

Semantic Web

Final assignment

By Jelle Koster and Coen Welling
25-10-2014



UNIVERSITEIT VAN AMSTERDAM



VRIJE
UNIVERSITEIT
AMSTERDAM

Introduction

For the course Semantic Web we have tried to create a database which can view various information about Italian victims of the holocaust. The dataset we used has information about victims between the year 1943 and 1945. We decided to create an application in which the user can type in a name, and view information about that specific person. The website we have taken (<http://cdec.opendams.org>) shows all information about the several victims. We decided to focus us on some premium/important subjects of every search. In this way the user can view and select the most important information about victims of the concentration camps.

There currently isn't a way to view a clear summary of the victims in Italy between 1943 and 1945 from this holocaust. So our goal was to let people view more information about a specific person in the database, as well as a list of the connected family of the looked up person. Beneath those two tables of specific information, there is a map which will show the location of the concentration camp where the person was located.

In short, we have used two different sources of data. The first one is of course the database described as above, with the Italian victims of the holocaust from 1943 between 1945. The second database we used is DBPedia. This dataset contains an ontology with a lot of information about Wikipedia pages. We combines these two sources to show the information about victims, and with help of sgvizler we can show the information about each person in a table on the site. Underneath the tables we can show you the location of the concentration camp ,to which the user has been to, in a map. Again with help of Sgvizler.

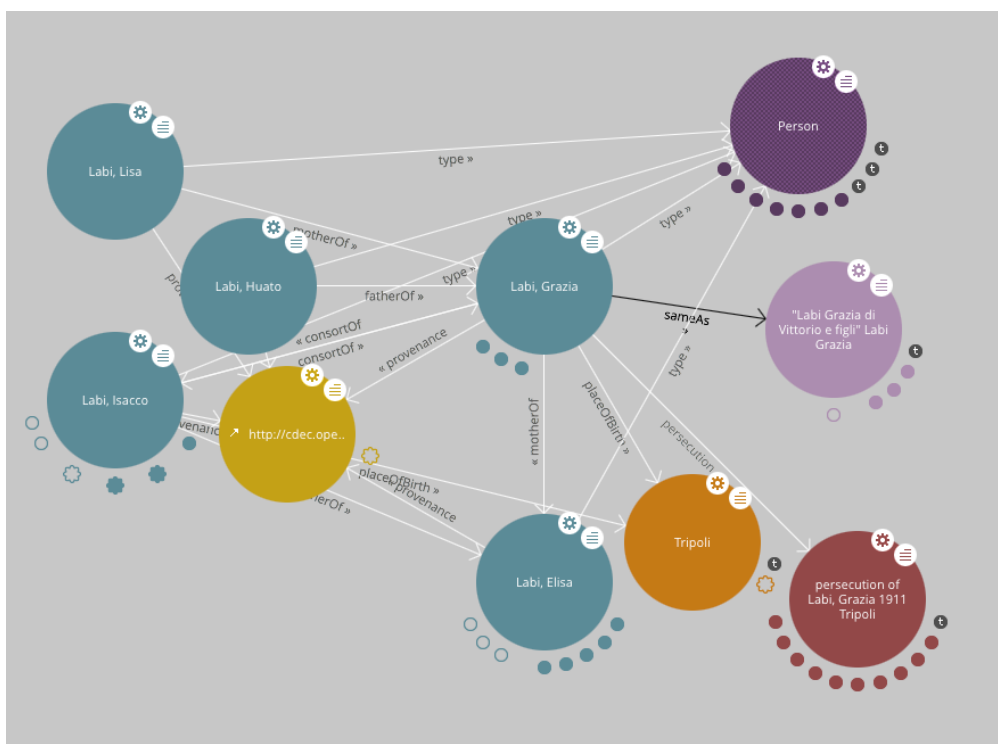
In this report we will show the steps we had to take to create our semantic web application.

Data Usage

The application we've built makes use of two different sources of data. Those are the previously described database of Italian victims of the holocaust and it makes use of LinkedGeoData. We use LinkedGeoData to show various locations of concentration camps. We thought it was interesting to take a very different subject than the ones provided on the Blackboard page of this course. After looking at the internet for quite a while, we came across the dataset of cdec.openadams.org. We found the dataset at datahub.io. This dataset had a relatively complete set of data with clear instructions of each attribute. Since we thought it was an interesting and (we presume) an uncommon dataset for this course, we decided to use this dataset. The website describes quite clear what's in the database. There is an excellent page on the website which describes the ontology of the dataset. A lot of classes and properties are defined on this page, which gave us an excellent idea how the dataset was built.

Another useful feature is the fact that the cdec.openadams.org has a data browsing Module. This data browsing module can be opened from the website. it's called *LODLIVE*. With this module you can find information from several subjects. You will start at one person or other instance. By clicking on it you will see it extends and it will show all the classes and properties it has. Once they are visible you can click on them as well to view more information about those subjects and so on. This page can also be opened by browsing the online database. In every page you will see a link to the LODLIVE module which will start a new LODLIVE page with the current page subject as a starting point.

Image 1: Example of a LODLIVE page.



Our application needs these sources in order to give the user a proper view of information of this subject. The basic information comes from the holocaust database and the LinkedGeoData can attach a visual link to the user by showing the location of concentration camps. This will give the user a good way to observe all the information without having to look up additional information from another source. Although the subject is quite specific, it will still be a nice way to view and perhaps localize victims more efficient at a single website.

Visual Application Design

The userbase for our application are humans. We did go with a plain look and feel to the website with a table to view all of our information. We believe that in this way it's still easily readable for the Human user. The results aren't given in plain HTML but with a javascript piece of software called *sgvizler*. This gave us the opportunity to create a clear view of what's in the data. Because we used tables to view the most of our data, it's as well possible to create a very similar code which let's the computer read all the data. So the focus of our user still is our human reader, but the code can easily be converted to let a machine read all of our data.

The user will first see a short introduction page with some info of the site. The user can click on a button to continue and search for victims of the shoah. After clicking the button the user is presented with a search bar in which the user can type in a name of a shoah victim. If the name, which the user has typed, is available in the in the database the website will show three tables. Each table will show different values of the person in the database. The first table will show personal information. This information consists of: *Name, Gender, Date of birth, Place of birth, Date of death, District of residence and Shoah Survivor*. The seven first attributes will view the information just as you expect, in plain text/numbers. The last attribute *Shoah Survivor* will view a 0 if the person hasn't survived the holocaust and a 1 if the person did survive the holocaust.

The second table will show several relations (if applicable) of the person in the database. This information consists of: *Names, Gender, Father of, Mother of, Brother or sister of, Consort of and Widow of*. All attributes are shown for a specific person. If, however a certain attribute isn't applicable to a certain person, the field of that attribute is seen as empty.

The third and final table will show information about the persecution of the person. This information consists of: *Persecution*, *Abstract*, *Arrest Date*, *Arrest place*, *Detention place*, *Departure date*, *Arrival date* and *Nazi camp*. *Persecution* shows in a quick way, who and from where the person is. *Abstract* will give a summary of information about the person. *Nazi camp* will show the user to which concentration camp the person has been. All other attributes speak for themselves.

The final piece of information we want to show the user is a location of the concentration camp where the person has been to. Beneath the three tables is a map. This map will show the location of the concentration camp, with a short abstract from the person who went there. This map is made with help of sgvizler and DBPedia. With some queries we extracted the latitudes and longitudes of the concentration camps as given by DBPedia. After those were obtained we entered the information in an sgvizler animation. In turn this will show us the location of the corresponding concentration camp. We decided it was essential to give the user a clear visual piece of information besides the three informative tables.

The application uses several sources of information. A lot of the data is reused. We decided to reuse as much data as possible. We also tried to make our application/data also reusable. A first way to achieve this is to have clear markers and information about our application. A second and more important way to achieve this is by creating an ontology. This ontology will show other users of our application the structure of data. Perhaps in a more easier and more understandable way than at first.

image 2: All three tables with data.

Names	Gender	Date_of_birth	Place_of_birth	Date_of_death	District_of_residence	Shoah_survivor
Labi, Grazia	F	1911	Tripoli		Teramo	1

Names	Gender	Father_of	Mother_of	Brother_or_sister_of	Consort_of	Widow_of
Labi, Grazia	F		Labi, Elisa		Labi, Isacco	
Labi, Grazia	F		Labi, Scialom		Labi, Isacco	
Labi, Grazia	F		Labi, Rubina		Labi, Isacco	
Labi, Grazia	F		Labi, Herzl		Labi, Isacco	

Persecution	Abstract	Arrest_date	Arrest_place	Detention_place	Departure_date	Arrival_date	Nazi_camp
persecution of Labi, Grazia 1911 Tripoli	Grazia Labi, daughter of Huato Labi and Lisa Labi. She was born in Libya, in Tripoli, on 1911. She was married with Isacco Labi. She was arrested in Civitella del Tronto (Teramo). She was deported to Bergen Belsen. She survived the Shoah.	30/1943	Civitella del Tronto	CIVITELLA DEL TRONTO campo	16/05/1944	20/05/1944	Bergen Belsen

Technical Application Design

To create the website as described above, we had to use several programs and queries to achieve the above result. In our webpage we included a search bar for the user. The information entered into the search bar is connected to a variable. This variable is then used in all of the three queries as described above. If the name, which is given by the user, isn't present in the database the query will not show results. The user can always try to type in a different name. To make it easier for the user, an example name is given in the search bar. An added explanation tells the user how to type a name in a correct way (Lastname, Firstname).

The way we created the queries was by using the endpoints provided by the datasets. Both DBpedia and the Shoah database have endpoints where the user can query and see the results of a query immediately. We created an sgvizler component in our website which can access those endpoints and thus we can execute our queries within our HTML page itself. The queries which were used for this application can be viewed in the appendix. An example of the source from our HTML-code can be seen below. The complete code can be seen at the source of the webpage.

Setting our prefixes and endpoint

```
<script type="text/javascript">
  sgvizler
    .defaultEndpointURL("http://lod.xdams.org/sparql#")
    .prefix('', 'http://cdec.opendams.org/lod/shoah/')
    .prefix('rdf', 'http://www.w3.org/1999/02/22-rdf-syntax-ns#')
    .prefix('rdfs', 'http://www.w3.org/2000/01/rdf-schema#')
    .prefix('gender', 'http://xmlns.com/foaf/0.1/gender')
    .prefix('familyName', 'http://xmlns.com/foaf/0.1/familyName')
    .prefix('abstract', 'http://purl.org/dc/elements/1.1/abstract');
</script>
</head>
```

Here the value from the text box is put into the query

```
<input type="text" id="name" value="Labi, Grazia">
$(document).ready(function(){
  $("button").click(function(){
    $("#sgvzl_personal_data").attr("data-sgvizler-query", "SELECT ?Names
?Gender ?Date_of_birth ?Place_of_birth ?Date_of_death ?District_of_residence ?
Shoah_survivor WHERE{ {?personID rdfs:label '' + $('#name').val() + ,,},.....
```

Ontology

The domain of our ontology is ,as said above, Italian victims of the Holocaust between 1943 and 1945. We have used a few queries to get information out of the database from cdec.openadams database. We decided to divide the ontology in 4 major classes. These classes are:

- **Persecutions:**
 - describes information about when and where a person is located to an extinctioncamp.
- **Persons:**
 - Describes personal information about the person.
- **Locations:**
 - All the locaties of possible extermination camps.
- **Relations:**
 - All the relations a victim could have with another victim based on family relations.

Every class above exists of several subclasses:

- **Persecutions:**
 - detentionPlace, arrestPlace, abstract, arrivalDate, toNaziCampLabel, arrestDate, persecution, departureDate
- **Persons:**
 - gender, shoahSurvivor, dateOfDeath, districtOfResidence, placeOfBirth, dateOfBirth
- **Locations:**
 - Auschwitz, Bergen Belsen, Buchenwald, Dachau, Flossenbug, Fort Breedonk, Golling, Incerto, Innsbruck, Mannheim, Markt Pongau, Meppen, Natzweiler, Pulenkircher, Ravensbrueck, Reichenau, Viaggio
- **Relations:**
 - brother_sisterOf, consortOf, fatherOf, motherOf, widowOf

