Semantic Web Technology Project

Automatic Family Tree Generator using DBpedia Data

Group members:

Chris Pool	S2816539
Leonardo Velozo	S1668501
Ruben de Jong	S2616726

File repository: https://github.com/chrispool/semWeb-project

Resources:

Table of Content

1. Introduction

This report is part of the Semantic Web Technology course 2015. The Semantic web aims to enrich data knowledge. The current World wide web is intended for human readers and has a focus on visual presentation. Moreover, HTML does not make it easy to obtain the logical structure/meaning of a page. One other issue is that creators have a lot of freedom, this generates inconsistency in coding. The Semantic web tries to extend the current web giving it a well-defined, consistent meaning, better enabling computers and people to work in cooperation.

For our project we aim to create a program that automatically generates family trees using the data from DBpedia using the properties “Spouse, Children (of), mother (of), father (of) etc.”(Info boxes Wikipedia) and if necessary information extracted from the corresponded Wikipedia text (Wikipedia abstract). Our goal is to enrich the knowledge of family relationships, giving inversed properties like mother-son and uncle-nephew relationships and also inferring subproperty relationships like son-of is a sub-property of child-of. This results is an enhanced family tree and its family relationships will be summarized in a graphical representation.

Topic / Research question
A software generated family tree from existing Wikipedia and DBpedia data, including and adding entities (persons) that do not have a direct link made to each other and adding the different kind of relationships between these entities.

Content
Firstly we determine the scope of the project using existing literature on Entity recognition and information extraction from Wikipedia. Secondly we discuss the “Tools” we use to create the *Family Tree Generator* (FTG) and the processing of the FTG. Thirdly we evaluate the results of the FTG followed up by discussing the newly generated information and draw conclusions based on these results.

2. Literature

The general idea behind information extraction (IE) is automatically retrieving certain types of information from natural language text. Therefore, IE is a form of natural language processing in which certain types of information must be recognized and extracted from text. (Wimalasuriva & Dou, 2010) Our project is a good example of an IE system. It is a system that processes a database and web pages and extracts information regarding persons (entities) and their family ties. To do this a model has been made to guide this process. The System will attempt to retrieve information matching this model and ignore other types of data.

To put our system into perspective we need to know where IE is positioned. Russell and Norvig (1995) state that IE lies mid-way between information **retrieval (IR) systems**, which only find documents that are related to the user's requirements, and **text understanding systems** that attempt to analyze text and extract their semantic contents. Studies on IR have produced many productive systems, such as web-based search engines, while text understanding systems have not been that successful. Since the difficulty associated with IE systems lies in between these two categories, their success has also been somewhere in between the levels achieved by IR and text understanding systems.

Ontology-based information extraction (OBIE) has recently emerged as a subfield of IE. (Hoffman & Weld, 2008) Here, ontologies are used by the information extraction process and the output is generally presented through an ontology. Since OBIE is a subfield of IE, which is generally seen as a subfield of natural language processing, it is reasonable to limit the inputs to natural language text. Russell and Norvig state that they can be either unstructured (e.g. text files) or semi-structured (e.g. web pages using a particular template such as pages from Wikipedia). Systems that use images, diagrams or videos as input cannot thus be categorized as OBIE systems.

Since our program is characterized as an OBIE system the following potential benefits are listed by Wimalasurva & Dou (2010):

1. *Automatically processing the information contained in natural language text*
2. *Creating semantic contents for the Semantic Web*
3. *Improving the quality of ontologies*

Especially improving of the quality of ontologies is represented in our system, although

the success of the Semantic Web relies heavily on the existence of semantic contents that can be processed by software agents, the correctness of this content is also very important.

3. Method

The goal of our project is to create a family tree based on the information on DBpedia and Wikipedia. We create two different family trees, one based on the *father* and *mother* properties on DBpedia and the second family tree based on the text available on Wikipedia. We use the DBPedia tree as our gold standard to evaluate how good our Wikipedia family tree is. We can use the DBpedia tree to create new properties like grandmother, grandfather, aunt and uncle.

DPBedia tree

This tree is based on the DBpedia properties *Father* and *Mother*. We enter a DBpedia URI of a person and our system recursively retrieves the father and mother of that person and continues to retrieve the parents of the father and mother of that person. The number of people added each iteration grows exponential so we limit the number of iterations to 128 (7 generations)

We use an object oriented approach where each person is an object that has the properties father and mother and is added to the tree object. We can use these objects to create the tree when processing is done.

To retrieve the information from DBPedia we use the SparqlWrapper[1] library that helps in making Sparql queries for DBpedia, we use a very simple query:

```
self.wrapper.setQuery("""
                PREFIX db: <http://dbpedia.org/resource/>
                select ?property ?value
                where {
                {
                  """ + resource + """ ?property ?value.
                }

                }
                """)
```

[1] https://rdflib.github.io/sparqlwrapper/

to retrieve the information. The resource variable contains an uri of a person, for example *beatrix_of_the_netherlands.* As result we get an object where we look for the properties *Father* and *Mother*.

Wikipedia tree

We also create a family tree based on WIkipedia text to retrieve relationships that are not in DBPedia to improve DBpedia. There are cases where the information about parents is in the text but not in the infoboxes. With automatically retrieving this information from the text we try to improve the information on DBpedia. For creating this tree we download the page of each person using the Python Wikipedia library[2]. This library downloads the entire page and formats it to plain text. In this text we look for sentences that contain 'daughter of', 'son of', 'child of'. We then use the Stanford NER tagger[3] to tag the named entities in these sentences. Entities of the type person that occur after the substring are seen as the father or mother. With this very simple approach the number of true positives are high but also the true negatives are quite high because sometimes other family members are mentioned as well.

Also the Stanford NER tagger has difficulties with names of royals as you can see in the next example:

Princess Beatrix of the Netherlands (Beatrix Wilhelmina Armgard, Dutch pronunciation: [ˈbeːjaˌtrɪks ˌʋɪlɦɛl ˈmina ˈɑrmɣɑrt] (listen); 31 January 1938) reigned as Queen of the Netherlands from 1980 until her abdication in 2013. Princess Beatrix is the eldest daughter of Queen Juliana and her husband, Prince Bernhard of Lippe-Biesterfeld. Upon her mother's accession in 1948, she became heir presumptive. When her mother abdicated on 30 April 1980, Beatrix succeeded her as Queen.

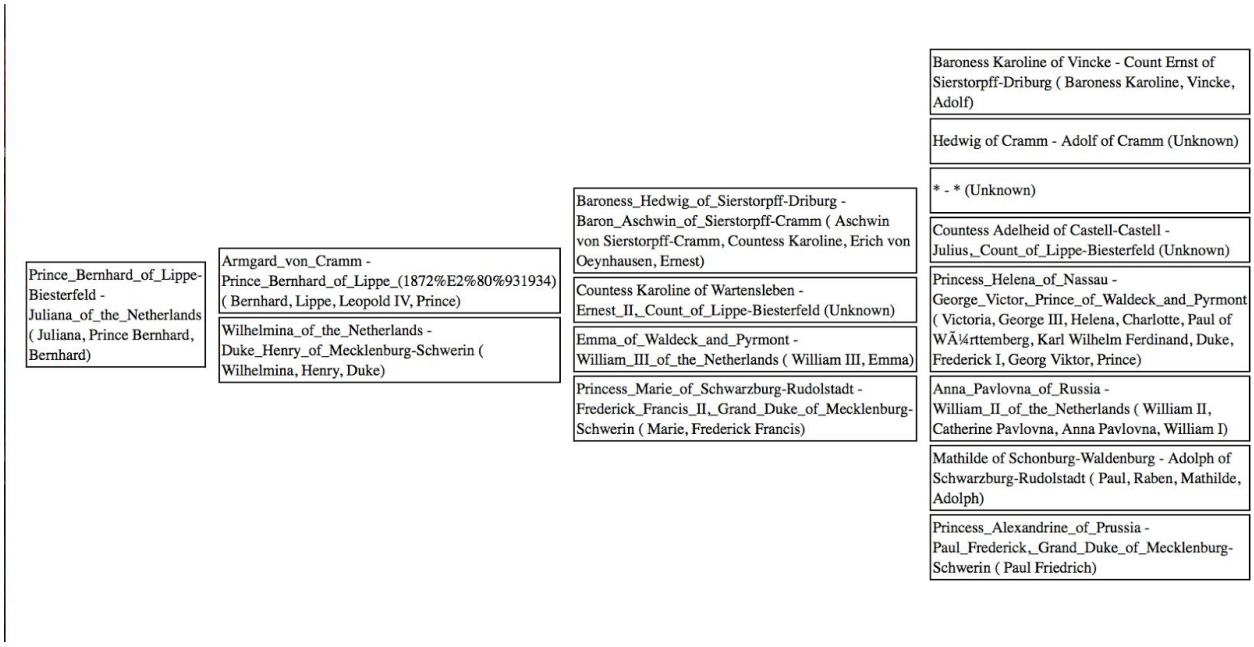
Potential tags:
ORGANIZATION
LOCATION
PERSON

This is an example where Lippe-Biesterfeld is classified as a location where it should be classified as a person. In cases where there is a structure 'Prince X of Y' the X is often correctly classified but the Y is often seen as a location. These names are quite common in royal families, one way to solve this is to train a classifier ourselves using (a part of) our gold data. The results of this method are described in the next chapter.

[2] https://pypi.python.org/pypi/wikipedia/
[3] http://nlp.stanford.edu/software/CRF-NER.shtml

4. Results

Both family trees are exported to a HTML file that makes it easy to evaluate the results. Below are the results for the family of Princess Beatrix of the Netherlands.



The first two names are the names from the DBpedia tree, the names between parenthesis are retrieved from the Wikipedia page. We manually evaluated these results with these rules:

- We ignore duplicates
- Missing titles like Queen, King, Prince, Counts are not seen as wrong
- Cases where most of the names overlap are seen as correct.

We evaluate with three persons and only when at least two judges agree we use that score. We give a point when one of the names of gold standard occurs in our Wikipedia Tree. Because of this method our score should not be seen as the accuracy but as the precision. Calculating the recall is more challenging because of the duplicates.

We have 442 persons in our family tree derived from DBpedia and we found using Wikipedia 161 of them. That results in a score of 36%

The complete evaluation and output of our system can be found in our GitHub repository.

5. Discussion

Our Automatic Generating Family Tree program can be used as starting point for an improved and extended program in the future. At this stage the program extracts the mother and father properties from DBPedia and creates a family tree using only this information. In addition the program extracts mother and father relationships from the corresponding Wikipedia text by looking for named entities in sentences with ‘daughter of’, ‘son of’ or ‘child of’ in it. By doing this we aim to enhance the information in the info boxes with automatically extracted information from the Wikipedia text. There are many possible improvements to our program. The most important improvements will be summed up as follows.

A first possible improvement is to extend the program so that it looks in both directions of the family tree. So, not only looking at older family member entities (mother / father relationships) but also at younger ones (children relationships). Another related improvement is to extract more family link information by applying logical reasoning. Relations that can be extracted in this way are for example aunt, uncle, nephew, niece, grandparents and grandchildren.

Thirdly a good working program can check the correctness of existing relations in the information boxes by comparing them to the information extracted from the Wikipedia text. This will improve the quality of the knowledge of family relationships. Another important improvement is finding the correct DBpedia page from the predicted entities extracted from the Wikipedia text. The name in the text does not correspond often to the name of the DBPedia link (URI), and therefore it complicates to link entities from the text to the right corresponding DBPedia page.

The last two possible improvements have to do with User Interface of the program. The output of the program is produced with HTML tables. In the future it would be nice to create an interactive User Interface in which first the most basic information is shown and secondly the user can choose part of the family tree to see more information about it. The last improvement is to extend the information of each entity in this HTML output. An example is to add a corresponded picture and some background information of the entity.

References

Russell, S., & Norvig, P. (1995). Artificial Intelligence: A Modern Approach, (Prentice-Hall, Englewood Cliffs, NJ) 848–850.

Wimalasuriya, D. C., & Dou, D. (2010). Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science*.

Wu, F., Hoffman, R., Weld, D.S. Information Extraction from Wikipedia: Moving Down the Long Tail. New York, ACM 2008.