Efficient Search and Browsing of CSV Datasets

Agata Filipowska[1,2], Krzysztof Węcel[1,2], and Dominik Filipiak[1,2]

[1] Department of Information Systems
Faculty of Informatics and Electronic Economy
Poznań University of Economics
a.filipowska; k.wecel@kie.ue.poznan.pl
[2] Instytut Informatyki Gospodarczej Sp. z o.o.
ul. Rubiez 12G/6, 61-612 Poznań, Poland

Abstract. The paper presents an application developed within the FP7 LOD2 project that supports efficient search and browsing of CSV files. The application indexes CSV files using DBpedia categories and sophisticated strategies for identification of the best descriptors. The paper introduces the application as well as presents its usage scenario. To the best of our knowledge no similar solution exists.

1 Introduction

The aim of the paper is to present the *CSV Browser Application* that may be used for cataloguing CSV datasets that may be found on the Web.

The research goal was to develop a solution for efficient search and browsing of various datasets. Efficient in this case means, available through one platform, properly indexed and easy to find. To fulfil this goal, we have developed a Java Web Application, which reads CSV files and produces DBpedia-based annotations (indices) that are used, while browsing resources with the Web application.

The paper is structured as follows. Section 2 presents the *CSV Browser Application* and describes how the tool works. Section 3 provides the usage example. Section 4 presents conclusions and outlines the future work.

2 CSV Browser Application

2.1 Data Sources

The application may catalogue datasets from every data source storing CSV files. In our experiments, we processed datasets that are available at `http://publicdata.eu`.

2.2 How the application works?

The processing of a CSV file, in order to obtain annotations (indices), that are used, while browsing datasets, is performed in the following steps:

1. An initial CSV file (retrieved from a source) is processed using a simple heuristics. The goal is to distinguish between columns storing the data and headers of these columns. We assume that columns' headers should be on the top, the row headers (identifiers) should be included on the left side of the table with the data. Anything, but number, "N/A" or "-" is a field that describes a column or a row.
   The application reads the CSV file and sends headers together with the data from the first 100 rows to the DBpedia Spotlight API[3] to annotate them with matching resources from DBpedia.
2. For each resource retrieved, using SPARQL we retrieve its categories from DBpedia, then categories of these categories, etc. (up to three levels of hierarchy). Using the retrieved data, we create a graph of concepts, e.g. Figure 1 presents a graph retrieved for concepts:
   http://dbpedia.org/resource/Lyon, http://dbpedia.org/resource/Hamburg,
   http://dbpedia.org/resource/Rome, http://dbpedia.org/resource/Milan,
   http://dbpedia.org/resource/Berlin, http://dbpedia.org/resource/Paris.
3. The next step concerns a reduction of a set of categories. The categories bringing little to a description of a dataset such as e.g. `Main_topic_classification` are removed using the blacklist[4]. Then, we check, which from sub-graphs retrieved based on relations between DBpedia categories is the largest (taking into account the number of nodes) and we remove other sub-graphs from further analysis. At this stage, the categories' graph describing the dataset is created.
4. Having a categories' graph, it is possible to apply algorithms known from the graph theory, mostly measuring the nodes centrality, but also PageRank and HITS, in order to define a *leading category* that best describes the content of a CSV file. In the prototype we used the Focused Betweenness Centrality measure. This way the importance (weight) of specific categories (nodes) as descriptors of a CSV resource is quantified.

As a result, each CSV file gets indices enabling search and browsing of available CSV resources using the DBpedia categories. The screenshot of the CSV Browser App is presented in the Figure 2.

The App menu contains three elements:

– Category Search – a field where one may put a DBpedia category in order to find a dataset related to it (auto-completion is enabled),
– Matching Algorithm – which refers to the category matching strategy applied in step 4 of the above mentioned procedure,
– DBpedia Category Tree – enabling browsing datasets using the tree-like menu. It is important to note that the menu is dynamic, as the category structure in DBpedia is a lattice, what means that after choosing a subcategory of a category, one gets new supercategories of a chosen category.

The main part of the screen presents a table with all datasets for a given category. Each dataset is described using its original name and presenting the column, which

[3] http://spotlight.dbpedia.org
[4] http://uimr.deri.ie/sites/StopUris

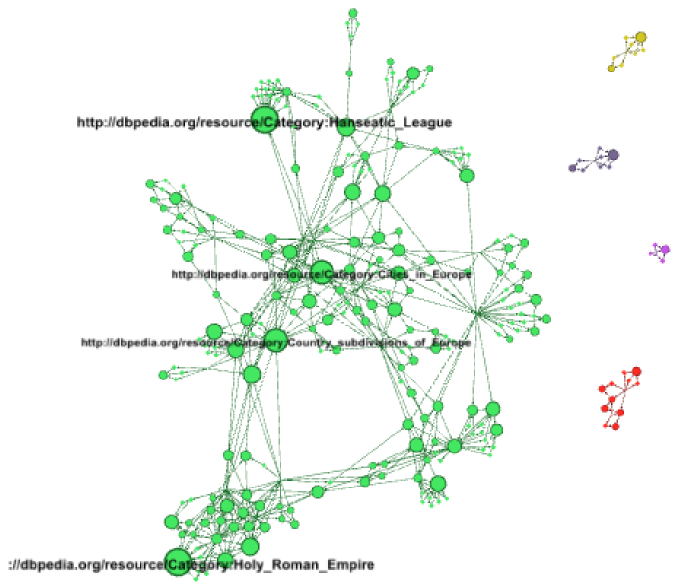
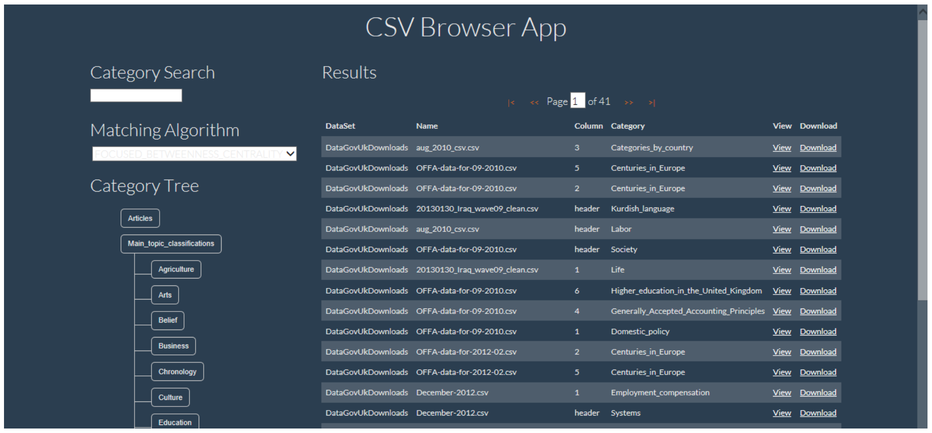

http://dbpedia.org/resource/Category:Hanseatic_League
http://dbpedia.org/resource/Category:Cities_in_Europe
http://dbpedia.org/resource/Category:Country_subdivisions_of_Europe
://dbpedia.org/resource/Category:Holy_Roman_Empire

Fig. 1: A graph of related concepts (after performing step no. 2).
Source: Gephi screenshot



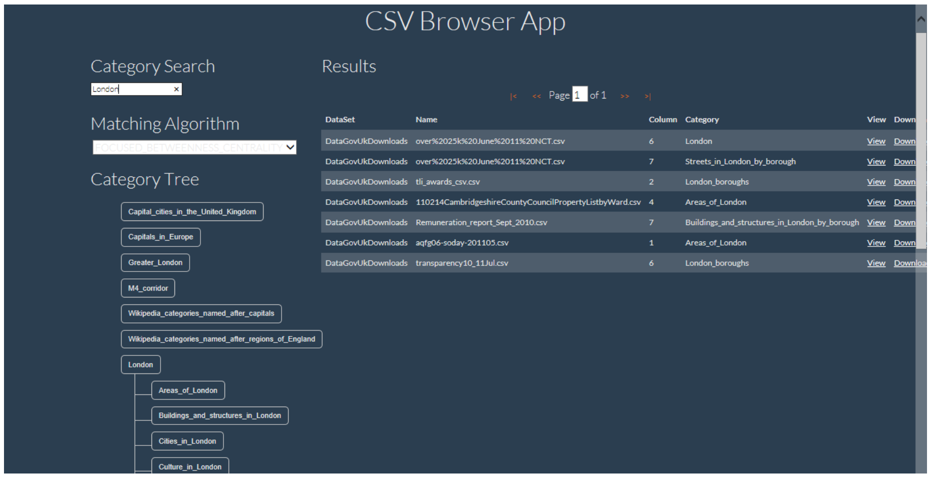
DataSet	Name	Column	Category	View	Download
DataGovUkDownloads	aug_2010_csv.csv	3	Categories_by_country	View	Download
DataGovUkDownloads	OFFA-data-for-09-2010.csv	5	Centuries_in_Europe	View	Download
DataGovUkDownloads	OFFA-data-for-09-2010.csv	2	Centuries_in_Europe	View	Download
DataGovUkDownloads	20130130_Iraq_wave09_clean.csv	header	Kurdish_language	View	Download
DataGovUkDownloads	aug_2010_csv.csv	header	Labor	View	Download
DataGovUkDownloads	OFFA-data-for-09-2010.csv	header	Society	View	Download
DataGovUkDownloads	20130130_Iraq_wave09_clean.csv	1	Life	View	Download
DataGovUkDownloads	OFFA-data-for-09-2010.csv	6	Higher_education_in_the_United_Kingdom	View	Download
DataGovUkDownloads	OFFA-data-for-09-2010.csv	4	Generally_Accepted_Accounting_Principles	View	Download
DataGovUkDownloads	OFFA-data-for-09-2010.csv	1	Domestic_policy	View	Download
DataGovUkDownloads	OFFA-data-for-2012-02.csv	2	Centuries_in_Europe	View	Download
DataGovUkDownloads	OFFA-data-for-2012-02.csv	5	Centuries_in_Europe	View	Download
DataGovUkDownloads	December-2012.csv	1	Employment_compensation	View	Download
DataGovUkDownloads	December-2012.csv	header	Systems	View	Download

Fig. 2: CSV Browser Application.
Source: screenshot

stores data based on which a certain category has been assigned as a dataset index. It is also possible to download a dataset or view it within the application.

3 Usage example

The application usage example is presented in the Figure 3. It presents search results for the category London. It may be noticed that there are 7 different datasets available for this category. The datasets concern such categories as e.g. Areas_of_London or London_boroughs. The tree-like menu also changed – was adjusted to the relations of the London category to other categories in DBpedia.



Category Search: London

Matching Algorithm: FOCUSED_BETWEENNESS_CENTRALITY

DataSet	Name	Column	Category	View	Down
DataGovUkDownloads	over%2025k%20June%2011%20NCT.csv	6	London	View	Down
DataGovUkDownloads	over%2025k%20June%2011%20NCT.csv	7	Streets_in_London_by_borough	View	Down
DataGovUkDownloads	tli_awards_csv.csv	2	London_boroughs	View	Down
DataGovUkDownloads	110214CambridgeshireCountyCouncilPropertyListbyWard.csv	4	Areas_of_London	View	Down
DataGovUkDownloads	Remuneration_report_Sept_2010.csv	7	Buildings_and_structures_in_London_by_borough	View	Down
DataGovUkDownloads	aqfg06-soday-201105.csv	1	Areas_of_London	View	Down
DataGovUkDownloads	transparency10_11Jul.csv	6	London_boroughs	View	Downloa

Fig. 3: CSV Browser Application Usage Example.
Source: screenshot

4 Conclusions and Future Work

The paper presents demonstration of the application enabling indexing of CSV datasets, that makes searching for these datasets as well as browsing them more efficient. Efficient in this case means, available through one platform, properly indexed and easy to find. Each CSV file is indexed not taking into account a description of the dataset, but its content. We have also developed a version of this application to process the Open Data Support datasets. In this case however, we processed only titles and descriptions of the datasets.

Next steps concern assessing an accuracy of the indices created, integration with CKAN as well as putting the indexing component in the pipeline of CSV to RDF translation.

Acknowledgement This work was supported by a grant from the EU 7th Framework Programme for the project LOD2 Creating Knowledge out of Interlinked Data (GA no. 288176).