WebSmatch: a tool for Open Data

Emmanuel Castanier[(a)], Remi Coletta[(a)], Patrick Valduriez[(a)],
Christian Frisch[(b)]
(a) INRIA and LIRMM, Montpellier, France (b) Data Publica, Paris, France
(a) {FirstName.Lastname}@inria.fr (b) christian.frisch@data-publica.com

ABSTRACT

Working with open data sources can yield high value information but raises major problems in terms of metadata extraction, data source integration and visualization. In this paper we describe a demonstration of WebSmatch, a flexible environment for Web data integration, based on a real, end-to-end data integration scenario over public data from Data Publica [1]. The demonstration focuses on poorly structured input data sources (XLS files).

1. INTRODUCTION

Recent open data government initiatives, such as `data.gov`, `data.gov.uk`, `data.gouv.fr` promote the idea that certain data produced by public organizations should be freely available to everyone to use and republish as they wish. As a result, a lot of open data sources are now available on public organization's web sites, in various formats.

A large share of the available open data comes from large institutions (such as Eurostat, World bank, UN....) using structured data formats such as SDMX or RDF. However, the majority of the data that can be found on open data portals is available as unstructured data (such as spreadsheets).

2. DEMONSTRATION SCENARIO

In this section, we describe a real example of WebSmatch usage. Data Publica provides more than 12 000 files of public data [1]. However, even though data formats become richer and richer in terms of semantics and expressivity (e.g. RDF), most data producers do not use them much in practice, because they require too much upfront work, and keep using simpler tools like Excel. Unfortunately, no integration tool is able to deal in an effective way with spreadsheets. Only few initiatives (OpenII [3] and Google Refine [2]) deal with Excel files. However, their importers are very simple and make some strict restrictions over the input spreadsheets.

Input files

For simplicity purposes, the scenario of this example involves only 2 data sources. To be representative of real-life public data, we choose two spreadsheet files:

`http://www.data-publica.com/publication/1341` is an Excel file. It contains data from the Climatic Research Unit (`http://www.cru.uea.ac.uk/`) about the temperature evolution in the world over the last decades. This file is quite well formed, it only contains some blank lines and comments.

"BP Statistical Review of World Energy 2011" [3] made by BP (`http://www.bp.com/`). This spreadsheet is much more complex: it involves several sheets, with several tables per sheet. It contains several blank lines and comments, making it hard to automatically detect the table.

3. DEMONSTRATION

WebSmatch workflow consists in 3 important steps (see Figure 1): detection of tables in XLS file, detection of data and metadata in each table and then matching with existing concepts (like the DSPL ones [4]) in the system (a concept contains all needed informations like : name and instances).

Metadata Detection

It is important to note that Excel files (such as .xls, for which there is no XML version) are not structured at all. They can contain lots of artifacts such as blank lines, blank columns, titles, comments, and not only a simple table.

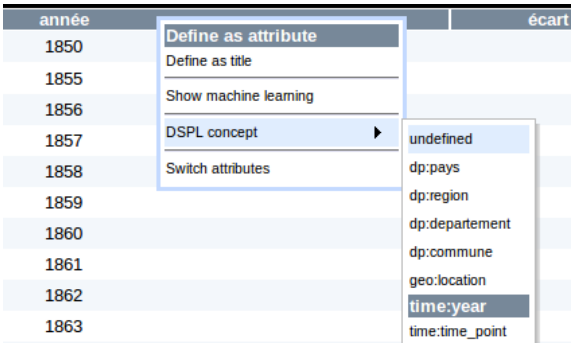

Figure 2: Table and metadata detection

To get all the metadata and data, the chosen file is parsed and then, two processes are applied to it. The first process relies on a combination of computer vision algorithms [2].

The second process uses past experience based on several criteria: the discrepancy measures, the datatype of a cell,

[1] `http://www.data-publica.com`
[2] `http://code.google.com/p/google-refine/`
[3] `http://www.bp.com/assets/bp_internet/globalbp/globalbp_uk_english/reports_and_publications/statistical_energy_review_2011/STAGING/local_assets/spreadsheets/statistical_review_of_world_energy_full_report_2011.xls`
[4] `http://developers.google.com/public-data/`

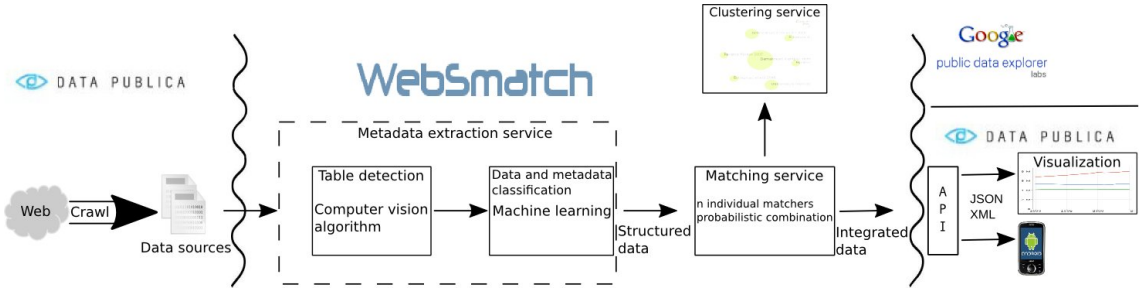

Figure 1: WebSmatch complete process

the data type of the neighborhood of a cell. WebSmatch detects each important component in the spreadsheet file such as: titles, comments, table data, table header (see Figure 2). Machine learning is able to capture several spreadsheet users habits, such as: "if cells on the very first line of a connected component have string datatype and cells of the second line have a numeric datatype : first line contains metadata" or "cells having the string datatype and void neighborhood and behind a table often are a title".

Matching

WebSmatch relies on YAM++ to perform the matching task. YAM++ combines 14 different matching techniques, divided in 3 main groups: string matchers, dictionary and thesaurus matchers based on Wordnet[5] and instance-based matchers. YAM++ powerful instance-based matcher is one of the main reasons for YAM++ excellent results (first position) at the 2012 competition of the Ontology Alignment Evaluation Initiative (`http://oaei.ontologymatching.org`).

Figure 2 show the cell "année" (i.e. year in french), which has been previously detected as metadata. This cell is detected as "time:year" concept by applying the instance-based matcher on its data collection $\{1990, 1991, \ldots\}$.

Figure 3 show the detection of a "dp:pays" concept. User is also able to replace an incorrect instance with the good one.

Figure 3: Replace instance after concept matching

Visualization

By detecting blank cells, we are able to convert bi-dimensionnal tables from the initial spreadsheet into classical (SQL-like) flat tables. Thanks to the matching process, we are also able to identify concepts (from DSPL) over the data sources and to detect common attributes in order to produce integrated data.

At this step, we have distinguished data and metadata from the initial Excel files, and flatted bi-dimensionnal tables.

[5] `http://wordnet.princeton.edu/`

We can easily generate an XML file describing the metadata (title, header, concepts) and the `.csv` files containing the data to fit the strict DSPL input format. As a result, we can take advantage of powerful visualization tools like Data Publica's one (Figure 4) or Google Data Explorer.

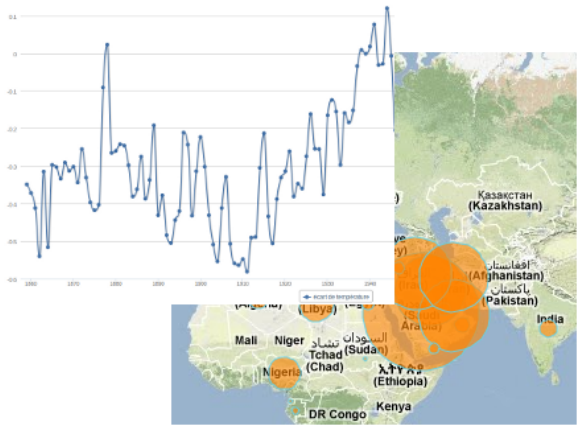

Figure 4: Vizualisation with Data Publica's tool

4. CONCLUSION

In this paper, we described WebSmatch, a flexible environment for Web data integration, based on a real data integration scenario over public data from Data Publica. We chose a typical scenario that involves problems not solved by currents tools: poorly structured input data sources (XLS files) and rich visualization of integrated data. WebSmatch supports the full process of importing, refining and integrating data sources and uses third party tools for high quality visualization and data delivery. Working on such data sources with WebSmatch will require only a minimal manual effort (such as replacing badly detected instances). The video of the whole demonstration is available at `http://www.youtube.com/watch?v=A8ho3hu2vOU`.

5. REFERENCES

[1] F. Bancilhon and B. Gans. Size and structure of the french public sector information. In *gov.opendata.at*, 2011.

[2] R. Coletta, E. Castanier, P. Valduriez, C. Frisch, D. Ngo, and Z. Bellahsene. Public data integration with websmatch. *WOD*, 2012.

[3] L. Seligman and al. OpenII: an open source information integration toolkit. In *Int. SIGMOD Conference*, pages 1057–1060, 2010.