Policy Enforcement for the Web of Data

Truc-Vien T. Nguyen and **Nicoletta Fornara** [1][2]

Abstract. Collecting data from the web is essential for research in social science, like digital marketing and applied economics, and for data research areas like Information Retrieval and Big Data. However, most of current methods for automatically collecting web data do not take into account principles of data protection policies. Therefore, it is crucial to realize a method that complies with legal, ethical, and license policies established by governments and associations. In this paper, we present a general framework for automatically collecting data from social networks which is able to preserve data privacy. We propose an ontology to capture the nature of social network data, and a procedure for translating web data into an OWL 2 ontology. Then we propose to use Semantic Web Technologies for formally expressing legal and ethical policies that regulate web data collection and mechanisms for being compliant with those policies.

1 Introduction

With the proliferation of big amount of data on the web (like social networks, blogs, and consumer reviews), web data collection is becoming increasingly important for research in social science, like digital marketing and applied economics, and for data research areas like Information Retrieval and Big Data. As networking sites become more ubiquitous, people are using them for more private and more intimate interactions but often without thinking through the privacy implications of what they are doing.

We think it's the right time to enforce the adoption of ethical guidelines and legal rules for guaranteeing integrity and transparency of the process of collecting data, and for conducting analyses that respond to privacy and confidentiality wishes of the users. Some of these ethical guidelines, laws and licenses express policies and norms[3], in particular they express obligations, having the form "we shall", and prohibitions with the form "we shall not", on how data can be collected, stored, used, and so on. Those policies are usually expressed in natural language (e.g. English): therefore, in order to comply with policies coming from different sources, researchers need to read, understand, combine, and finally apply them. But when big amounts of data are treated for automatic extraction by means of specialized software, being compliant with those policies becomes very difficult. The problem of applying these policies to the collection and subsequent use of web-based data is further complicated by the fact that top-down policies (provided by data publishers) also need to be integrated by additional (bottom-up) constraints, provided by data collectors.

In this paper, we propose to start to tackle this problem, with a specific focus on guaranteeing that the activities performed during data collection are compliant with given legal and ethical policies, and guaranteeing that the way those data are stored and re-used is also compliant with those policies. Using Semantic Web technologies, we perform automatic data extraction from social network for representing the data extracted and for reasoning on their semantics. Then we formally express policies that regulate how data is manipulated in order to be compliant with ethical guidelines and laws. The novelty of our work is twofold. First, nowadays there are no studies or tools for collecting non-reactive data from the Web that follow formal models of policy and norms representation. Second, we exploit techniques for automatic extraction and representation of knowledge from semi-structured and unstructured data in order to identify sensitive information and violated policies, following on-line research ethical guidelines and legal constrains.

This paper is organized as follows: Section 2 discusses related work and describes our motivation for policy formalization and enforcement. Section 3 presents our ontology for social network, and the procedure for translating web data into an OWL 2 ontology. In Section 4 policies for regulating data collection and manipulation, which are inspired from European Union directive on data collection, are formalized and enforced using Semantic Web Technologies. Section 5 presents the experiments we perform for investigating the applicability of the proposed framework. Finally in Section 6 some conclusions are drawn.

2 Related Work and Motivation

In this section we discuss the state of the art on formal models for norms and policies modelling and reasoning with a special focus on approaches that use Semantic Web technologies and we clarify the reasons why in the work presented in this paper we adopted and improved one of them [5, 7].

2.1 Related Work

The declarative specification of norms [6, 3] and policies [15, 14] is a widely studied concept in distributed artificial intelligence and in the specification of open interaction systems. Their declarative specification with a language with a formal semantics makes possible (i) to develop software agents able to automatically reason on them and software systems able to monitor their fulfilment or violation; (ii) in

[1] University of Lugano, Switzerland, email: thi.truc.vien.nguyen, nicoletta.fornara@usi.ch

IAT4SIS '14, August 18 2014, Prague, Czech Republic
Copyright 2014 ACM 978-1-4503-2890-6/14/08$15.00.
http://dx.doi.org/10.1145/2655985.2655988

[3] Taking into account the American tradition of using the term policies for expressing obligations, prohibitions and permissions and the European tradition of calling them norms, in this paper we will use those two terms as synonymous.

open context like in the Internet, by using a shared vocabulary, to create distributed systems where various components are developed by different companies without the need to impose constrains on their internal architecture.

Although literature shows numerous works which provide the specification of norms, very few of them adopted Semantic Web Technologies [8]. In this paper we propose to use Semantic Web Technologies for both the formalization of the data collected from the Web and the specification of the policies devised for regulating the use of such a data. This choice is due to the fact that Semantic Web Technologies are an international standard, and because, in the spirit of Semantic Web, we want to present a model of policies and the mechanisms for their enforcement that can be re-used for the specification and enforcement of other norms/policies. Moreover thanks to the fact thta OWL 2 is a decidable fragment of FOL it is possible to exploit the reasoning capabilities of one of the available OWL reasoner for checking what policies are active and for reasoning and the actions required for their fulfillment.

The policy formalization presented in this paper is strictly related to the work presented in [5, 7] where the content and the activation conditions of norms are expressed using OWL 2[4] classes that are defined using OWL axioms.

Another approach to policies specification that uses Semantic Web Technologies is the OWL-POLAR framework presented in [14]. The activation conditions of policies are expressed using *conjunctive semantic formulas*, that is conjunction of atomic assertions expressed using the concepts (classes and relations) from an OWL-DL ontology on a vector of variables. This approach presents two main drawbacks: the expressivity of the formal language chosen and the availability of tools for directly evaluating these formulas. Another limit is that for evaluating if a norm is active it is necessary to convert activation conditions into SPARQL queries and evaluate them on an OWL ontology.

In [1] access policies for regulating the use of RDF Graph Stores accessible by means of a SPARQL 1.1 endpoint are proposed. Even if this work is not focused on policy for directly regulating the behaviour of autonomous agents, it presents a proposal of formalizing access condition of policies by using as formal language SPARQL 1.1 and therefore it is interesting to compare it to the previously presented approaches. A crucial difference between this work and the previous ones is that the access conditions are focused on looking for the RDF Graphs that a user, in a given context, is allowed to access, and therefore given that the data stores are formalized using RDF Graph, SPARQL 1.1 is the language chosen for expressing those conditions.

2.2 Motivation

Several associations have established data protection policies in order to support sound and ethical practice in the conduct of survey and public opinion research. For example the American Association for Public Opinion Research (AAPOR) proposes the Code of Professional Ethics and Practices[5], which provides some guidelines for professional ethics and practices. Many data publishers have their own policies on how the resources published on their servers may be used. Examples of top-down policies of how the resources available on a social software like Facebook can be used for automatic data collection are the Automated Data Collection Terms[6]. An example of legal constrains on the processing of personal data within the European Union is the Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995[7].

In this paper we take our use cases from the EU Directive 95/46/EC that states the necessity of anonymization at point (26) defined the notion of personal data and processing of personal data in Article 2 and constraints personal data processing in Article 8, as presented in the following.

> (26) Whereas the principles of protection must apply to any information concerning an identified or identifiable person;...; whereas the principles of protection shall *not apply to data rendered anonymous* in such a way that the data subject is no longer identifiable;...

> **Article 2**
> (a) ‘personal data’ shall mean any information relating to an identified or identifiable natural person (‘data subject’); an identifiable person is one who can be identified, directly or indirectly;
> (b) ‘processing of personal data’ (‘processing’) shall mean any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction;

> **Article 8**
> 1. Member States shall prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-union membership, and the processing of data concerning health or sex life.

Starting from the aforementioned statements of the EU Directive, as a first important step we will formalize the following two policies in a way that will allow to a software tool, like a crawler, to collect data from different web sites (like for example Facebook or Twitter or Blogs) and automatically reason on the fulfillment of those policies:

1. It is obligatory to make anonymous all personal properties relating to an identified or identifiable natural person. Those properties include: username, user ID, first name, last name, full name, web site.
2. It is obligatory to prohibit the processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs.

The first policy is related to properties of a user. Whereas first name, last name belong to only people, the other properties can belong to other entities like organization, community or location in social networks. The second policy should be considered in a context of human statements or sentences, like the content of user posts, statuses, or comments on each post. We plan to formalize and automatically enforce those policies by using knowledge representation, semantic web languages and reasoning mechanism.

3 The Social Network (SN) Ontology

In this section, we describe the framework for collecting data from an online social network and for translating those data into an OWL 2 ontology, in order to be able to automatically apply data protection privacy policies to the collected data. We also propose an OWL 2 ontology to capture the categories and relationships of a generic social network.

[4] The W3C standard language for ontology specification `http://www.w3.org/TR/owl2-syntax/`.

[5] Available at `http://www.aapor.org/AM/Template.cfm?Section=AAPOR_Code_of_Ethics&Template=/CM/ContentDisplay.cfm&ContentID=4248`

[6] Available at `https://www.facebook.com/apps/site_scraping_tos_terms.php`

[7] Available at `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML`

3.1 Data Collection Procedure

We collected the data from Facebook, a popular social network with a huge amount of users. The reason for choosing Facebook is that it is one of the most popular social networking sites. Facebook reached 1.11 billion active users in the world as of March 2013[8].

We used RestFB[9] (a simple and flexible Facebook Graph API[10] written in Java) to collect users' information, statuses, posts and photos. The Graph API is the primary way that data is retrieved from or posted to Facebook. It is a low-level HTTP-based API that you can use to query data, post new stories, upload photos and many other tasks. In order to access to Facebook data via the API, the system needs to obtain an access token which provides temporary, secure access to Facebook. With the access token, the system is then able to retrieve Facebook data in terms of different fields: personal information, statuses, posts and photos.

The collection procedure is illustrated in algorithm 3.1. Starting from a set of "seed" users $\mathcal{U}$, we collected their public information, statuses, posts and photos. Then $\mathcal{U}$ is repeatedly expanded until we get some large amount of data. The expansion is realized as follow. For each user u in $\mathcal{U}$, we collect u's public statuses/posts/photos. For each status/post/photo p, we take the set of people who like p, the comments made in p, people who wrote or who liked those comments. Then we obtain a list of "new" users to add to $\mathcal{U}$. The collection and expansion are repeated until we get an amount of data, which, is defined by a threshold. The resulting dataset contains about 1611 users, 1213 posts, 1090 comments, and 6612 personal names (including first name, last name, full name, username, web site, etc).

Algorithm 3.1: SOCIAL_NETWORK_DATA_COLLECTION()

```
$\mathcal{U} = set\ of\ seed\ users$
$\mathcal{DS} = \emptyset$
$\mathcal{PS} = \emptyset$
repeat
  $\mathcal{UT} = \emptyset$
  for each $\langle user : u \rangle \in \mathcal{U}$
  do
    $\mathcal{D} \leftarrow personal\ information\ from\ u$
    $\mathcal{P} \leftarrow set\ of\ public\ posts\ from\ u$
    $\mathcal{DS} \leftarrow \mathcal{DS} \cup \{u, \mathcal{D}\}$
    $\mathcal{PS} \leftarrow \mathcal{PS} \cup \{u, \mathcal{P}\}$
    for each $\langle post : p \rangle \in \mathcal{P}$
    do
      $\mathcal{T} \leftarrow set\ of\ people\ who\ are\ tagged\ in\ p$
      $\mathcal{L} \leftarrow set\ of\ people\ who\ like\ p$
      $\mathcal{UT} \leftarrow \mathcal{UT} \cup \mathcal{T} \cup \mathcal{L}$
      $\mathcal{C} \leftarrow set\ of\ comments\ in\ p$
      for each $\langle comment : c \rangle \in \mathcal{C}$
      do
        $a :\ user\ who\ wrote\ comment\ c$
        $\mathcal{K} \leftarrow set\ of\ people\ who\ like\ c$
        $\mathcal{UT} \leftarrow \mathcal{UT} \cup \{a\} \cup \mathcal{K}$
  $\mathcal{U} = \mathcal{UT}$
until $\mathcal{DS} = limit$
return $(\mathcal{DS}\ and\ \mathcal{PS})$
```

3.2 An Ontology of Social Media Data

When web data proliferates, privacy becomes more and more an important topic surrounding social media systems. Depending on the kind of the data, setting up privacy preferences is difficult to deal with and can lead to a lot of confusion. For example, if someone posts a picture and tags his/her friends in it, each of the tagged people can contribute additional policy constraints that can narrow access to it. In open context like social networks, a more refined taxonomy is very essential and could help us move forward toward better privacy controls for online social media systems, in line with the development of Semantic Web technologies and languages.

As we are aware, there are at least two web-based ontologies: The BBC Core Concepts Ontology[11] and the SIOC (Semantically-Interlinked Online Communities) Ontology[12]. However, these two are simple ontologies which do not fit to the context of social media or blogs. Therefore we propose an ontology, designed from scratch, that can capture the nature of social media, such as users, posts, comments, categories, relationships and interactions between users. The ontology, designed using Protégé[13], is illustrated in Figure 1. The rationale behind the Social Network (SN) ontology is to provide a minimal but sufficient ontology suitable for social networking environments.

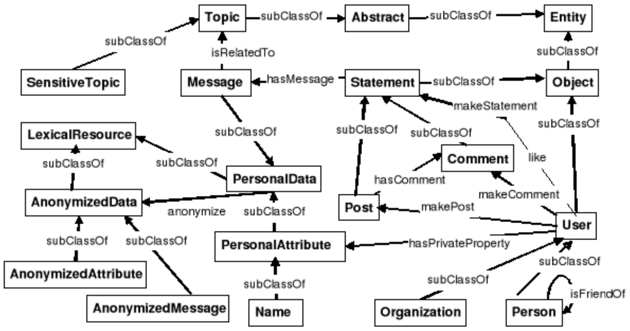

Figure 1. Social Network (SN) ontology

In the SN ontology there are two main branches: Entity → Object and LexicalResource → PersonalData. The first branch characterizes users, their statuses and interactions in the social network. A user can be a person, a location, or an organization (a community is considered as an organization). The second branch describes user's properties, such as first name, last name, username, website. User's properties are represented as binary relations. For instance, livesIn indicates that a person lives in a specific location, makePost indicates that a person makes a post, makeComment indicates that a person comments on a post.

Note that in Facebook, there are status, post and photo updates that allow users to discuss their thoughts, whereabouts, or important information with their friends. A post is created by the user (on his/her own wall or the wall of a friend), and it may include any kind of content such as shared links, checkins, photos and status updates. A status is an update posted by the user on his/her own wall. In our *SN* ontology, we use the term "post" to represent all kinds of updates, including status, photo and sharing updates.

The procedure for translating network data into an OWL 2 ontology that has as ABox the SN ontology is illustrated in algorithm 3.2. It is based on the idea of following the progress of how users interact with their network: (1) a user creates statuses, posts photos, shares links or posts information on his/her friend's wall; (2) next, his/her friends start liking the posts, commenting on them, and sharing them on their own wall. From this progress, two main steps should be taken in the algorithm: (1) create individuals in the ontology corresponding to person, posts, comments; (2) create binary relations between users/posts/comments.

[8] `http://investor.fb.com/releasedetail.cfm?ReleaseID=761090`
[9] `http://restfb.com/`
[10] `https://developers.facebook.com/docs/graph-api`
[11] `http://www.bbc.co.uk/ontologies/coreconcepts`
[12] `http://sioc-project.org/ontology`
[13] http://protege.stanford.edu/

Algorithm 3.2: NETWORK_DATA_TO_ONTOLOGY()

```
$\mathcal{U} = set\ of\ users$
$\mathcal{O} = \emptyset$
for each $\langle user : u \rangle \in \mathcal{U}$
  do
    $create\ individual\ person\ u$
    $\mathcal{O} \leftarrow \mathcal{O} \cup \{u\}$
    $\mathcal{P} \leftarrow set\ of\ public\ posts\ from\ u$
    for each $\langle post : p \rangle \in \mathcal{P}$
      do
        $create\ individual\ post\ p$
        $create\ objectProperty\ o1 = \{u\ makePost\ p\}$
        $\mathcal{O} \leftarrow \mathcal{O} \cup \{o1\}$
        $\mathcal{L} \leftarrow set\ of\ people\ who\ like\ p$
        for each $\langle person : l \rangle \in \mathcal{L}$
          do
            $create\ individual\ person\ l$
            $create\ objectProperty\ o2 = \{l\ likes\ p\}$
            $\mathcal{O} \leftarrow \mathcal{O} \cup \{o2\}$
        $\mathcal{C} \leftarrow set\ of\ comments\ in\ p$
        for each $\langle comment : c \rangle \in \mathcal{C}$
          do
            $create\ individual\ comment\ c$
            $create\ objectProperty\ o3 = \{p\ hasComment\ c\}$
            $\mathcal{O} \leftarrow \mathcal{O} \cup \{o3\}$
            $a :\ user\ who\ wrote\ comment\ c$
            $create\ objectProperty\ o4 = \{a\ makeComment\ c\}$
            $\mathcal{O} \leftarrow \mathcal{O} \cup \{o4\}$
            $\mathcal{K} \leftarrow set\ of\ people\ who\ like\ c$
            for each $\langle person : k \rangle \in \mathcal{K}$
              do
                $create\ individual\ person\ k$
                $create\ objectProperty\ o5 = \{k\ likes\ c\}$
                $\mathcal{O} \leftarrow \mathcal{O} \cup \{o5\}$
return $(\mathcal{O})$
```

3.3 Data Preprocessing

Prior to realize the policies presented in Section 22.2, the Social Network ontology has to be enriched with external knowledge. In fact, policy 1 requires the anonymization of personal data (such as first name, last name, web site). Therefore, in user's posts and comments, it is needed to: (i) recognize names appearing in their content, (ii) determine whether a name needs to be protected or not, since popular, famous names do not need to be anonymized. Policy 2 prescribes the removal of some content in case a post or a comment reveals racial or ethnic origin, political opinions, religious or philosophical beliefs. Thus it is necessary to determine if the content of posts/comments is related to such topics.

There are two main issues we need to tackle here. First, how to detect if some texts are private data or popular data. For instance, *Washington, Switzerland, Microsoft* are proper names but they are famous, popular names. Popular names may appear in the content of user's posts or comments but there is no need to make them anonymous. Second, how to detect if some content (of a post, a comment) is related to sensitive topics, including, but not limited to religion, ethnicity, or politics. If such is the case, those posts or comments shall not be stored.

Named Entity Recognition and Disambiguation

To recognize proper names in a text and to determine whether those names are popular or not, we employ Natural Language Processing (NLP) techniques, in particular we use Named Entity Recognition (NER) and disambiguation to Wikipedia (*D2W*). Named entity recognition purposes the detection and classification of text segments into pre-defined categories, such as Person, Organization, and Location. Disambiguation to Wikipedia (*D2W*) refers to the task of detecting and linking expressions in text to their referent Wikipedia pages. For instance, given a text "John McCarthy, 'great man' of computer science, wins major award.", a *D2W* system is expected to detect the text segment "John McCarthy" and link to the correct Wikipedia page *http://en.wikipedia.org/wiki/John_McCarthy_(computer_sci- entist)*, instead of other *John McCarthy* who are ambassador, senator or linguist. Since Wikipedia mainly contains popular names of people, organizations, locations, if a name can be linked to Wikipedia, it is very likely that it is a popular name. We use the *NER* system of [11] and the *D2W* system developed in [10].

Topic Detection

For the second issue, we apply a popular algorithm to detect if a given text is related to a pre-defined topic. The topic categories include, but not limited to: religion, ethnicity, politics. For each topic we construct a set of related terms. In order to determine how important a text is related to *set of terms*, we employ the popular tf-*idf* scheme [12]. It assigns to term t a weight in document d given by

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t$$

where tf stands for term frequency and idf stands for inverse document frequency. Tf-*idf* has been used by [13] to automatically identify topically related stories within a stream of news media. Following previous works, we apply tf-*idf* with set of related terms of a given topic, the content of a post/comment as a text document. Considering the *set of terms* as vectors, we use the cosine similarity between the term vector and a document vector as a measure of the score of the document for those terms.

4 Formalization and Fulfillment of Policies

In normative multiagent system literature it is widely recognized that norms and policies have the following main elements [3]: (1) they are used to define obligations, prohibitions, or permissions; (2) they may be defined in terms of roles; (3) they regulate the performance of *actions* or state of affairs; (4) they are *active* during a period of time that can be expressed through *activation* and *deactivation events* or *conditions*; (5) norms specify *sanctions* for norms violations, *rewards* for norm fulfillment; (6) finally norms are usually defined in a given *context*. Norms are usually studied from two different prospectives: (i) the development of techniques for norms enforcement and monitoring with the goal to keep the interaction among autonomous agents in open systems within certain boundaries [5]; (ii) the study of the techniques for developing agents able to reason and plan their actions on the basis of the norms that regulate their behaviour [2]. This second use is the one on which our work is focused.

In this section we specify how to formalize the obligations coming from the two data protection policies introduced in Section 2.2, we describe how to model them by extending the SN OWL 2 ontology and how to develop an agent able to reason on them and to perform the obliged actions. In particular we design an OWL 2 ontology for formally representing the obligations that come from them, that is their activation conditions, and the actions that should be performed. We take inspiration from several ontology-based access control models and knowledge representation/reasoning mechanism using OWL [7, 1, 14, 5, 4]. In particular we adapted to our current needs and extended the model of obligations presented in [5, 7].

4.1 Policies

The two policies presented in Section 2.2 can be reformulated as:

1. If a person has some personal data (such as firstname, lastname, website), then they have to be anonymized in order to store, retrieve, and use them.
2. If the content of a post, or of a comment reveals racial or ethnic origin, political opinions, religious or philosophical beliefs, then they have to be either anonymized or removed.

These policies and their consequent obligations can be formally modelled by using the application independent OWL ontology of obligations, events, and actions introduced in [5] and by extending it with the application dependent classes, properties, and axioms that are necessary for modelling the actions, events, and conditions of those specific obligations. This OWL ontology of obligations should be integrated into the Social Network ontology depicted in Figure 1 in order to completely formalize a real system able to automatically reason on the data collection process.

4.2 Modelling and Reasoning on Obligations

In Section 3.2 we presented the Social Network ontology, here we describe the classes, the properties, and the axioms necessary to model and reason on obligations. The OWL class ActivationCond-n is used to specify the activation condition of obligations, other OWL classes are used to partially express the content of obligations (i.e. the regulated actions). The advantage of this approach is that is allows to fully exploit the reasoning functionalities of OWL reasoners in the process of checking if an obligation is active and understanding which action should be performed for its fulfillment. An obligation is active if there is at least one individual that belongs to the activation condition class. The obliged actions are operations on the SN OWL Ontology. Given that it is not possible to update the content of an OWL ontology using the OWL language, we use an OWL library, OWL-API [9], for operating on the ontology from a Java program.

Policy 1 requires that personal information (such as first name, last name, web site) have to be anonymized. It includes two obligations, which requires the anonymization of (i) user attributes and (ii) personal names that may appear in the content of posts/comments.

Policy 1: Obligation-1

To deal with the first obligation, we have to take all user's personal information and then anonymize them in two steps: (i) first by asserting that those information are connected, by means of the object property $\text{anonymize}: \text{PersonalData} \rightarrow \text{AnonymizedData}$, to an anonymous individual that belongs to the $\text{AnonymizedAttribute} \sqsubseteq \text{AnonymizedData}$ class; (ii) second by taking all the users associated to those personal information and replace all the relations that associate a user to his own personal information with a relation from the user to the corresponding anonymous individual. The anonymization process is realized by a function written in Java, which, by using OWL-API, transforms a personal information into a unique anonymous attribute.

Obligation-1 is activated when in the SN ontology there is a personal information and it is not popular. The activation condition of this obligation is expressed with the following OWL class:

$\text{ActivationCond-1} \equiv \text{PersonalAttribute} \sqcap \text{hasPrivateData} \ni \text{True}$

where the $\text{hasPrivateData}: \text{PersonalData} \rightarrow \{\text{True}, \text{False}\}$ data property has been created by the algorithm described in Section 3.2. The anonymization procedure is composed by the steps below:

1. Retrieve a set $\mathcal{X}$ of user's personal information which are not popular by using the retrieve service of an OWL Reasoner and the activation condition class specified above.
2. Anonymize these information by asserting into the ontology that each information is related to a unique anonymous name that belongs to the AnonymizedAttribute class. The correspondence between names and antonymous attributes are stored in a file for possible future use.
3. Retrieve a set $\mathcal{Y}$ of users who possess the set of personal information retrieved in step (1). In order to retrieve those information we introduced into the OWL ontology the following class:
   $\text{PrivateUser} \equiv \text{User} \sqcap \exists \text{hasPrivateProperty.ActivationCond-1}$
   and then we use the retrieve service of an OWL Reasoner for getting all individuals who belong to this class, that is the users who have as private property a personal information.
4. Replace all the relations between a user in $\mathcal{Y}$ having the personal information in $\mathcal{X}$ with their anonymous individuals.

Policy 1: Obligation-2

The second obligation requires to anonymize all personal information that appear in the content of posts/comments. To do this we follow a procedure similar to the one used for fulfilling Obligation-1. However, two more steps are needed: detect personal information in the content of these posts/comments and reconstruct the content with the personal information replaced by anonymous values.

Obligation-2 is activated when in the SN Ontology there is a message (the content of a post or of a comment) and it contains personal information. This activation condition is expressed with the following OWL class:

$\text{ActivationCond-2} \equiv \text{Message} \sqcap \text{hasPrivateData} \ni \text{True}$

The procedure for fulfilling Obligation-2 is composed by the following steps:

1. Retrieve a set $\mathcal{X}$ of posts/comments which contain personal information that are not popular by using the retrieve service of an OWL Reasoner and the activation condition class specified above.
2. Detect personal information that appears in the content of these posts/comments and anonymize them by transforming each information into a unique anonymous value and reconstruct the posts/comments with the anonymized content.
3. Add to the ontology the assertions for connecting the content of a post/comment to its anonymized value by means of the anonymize property described above where the class $\text{Message} \sqsubseteq \text{PersonalData}$ and $\text{AnonymizedMessage} \sqsubseteq \text{AnonymizedData}$.
4. Retrieve a set $\mathcal{Y}$ of users who make the set of posts/comments in step (1). In order to retrieve those information we introduced into the OWL ontology the following class:
   $\text{UserWithPrivateStat}$
   $\equiv \text{User} \sqcap \exists \text{makeStatement} \, (\exists \text{hasMessage.ActivationCond-2})$
   and then we use the retrieve service of an OWL Reasoner for getting all individuals who belong to this class, that is the users who have a statement whose content belongs to the activation class of Obligation 2.
5. Replace all the relations between the posts/comments of users in $\mathcal{Y}$ having content in $\mathcal{X}$ in with their anonymized values. The replacement is done by using a Java program and OWL API library.

Policy 2: Obligation-3

In order to realize Policy 2, which states that sensitive topics in the content of posts or comments have to be removed, we take the content of posts/comments and detect their topics with the method described in Section 3.3. Then we insert the assertions in the ontology for connecting the content of posts/comments to their topic by using the $\text{isRelatedTo}: \text{Message} \rightarrow \text{Topic}$ property. Next, first we retrieve all the posts/comments which contains sensitive topics (they belong to the $\text{Sensitive} \sqsubseteq \text{Topic}$ class), second we take all the users who make those posts/comments and remove in the ontology all the assertions that connect the user to the post/comment.

Obligation-3 is activated when in the SN Ontology there is a statement (post or comment) whose content is related to a sensitive topic. This activation condition is expressed with the following OWL class:

$\text{ActivationCond-3} \equiv$
$\text{Statement} \sqcap \exists \text{hasMessage.}(\exists \text{isRelatedTo.SensitiveTopic})$

The procedure for fulfilling Obligation-3 is composed by the steps described below:

1. Retrieve the set $\mathcal{X}$ of all the posts/comments which contains *sensitive* topics by using the retrieve service of an OWL Reasoner and the ActivationCond-3 class.

2. Retrieve a set $\mathcal{Y}$ of users who make the posts/comments retrieved in step (1). In order to retrieve those information we introduced into the OWL ontology the following class:

   $\mathsf{UserWithSensitiveStat} \equiv$
   $\mathsf{User} \sqcap \exists \mathsf{makeStatement.ActivationCond\text{-}3}$
   and then we use the retrieve service of an OWL Reasoner for getting all individuals who belong to this class.
3. Remove from the ontology the assertions that connect the posts/comments in $\mathcal{X}$ to the users in $\mathcal{Y}$. The removal is done by using a Java program and the OWL API library.

5 Experiments

Our experiments aim at investigating the applicability of the method, the effectiveness of policy enforcement.

5.1 Experimental setup

We use the RestFB software to collect data from Facebook. This data, as described in section 3, includes 1611 users, 1213 posts, and 1090 comments. We transform the data into ontology using OWL API library, following the algorithm in section 3.2. The data is processed using natural language processing software as presented in section 3.3 for named entity recognition, entity disambiguation, topic detection.

5.2 Evaluation

The response time for each phase and each obligation is given in table 1. Using the PC with Intel(R) Core(TM) 2 Quad CPU Q9650 @ 3.00Ghz and 4GB RAM, our system only takes about 15 minute for translating facebook data to ontology. The realization of the obligations takes about 140 minutes, 121 minutes, 117 minutes for **Obligation-1, Policy 1**, **Obligation-2, Policy 1**, and **Obligation-3, Policy 2** on the data extracted (∼8,000 properties, ∼2,000 text contents of posts/comments), respectively.

Phase	Time (minutes)
Translation	15
Obligation 1	140
Obligation 2	121
Obligation 3	117

Table 1. Response time in each phase/obligation

6 Conclusion

The study of normative and policy-based systems and their use in different fields of application is a well-known topic of research in the Agent and Multiagent system community. However during the process of web data collection, privacy concerns have still not been studied in much depth. Regarding that collecting web data is very essential, in particular for social science fields, the idea is to encode privacy rules/laws as ontology and employ them in enforcing the rules automatically. Benefiting from the nice properties of the OWL language and its robust reasoning mechanism, the model could well enforce and apply the individual policies.

To our knowledge, this is the first framework of privacy control for web data collection, without the need for reading obligations in natural language, understanding, and finally applying them manually. Our work demonstrate that the data collection can be done in compliant with privacy enforcement in an automatic way, by exploiting the robustness of Semantic Web technologies (in particular an OWL 2 ontology) for formally expressing policies, for representing the data extracted from social network and for reasoning on their semantics.

In the future, it would be nice to improve the model to a more general extent, e.g. in order to make it usable in a general policy context by exploiting the effectiveness of OWL language and reasoner. Also, in this paper, we make use mainly the reasoner coupled with OWL-API library, we can also benefit from several query languages, such as SPARQL to encode policies and to compare with the current method.

ACKNOWLEDGEMENTS

The work described in this paper is supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) project nr. C11.0128 within the COST Action IS1004 WEBDATANET.

REFERENCES

[1] Luca Costabello, Serena Villata, and Fabien Gandon, 'Context-aware access control for rdf graph stores', in *ECAI*, eds., Luc De Raedt, Christian Bessire, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pp. 282–287. IOS Press, (2012).

[2] Natalia Criado, Estefania Argente, and Vicent Botti, 'Rational Strategies for Norm Compliance in the n-BDI Proposal', in *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, eds., Marina De Vos, Nicoletta Fornara, Jeremy V. Pitt, and George A. Vouros, volume 6541 of *LNCS*, 1–20, Springer, (2011).

[3] Karen da Silva Figueiredo, Viviane Torres da Silva, and Christiano de Oliveira Braga, 'Modeling Norms in Multi-agent Systems with NormML.', in *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, volume 6541 of *LNCS*, 39–57, Springer, (2010).

[4] T. Finin, A. Joshi, L. Kagal, J. Niu, R. Sandhu, W. Winsborough, and B. Thuraisingham, 'ROWLBAC: Representing role based access control in OWL', in *Proceedings of the SACMAT*, pp. 73–82, New York, NY, USA, (2008). ACM.

[5] Nicoletta Fornara, 'Specifying and Monitoring Obligations in Open Multiagent Systems using Semantic Web Technology', in *Semantic Agent Systems: Foundations and Applications*, volume 344 of *Studies in Computational Intelligence*, chapter 2, 25–46, Springer-Verlag, (2011).

[6] Nicoletta Fornara and Marco Colombetti, 'Specifying and Enforcing Norms in Artificial Institutions', in *Declarative Agent Languages and Technologies VI*, volume 5397, 1–17, Springer Berlin / Heidelberg, (2009).

[7] Nicoletta Fornara and Charalampos Tampitsikas, 'Semantic Technologies for Open Interaction Systems', *Artificial Intelligence Review*, **39**, 63–79, (2013).

[8] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph, *Foundations of Semantic Web Technologies*, Chapman & Hall/CRC, 2009.

[9] Matthew Horridge and Sean Bechhofer, 'The OWL API: A Java API for OWL Ontologies', *Semant. web*, **2**(1), 11–21, (January 2011).

[10] Truc Vien T. Nguyen, 'Disambiguation to Wikipedia: A Language and Domain independent approach', in *Proceedings of the 9th Asia Information Retrieval Societies Conference*, Singapore, (December 2013).

[11] Truc Vien T. Nguyen and Alessandro Moschitti, 'Structural reranking models for named entity recognition', *Intelligenza Artificiale*, **6**, (December 2012).

[12] Gerard Salton and Michael J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA, 1986.

[13] Michael Schultz and Mark Liberman, 'Topic detection and tracking using IDF-weighted cosine coefficient', in *Proceedings of the DARPA Broadcast News Workshop*, pp. 189–192. Morgan Kaufmann Publishers, Inc, (1999).

[14] Murat Sensoy, Timothy J. Norman, Wamberto W. Vasconcelos, and Katia Sycara, 'OWL-POLAR: A framework for semantic policy representation and reasoning', *Web Semantics: Science, Services and Agents on the World Wide Web*, **12-13**, 148–160, (April 2012).

[15] Gianluca Tonti, Jeffrey M. Bradshaw, Renia Jeffers, Rebecca Montanari, Niranjan Suri, and Andrzej Uszok, 'Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder.', in *International Semantic Web Conference*, eds., Dieter Fensel, Katia P. Sycara, and John Mylopoulos, volume 2870 of *LNCS*, pp. 419–437. Springer, (2003).