtained in our configuration-phase evaluation (∼1 second for triple).

Following section presents our conclusions of this paper.

7 Conclusions

In this paper, we investigated the protection of sensitive information for RDF documents before publication on the Web. We proposed a protection approach, consisting on four phases: (i) *Reducing-Complexity phase*, where the input, a set of RDF documents (D) in any serialization format, is analyzed to reduce its graph complexity; (ii) *Intersection phase*, where similar nodes (IN) between the reduced graph from the data to the published (D) and the one from the background knowledge (BK) are identified as potential keys; (iii) *Configuration phase* in which the expert user analyzes and selects the triples that contain at least one potential key, considered as disclosure sources (EU); and (iv) *protection phase* that executes an generalization operation over the selected triple.

We evaluated the viability and performance of our anonymization approach with several datasets available on the Web. Results show that our approach decreases the interaction of the expert user by reducing the complexity of the graph structure (reducing-complexity phase), identifying potential keys (intersection phase), and suggesting potential disclosure sources through a graphic user interface to the expert user. Moreover, we evaluated our approach with respect to the state of the art, demonstrating that our proposal overcomes existing solutions, and these latter are not able to manage linked and heterogeneous resources.

To select an adequate threshold for the reducing-complexity and intersection phases, the structure of the dataset needs to be analyzed before. For instance, the dataset **Data 1**, from our experimental evaluation, has a depth equal to 2 and it is composed by sub-graphs that are not linked, so to compare the root nodes, a threshold similarity less than 0.50 is required since they do not have incoming properties.

We are currently working on a new graphic user interface based on graph visualization to better illustrate the relations among the datasets. Furthermore, we are testing different similarity function to provide a better reducing-complexity and intersection processes for heterogeneous datasets. For the intersection phase, new semantic similarity functions are required to recognize potential keys that are not from the same domain (e.g., SameAs service). Additionally, the datasets can be enriched with new properties (relations) or extra inferred information

as a pre-step in order to better perform the similarities (e.g., syntactic and semantic datatype inference as in [11, 18], respectively).

References

[1] M. Davis A. Phillips. Tags for Identifying Languages. `https://tools.ietf.org/html/bcp47`. Online; accessed 2017-09-11.

[2] Ainur Abdrashitov and Anton Spivak. Sensor data anonymization based on genetic algorithm clustering with l-diversity. *2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT)*, pages 3–8, 2016.

[3] Olivia Angiuli and Jim Waldo. Statistical tradeoffs between generalization and suppression in the de-identification of large-scale data sets. In *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, volume 2, pages 589–593. IEEE, 2016.

[4] Yotam Aron. *Information privacy for linked data*. PhD thesis, Massachusetts Institute of Technology, 2013.

[5] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[6] Claudio Bettini, Xiaoyang Sean Wang, and Sushil Jajodia. The role of quasi-identifiers in k-anonymity revisited. *CoRR*, abs/cs/0611035, 2006.

[7] Alina Campan and Traian Marius Truta. A clustering approach for data and structural anonymity in social networks, 2008.

[8] Sean Chester, Bruce Kapron, Ganesh Ramesh, Gautam Srivastava, Alex Thomo, and S. Venkatesh. k-anonymization of social networks by vertex addition. In *In Proc. 15th Adbis (2), Volume 789 Of Ceur Workshop Proceedings*, pages 107–116, 2011.

[9] Richard Cyganiak, David Wood, and Markus Lanthaler. RDF 1.1 Concepts and Abstract Syntax. Technical report, 2014. Online; accessed 2016-12-06.

[10] Irvin Dongo, Firas Al Khalil, Richard Chbeir, and Yudith Cardinale. *Semantic Web Datatype Similarity: Towards Better RDF Document Matching*, pages 189–205. Springer International Publishing, Cham, 2017.

[11] Irvin Dongo, Yudith Cardinale, and Richard Chbeir. Rdf-f: Rdf datatype inferring framework. *Data Science and Engineering*, 3(2):115–135, Jun 2018.

[12] Martin Duerst and Michael Suignard. Internationalized Resource Identifiers (IRIs). Technical report, Microsoft Corporation, 2004.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. *Calibrating Noise to Sensitivity in Private Data Analysis*, pages 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[14] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, August 1999.

[15] Christian Fluhr. From text to rdf. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, pages 221–222, Paris, France, France, 2013. Le Centre de Hautes Etudes Internationales d'informatique Documentaire.

[16] Y. Gao, T. Luo, J. Li, and C. Wang. Research on k anonymity algorithm based on association analysis of data utility. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 426–432, 2017.

[17] P. Gayathri and V. V. Rajendran. Semantic search on summarized rdf triples. In *2017 International Conference on Intelligent Computing and Control (I2C2)*, pages 1–6, June 2017.

[18] Kalpa Gunaratna, Krishnaprasad Thirunarayan, Amit Sheth, and Gong Cheng. Gleaning types for literals in rdf triples with application to entity summarization. In *Proc. of the 13th International Conference on The SW.*, pages 85–100, NY, USA, 2016.

[19] Jianmin Han, Huiqun Yu, and Juan Yu. An improved l-diversity model for numerical sensitive attributes. In *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pages 938–943. IEEE, 2008.

[20] Kimia Hassanzadeh, Marek Reformat, Witold Pedrycz, Iqbal Jamal, and John Berezowski. T2r: System for converting textual documents into rdf triples. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 03*, WI-IAT '13, pages 221–228, Washington, DC, USA, 2013. IEEE Computer Society.

[21] Michael Hausenblas, Li Ding, and Vassilios Peristeras. Linked open government data. *IEEE Intelligent Systems*, 27:11–15, 2012.

[22] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114, August 2008.

[23] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1–2):1021–1032, September 2010.

[24] B Heitmann, Felix Hermsen, and S Decker. k- rdf-neighbourhood anonymity: Combining structural and attribute-based anonymisation for linked data. In *5th Workshop on Society, Privacy and the Semantic Web–Policy and Technology (PrivOn2017)(PrivOn), C. Brewster, M. Cheatham, M. dAquin, S. Decker and S. Kirrane, eds, CEUR Workshop Proceedings, Aachen*, 2017.

[25] Jyun-Yao Huang, Christoph Lange, and Sören Auer. Streaming transformation of xml to rdf using xpath-based mappings. In *Proceedings of the 11th International Conference on Semantic Systems*, SEMANTICS '15, pages 129–136, New York, NY, USA, 2015. ACM.

[26] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 106–115. IEEE, 2007.

[27] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 93–106, New York, NY, USA, 2008. ACM.

[28] Lian Liu, Jie Wang, Jinze Liu, and Jun Zhang. Privacy preserving in social networks against sensitive edge disclosure. Technical report, Technical Report Technical Report CMIDA-HiPSCCS 006-08, Department of Computer Science, University of Kentucky, KY, 2008.

[29] Maria Laura Maag, Ludovic Denoyer, and Patrick Gallinari. Graph anonymization using machine learning. In *Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on*, pages 1111–1118. IEEE, 2014.

[30] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 24–24. IEEE, 2006.

[31] Amirreza Masoumzadeh, James Joshi, and Hassan A. Karimi. Lbs (k, t)-anonymity: A spatio-temporal approach to anonymity for location-based service users. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 464–467, New York, NY, USA, 2009. ACM.

[32] Noman Mohammed, Rui Chen, Benjamin C.M. Fung, and Philip S. Yu. Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 493–501, New York, NY, USA, 2011. ACM.

[33] Andrew Nierman and H. V. Jagadish. Evaluating structural similarity in XML documents. In Mary F. Fernandez and Yannis Papakonstantinou, editors, *Proceedings of the Fifth International Workshop on the Web and Databases, WebDB 2002*, pages 61–66. University of California, 2002.

[34] Keiichiro Oishi, Yasuyuki Tahara, Yuichi Sei, and Akihiko Ohsuga. Proposal of l-diversity algorithm considering distance between sensitive attribute values. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2017.

[35] Vassilis Papakonstantinou, Giorgos Flouris, Irini Fundulaki, and Haridimos Kondylakis. Securing access to sensitive RDF data.

In *ESWC (Satellite Events)*, volume 8798 of *Lecture Notes in Computer Science*, pages 455–460. Springer, 2014.

[36] Peter F. Patel-Schneider Patrick J. Hayes. RDF 1.1 Semantics, W3C Recommendation 25 February 2014. `https://www.w3.org/TR/rdf11-mt/#literals-and-datatypes`, 2014. Online; accessed 2016-12-06.

[37] Jyothsna Rachapalli, Vaibhav Khadilkar, Murat Kantarcioglu, and Bhavani Thuraisingham. Redact: A framework for sanitizing rdf data. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, pages 157–158, New York, NY, USA, 2013. ACM.

[38] Jyothsna Rachapalli, Vaibhav Khadilkar, Murat Kantarcioglu, and Bhavani Thuraisingham. Rdf-x: A language for sanitizing rdf graphs. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 363–364, New York, NY, USA, 2014. ACM.

[39] Jyothsna Rachapalli, Vaibhav Khadilkar, Murat Kantarcioglu, and Bhavani Thuraisingham. Redaction based rdf access control language. In *Proceedings of the 19th ACM Symposium on Access Control Models and Technologies*, SACMAT '14, pages 177–180, New York, NY, USA, 2014. ACM.

[40] Jyothsna Rachapalli, Vaibhav Khadilkar, Murat Kantarcioglu, and Bhavani Thuraisingham. Towards fine grained rdf access control. In *Proceedings of the 19th ACM Symposium on Access Control Models and Technologies*, SACMAT '14, pages 165–176, New York, NY, USA, 2014. ACM.

[41] Filip Radulovic, Raúl García-Castro, and Asunción Gómez-Pérez. Towards the anonymisation of rdf data. In *SEKE*, 2015.

[42] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13(6):1010–1027, November 2001.

[43] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 1998.

[44] Yuichi Sei and Akihiko Ohsuga. Randomized addition of sensitive attributes for l-diversity. *2014 11th International Conference on Security and Cryptography (SECRYPT)*, pages 1–11, 2014.

[45] Yuichi Sei, Hiroshi Okumura, Takao Takenouchi, and Akihiko Ohsuga. Anonymization of sensitive quasi-identifiers for l-diversity and t-closeness. *IEEE Transactions on Dependable and Secure Computing*, 2017.

[46] Dipalee Shah and Rajesh Ingle. Privacy-preserving deletion to generalization-based anonymous database. In *Proceedings of the CUBE International Information Technology Conference*, CUBE '12, pages 459–463, New York, NY, USA, 2012. ACM.

[47] Moonshik Shin, Sunyong Yoo, Kwang H Lee, and Doheon Lee. Electronic medical records privacy preservation through k-anonymity clustering method. In *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, pages 1119–1124. IEEE, 2012.

[48] Rôney Reis C. Silva, Bruno C. Leal, Felipe T. Brito, Vânia M. P. Vidal, and Javam C. Machado. A differentially private approach for querying rdf data of social networks. In *Proceedings of the 21st International Database Engineering & Applications Symposium*, IDEAS 2017, pages 74–81, New York, NY, USA, 2017. ACM.

[49] Regina Ticona-Herrera, Joe Tekli, Richard Chbeir, Sébastien Laborie, Irvin Dongo, and Renato Guzman. *Toward RDF Normalization*, pages 261–275. Springer International Publishing, Cham, 2015.

[50] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[51] Gaoming Yang, Jingzhao Li, Shunxiang Zhang, and Li Yu. An enhanced l-diversity privacy preservation. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on*, pages 1115–1120. IEEE, 2013.

[52] Xiaowei Ying and Xintao Wu. Randomizing social networks: a spectrum preserving approach. In *SDM*, pages 739–750. SIAM, 2008.

[53] Mingxuan Yuan, Lei Chen, Philip S. Yu, and Ting Yu. Protecting sensitive labels in social network data anonymization. *IEEE Trans. on Knowl. and Data Eng.*, 25(3):633–647, March 2013.

[54] D. Zhang, T. Song, J. He, X. Shi, and Y. Dong. A similarity-oriented rdf graph matching algorithm for ranking linked data. In *2012 IEEE 12th International Conference on Computer and Information Technology*, pages 427–434, Oct 2012.

[55] Jianpei Zhang, Ying Zhao, Yue Yang, and Jing Yang. A k-anonymity clustering algorithm based on the information entropy. In *Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on*, pages 319–324. IEEE, 2014.

[56] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, pages 506–515, Washington, DC, USA, 2008. IEEE Computer Society.

Biographies

Irvin Dongo received his B.Sc. degree in Computer Science from the Catholic San Pablo University, Perú; and his M.Sc. and Ph.D. degrees from the University of Pau, France. He is currently under a postdoctoral position in Computer Science at École Supérieure des Technologies Industrielles Avancées (ESTIA). His research interests lie in normalization and anonymization of Web resources, knowledge-bases modeling (Semantic Web); policies and management of credentials, security

model and anonymization technique; and machine/deep learning techniques for an analysis and classification of data to discover patters and gesture recognition.

Richard Chbeir received his PhD in Computer Science from the University of INSA DE LYON-FRANCE in 2001 and then his Habilitation degree in 2010 from the University of Bourgogne. He is currently a Full Professor in the Computer Science Department in IUT de Bayonne in Anglet France. His current research interests are in the areas of multimedia information retrieval, XML and RSS Similarity, access control models, and digital ecosystems. Richard Chbeir has published in international journals, books, and conferences, and has served on the program committees of several international conferences. He is currently the Chair of the French Chapter ACM SIGAPP. Richard Chbeir teaches several courses in the Computer Science Department of the University of Pau University in Anglet-France.