Extracting Topic-based Microdata from WebDataCommons

Alexis Bonnin, Esteban Launay and Sacha Loriot

UFR de Sciences et Techniques, Nantes University

Abstract. With the explosion of the Web 2.0 (social web), the amount of websites hosted online became out of control. Crawlers (software designed for exploring the entire web and retrieve webpages) are no longer adequate to search specific topic. This paper is proposing a process to retrieve a basic data set of entity (associated to an URI) related to a city. Afterwards, this set is ready to train a different kind of crawler : Focused Crawlers.

Their goal is to decide if a webpage is relevant to the specified subject. Avoiding parts of the Internet, they are less time and resources consuming.

In order to achieve this task they need what is called a classifier. This one relies on the starting set mentioned earlier to decide which webpage collect or not.

Keywords: N-Quad, Crawler, Focused Crawler, Microdata, ontology

1 Introduction

Crawling over the web is returning a tremendous number of webpages. To give you a figure, we gathered more than **900** GB of data from a crawl performed in 2016 by the WebDataCommons[1] project. This is really difficult to detect if a topic is having the interest of people or simply discussed considering that ton of information. Often worried about their numeric identity, cities like Nantes (France) wonder if their popularity on the web is evolving towards the right direction. So we undertook to build a data set referring to Nantes keeping in mind the possibility to, later, train a focused crawler. We show in this paper an approach to construct that set from data collected by WebDataCommons project. Indeed, this one is crawling the Common Crawl[2] searching only for webpages with embedded Microdata. Introduced with HTML 5, they allow to describe the content of a webpage. From that, we are able to constitute a strategy to identify data about Nantes.

[1] WebDataCommons : http://webdatacommons.org/
[2] Common Crawl : http://commoncrawl.org/

2 Related works

Our work follows the "*Open data* +" project [1] which propose a method to perform a pruning over a huge amount of data retrieved from the Web Data Commons. That pruning is realised using Scala[3] language by searching data matching with strings. However, it can be enhanced by working on a "decision tree" (detailed on section 4). Since, their core procedure concerning data treatment is well established, we are using it.

3 Problem presentation

The Common Crawl Foundation is a California registered non-profit which produces and maintains an open repository of web crawl data that is universally accessible and analyzable. The objective of our work is to find the domains with topic about Nantes from these data. The *Open data* + report showed that it was possible to get the websites which talk about Nantes using Web Data Commons project. This one performs a crawl on the Common crawl to extract website which contain microdata. The crawl is returning a file containing N-Quads. The main issue is to be certain that a domain (contained in a N-Quad) is really about Nantes. The approach of *Open data* + project and even our work is based on the microdata of the Web Data Commons.

Each dataset of the Web Data Commons contains more than 24 billion triples to analyse, it is 100 GB for 2012 and 1000 GB for 2016. Because of the large number of triple, it cannot be done into a basic semantic store so, the execution time must be taken in account. Furthermore, since we have no power on microdata filled by website developers, we suppose them correct and accurate. Finally, one of the main constraint is that we do not know which ontology we have to use. So, we have to define what is Nantes in a web semantic term. This problem is known as the problem of blind men and elephant. As describe in the picture 1, blind men are trying to learn what they are touching. This is an analogy with the problem of defining Nantes because we have separated data and we do not know what are exactly the Nantes' properties, and all microdata from a same domain are separated and cannot be analyse as a group.

_:node42f465294ae43678bfc79db3dac5
<http://www.w3.org/2006/vcard/ns#locality>
"Nantes" <http://fr.nomao.com/4494771.html> .

_:node42f465294ae43678bfc79db3dac5
<http://www.w3.org/2006/vcard/ns#country-name>
"France"
<http://fr.nomao.com/4494771.html> .

[3] Scala : https://www.scala-lang.org/

_:node42f465294ae43678bfc79db3dac5
<http://www.w3.org/2006/vcard/ns#postal-code>
"44300"
<http://fr.nomao.com/4494771.html> .

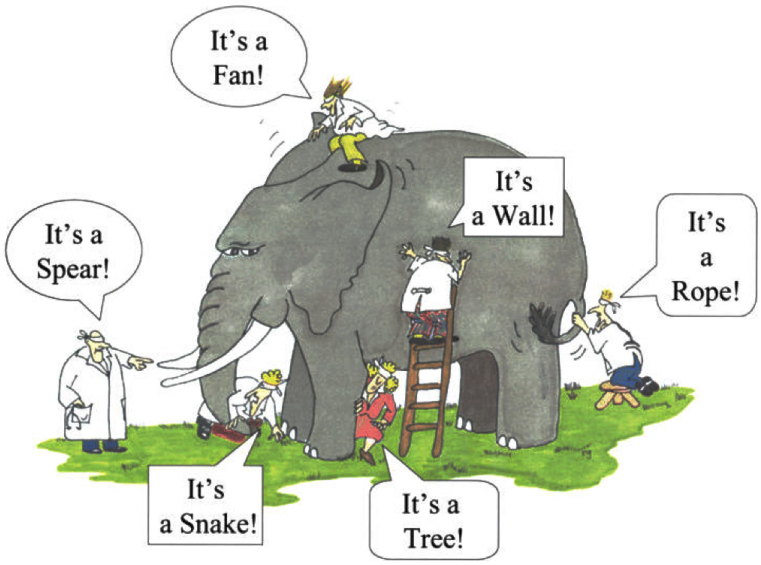

Fig. 1. *Blind Men and Elephant*

To achieve this project we assume that the data are were well formatted which means that it respects some well known ontology as vCard or schema.org.

4 Approach

We need to extract entity with microdata related to Nantes. The first thing to do is to find properties that could refer to Nantes. We choose three properties that are essential to define Nantes like its postal code, if it contains Nantes in it or its geodata. But there are some problems, the data retrieved with this method is uncertain : an entity with a postal code equals at "44300" is selected but we do not know its country and it can be the United States, so this is not Nantes related.

For example, we can retrieve N-Quad like this (simplified format):

- _:node42 postalCode 44300 http://fr.nomao.com
- _:kichler postalCode 44300 http://kichler

But the result is uncertain, we need complementary informations.

So, we have to define what is certain and not. We defined a basic decision tree which allows to collect as much microdata about Nantes as possible :

- Postal code = "44xxx" : uncertain
- Postal code = "44xxx" AND country = "France" : certain
- URL like "Nantes" : uncertain
- URL like "Nantes" AND country = "France" : certain

The first step of the process is a simple filter to get all the data that may be Nantes, we just check the postcode or if it contains "Nantes". It is evaluated on each entry of the data set :

_:node42 postalCode 44300 http://fr.nomao.com
_:kichler postcalCode 44300 http://kichler

The following algorithm was used to perform the first step :

Data: Line currentLine, File file, Results res
for *each currentLine in file* **do**
  if *currentLine.contain("Nantes") || currentLine.contain("postalcode> 44")* **then**
    res.add(currentLine)
  end
end

Algorithm 1: Step one

Then, the second step consists in aggregating the results by domains to simplify the run with the decision tree.

_:node42 postalCode 44300 http://fr.nomao.com | _:node42 country France http://fr.nomao.com _:kichler country USA http://kichler | _:kichler postalCode 44300 http://kichler

The last step consists in running the decision tree for each aggregate domain to validate the microdata obtained in the previous steps.

_:node42 postalCode 44300 http://fr.nomao.com|country France http://fr.nomao.com
:kichler postalCode 44300 http://kichler|:kichler country USA http://kichler

The following algorithm was used to apply the decision tree on the dataset.

Data: Line currentLine, File step1, Results res
for *each currentLine in step1* **do**
  if *currentLine.contain("country-name> France")* **then**
    res.add(currentLine)
  end
end

Algorithm 2: Step three

4.1 Experiment

As previously stated, we used Scala in addition of Spark[4] to analyse the data. The process used in this experiment is the one describe in the section 4, but, unlike the data for 2012, the data for 2016 are twice large and cannot be processed with the current resource. That's why the stage two has to be modified by splitting the data in different directories. Indeed, Spark is able to treat the data for each directory separately. After processing and retrieving the right data, we put them into a semantic store to process Sparql[5] query to analyse the results. We also compare the evolution of uses of micro-data since 2012.

5 Experimental study

The goal of this experimental study is to retrieve from the WebDataCommons all the microdata relative to Nantes of the year 2012 and 2016. By comparing the two results we can observe the evolution of the situation of Nantes into the numeric world.

5.1 Setup presentation

For our study, we use a setup which is the same as in the Capstone project but with little differences. Like the Capstone project, we used Apache Spark to process the data analyse and powered it with Scala. However, we do not make use of Apache Spark to run it on a distributed file system like Apache Hadoop[6].

The setup used to perform the different steps of the process are the following :

- RAM : 500 GB
- CPU : 64 cores

5.2 Results & interpretations

Execution time The figure 2 presents the different execution time of stages for the data of 2012 and 2016. We ran these stages for the city of Nantes. In general, the total execution time is around 5 hours using Apache Spark on a super computer. The stage 2 is longer because the script treats at the same time the raw data and the result of the stage 1.

Evolution of data (2012-2016) Once the data are processed, it has to be structured as N-Quads format : subject, predicate, object and graph, with correct syntax. So, some corrections may be needed following the previous process. The figure 3 presents some comparisons between results extracted in 2012 and 2016. In general, we can observe that there is around half information less about Nantes in 2016 than in 2012.

[4] Spark : http://spark.apache.org/
[5] Sparql : https://www.w3.org/TR/rdf-sparql-query/
[6] http://hadoop.apache.org/

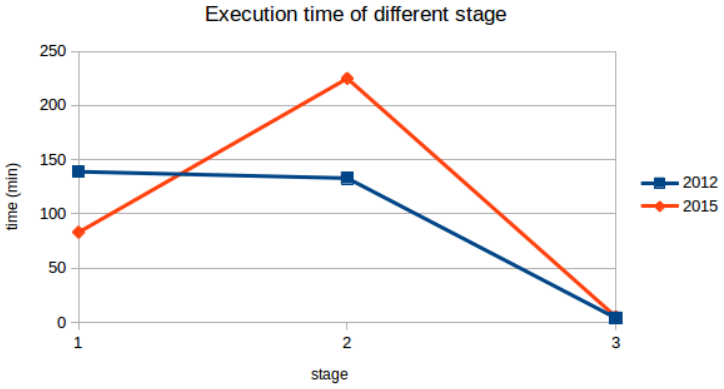

Fig. 2. *Execution time (min) for 2012 and 2016 data for each stage.*

Category	2012	2016
Result data file size (MB)	13.4	6.6
Number of N-Quads	79 651	37 051
Number of different graph	333	321

Fig. 3. Evolution of data between 2012 and 2016

Analyse from query on results data The first query (query 1.1) allows to count the total number of graph (website domain). This query was run on 2012 and 2016 data and is used for the statistics presented above (figure 3).

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?g (COUNT(?g) AS ?count)
{
    { ?s ?p ?o } UNION { GRAPH ?g { ?s ?p ?o } }
}
GROUP BY ?g
```

Query 1.1. Count the number of graph

The most used predicate analyse is computed with the query 1.2. We can observe that the most used ontology is vcard from w3c, in the 2012 and 2016 data. Then, we can see that schema.org is much used but the ontologies are more diversified.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?p (COUNT(?p) AS ?count)
{
```

```
    { ?s ?p ?o } UNION { GRAPH ?g { ?s ?p ?o } }
}
GROUP BY ?p
```

Query 1.2. Count the number of predicate

The last query presented in this paper (query 1.3) is about all the websites whose country are not France. This query could show where the topic Nantes is used in the world.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
PREFIX schema: <http://schema.org/>

SELECT DISTINCT ?g ?country (COUNT(*) AS ?count)
WHERE
{
        GRAPH ?g {
                ?s vcard:country-name ?country.
                FILTER NOT EXISTS {
                        FILTER (regex(?country,
                                "France","i")) .
                }
        }
}
GROUP BY ?g ?country
```

Query 1.3. Count the number of country which are not France.

There is no predicate "about" so we cannot determine which topic is the most frequent with a query. However, with the classification of the most frequent website using the result of the query 1.1 we can observe (by reading the name of the website) that the most frequently topic is about tourism or event in Nantes.

6 Conclusions

In this paper, we explained the various problems about processing a Topic-based search without having a set of clear data. We proposed a method to bypass this problem. The decision tree is essential in this process and have important running time.

To go further, we could use this first step to train a focused crawler to get more data which are related to Nantes but cannot be found with our first decision tree.

Bibliography

[1] A. Benabadji, H. Benyahia, A. Boussalem, T. Couraud, P. Gaultier, L. Lucas, and T. Minier. Senior capstone project open data +. 2017.

[2] R. Meusel, P. Mika, and R. Blanco. Focused crawling for structured data. 2014.