DBpedia Atlas: Mapping the Uncharted Lands of Linked Data

Fabio Valsecchi
Institute of Informatics and Telematics, CNR Pisa
fabio.valsecchi@iit.cnr.it

Matteo Abrate
Institute of Informatics and Telematics, CNR Pisa
matteo.abrate@iit.cnr.it

Clara Bacciu
Institute of Informatics and Telematics, CNR Pisa
clara.bacciu@iit.cnr.it

Maurizio Tesconi
Institute of Informatics and Telematics, CNR Pisa
maurizio.tesconi@iit.cnr.it

Andrea Marchetti
Institute of Informatics and Telematics, CNR Pisa
andrea.marchetti@iit.cnr.it

ABSTRACT

In the last few years, Linked Open Data sources have extremely increased in number. Despite their enormous potential, it is really hard to find effective and efficient ways for navigating and exploring them, mainly because of complexity and volume issues. In fact, application developers, students and researchers that are not experts in Semantic Web technologies often lose themselves in the intricacies of the Web of Data. We propose to address this problem by providing users with a map-like visualization that acts as an *entry point* for the exploration of a dataset. To this end, we adapt a spatialization approach, based on cartographic and information visualisation techniques, to make it suitable for Linked Data sets with a hierarchical ontological structure. Finally, we apply our method on DBpedia, implementing and testing a prototype web application that shows a comprehensive and organic representation of the more than 4 million instances defined by the dataset.

Categories and Subject Descriptors

H.5.0 [**Information Interfaces and Presentation**]: General

Keywords

Linked Data, Information Visualisation, Cartography

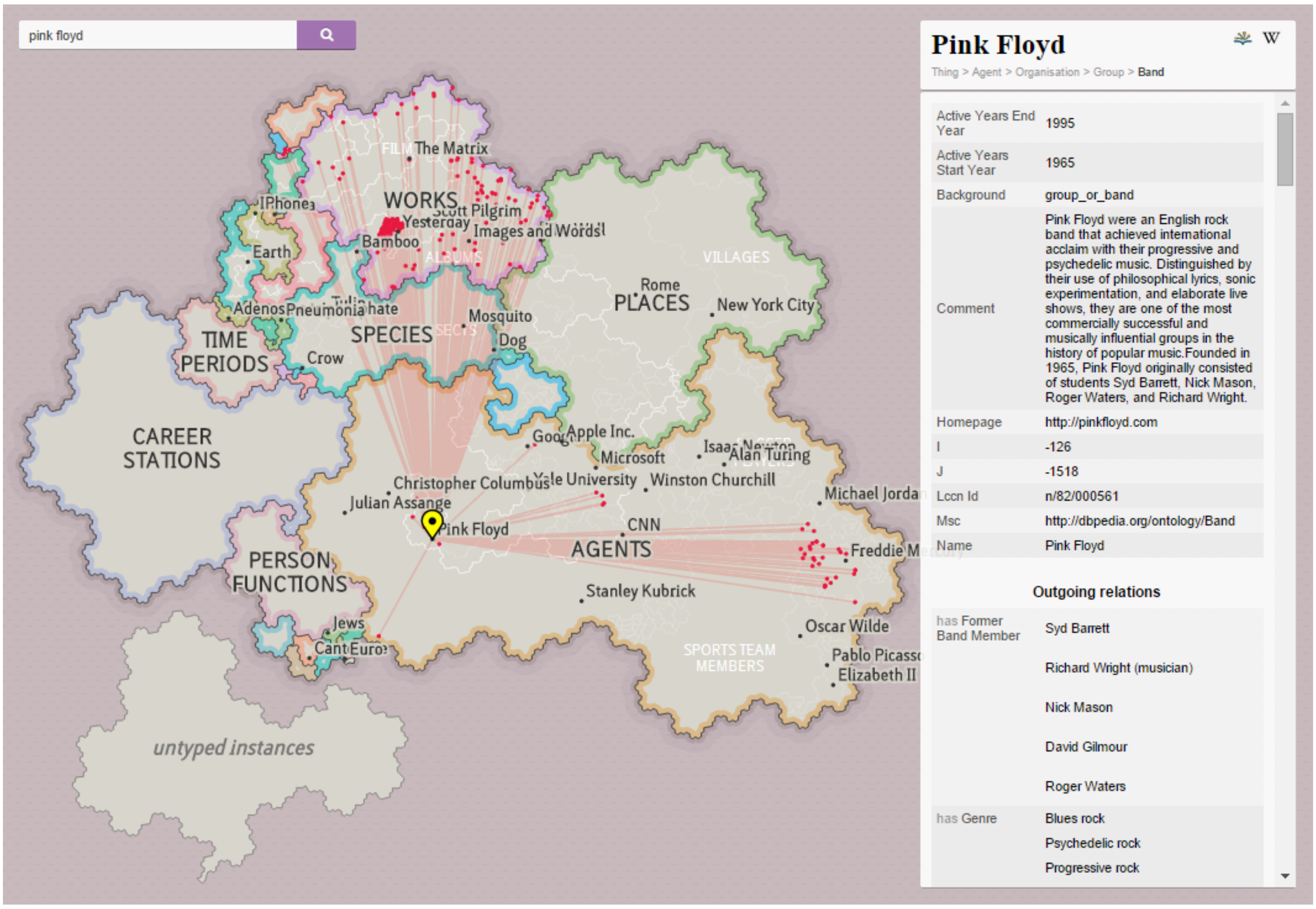
1. INTRODUCTION

During the last few years, the amount of available datasets based on the Linked Open Data (LOD) paradigm has extremely increased[1]. However, virtually no one outside the Semantic Web community is able to completely understand Linked Data and put its full potential at use. Other categories of users surely have interest in LOD sets, but, lacking a deep expertise, they may find it difficult to make sense of their content or structure [6]. In our opinion, such non-expert users (e.g., application developers, students, researchers in other fields) often have the need to look at a dataset and *see* the whole picture, getting an answer to the somewhat naive question "What is the dataset like?". More specifically, they can benefit from having a feel of how big it is in terms of instances, relationships and properties, what kind of entities it contains, how they are organized, how they are connected to each other, and so on. Answering those questions can prove to be fundamental in promoting knowledge about these datasets, fostering their growth and driving their adoption for a variety of applications. Information visualization techniques have already been proposed to address similar needs [4], because of their effective exploitation of the innate human ability of acquiring information through vision. Nevertheless, to the best of our knowledge, the existing works are either focused on the exploration of small groups of entities or on the presentation of aggregated data. What is currently missing is an *entry point*, something that could lead a user from an overview of the main features of a dataset to its tiniest details.

We propose to use a map-like interactive visualization to serve as such an entry point. If designed by taking cartographic principles into account, a map can leverage both innate visual perception abilities and learned map-reading skills to attain a high level of efficacy in communicating features of large scale, complex structures [15, 1]. A zoomable map also nicely embodies Ben Shneiderman's well-known Visual Information-Seeking Mantra (*"Overview first, zoom and filter, then details-on-demand"*) [14, 6], according to which the overview should always come first in a visualization, since it provides the general context of a dataset, and only in a second moment users should be able to load more detailed information. To obtain such a map, a process of *spatialization* (i.e., the assignment of position and shape to abstract, non-geometrical data) becomes necessary. We propose an adaptation of the work by Auber et al. on Gosper treemaps [2] to the case of LOD sets with a hierarchical ontological structure. The approach enables the automatic generation of stable 2D maps that show the entirety of the entities contained in the dataset, forming a hierarchy of regions according to their ontological class. Such maps can

[1] Statistics are available at `http://lod-cloud.net`.

WWW2015 Workshop: Linked Data on the Web (LDOW2015).



Pink Floyd

Thing > Agent > Organisation > Group > **Band**

Active Years End Year	1995
Active Years Start Year	1965
Background	group_or_band
Comment	Pink Floyd were an English rock band that achieved international acclaim with their progressive and psychedelic music. Distinguished by their use of philosophical lyrics, sonic experimentation, and elaborate live shows, they are one of the most commercially successful and musically influential groups in the history of popular music.Founded in 1965, Pink Floyd originally consisted of students Syd Barrett, Nick Mason, Roger Waters, and Richard Wright.
Homepage	http://pinkfloyd.com
I	-126
J	-1518
Lccn Id	n/82/000561
Msc	http://dbpedia.org/ontology/Band
Name	Pink Floyd

Outgoing relations

has Former Band Member	Syd Barrett
	Richard Wright (musician)
	Nick Mason
	David Gilmour
	Roger Waters
has Genre	Blues rock
	Psychedelic rock
	Progressive rock

Figure 1: A screenshot of DBpedia Atlas, available online at `http://wafi.iit.cnr.it/lod/dbpedia/atlas`. The code is open source and hosted on GitHub (`https://github.com/fabiovalse/dbpedia_atlas`). The search box on the top left allows users to search for a specific instance. On the map, a yellow placemark identifies the selected instance, while the red links show the locations of the resources related to it. The infobox on the right reports the information about the selected instance such as its label, classes, data properties, incoming and outgoing relations.

then be used as a foundational layer for the creation of a collection of *thematic maps* and ancillary charts, forming an *atlas* describing many different aspects of the dataset. Our method is applied to the English version of the DBpedia knowledge base [3], obtaining a comprehensive interactive visualization of the more than 4 million instances defined by its RDF triples, as well as additional representations of different aspects of the dataset. Users involved in preliminary tests of the resulting prototype were able to get insights about some non-obvious and not-so-known features of DBpedia, proving the usefulness of the approach not only as a presentation tool, but also as a visual exploration system.

1.1 Related Work

The need to visualize LOD is an important issue in the Semantic Web community. In fact, several works have already tackled the problem. LodLive [5] is an RDF browser that allows to explore LOD by manually creating a node-link diagram. Starting from a given URI, the user can expand the diagram by following links to other resources. RelFinder [8] addresses the task of revealing if and how two given resources are connected, by visually showing all the paths between them. gFacet [9] allows the navigation of a LOD set combining graph-based visualization with faceted filtering techniques. All the aforementioned applications make use of a node-link representation that allows to clearly identify the relations between resources, but fails to scale to large amounts of data. Among other solutions, DBpedia viewer [12] is a web application for searching resources and consulting the available information as text, images, geographical maps and raw data. LodView[2] is a tool for navigating LOD sources through a user-friendly interface based on a single-instance view. Spacetime [16] allows to implicitly perform SPARQL queries over spatio-temporal data and visualize their result on a geographical map connected to a timeline. Linked Data Query Wizard [10] is an analysis tool for searching resources, filtering them, refining and visualizing the output in the form of different diagrams. All the works mentioned above provide useful techniques for navi-

[2] http://lodview.it/

	Scale			Visualization Technique
	Whole Dataset	Subset	Single Instance	
LodLive		✓	✓	node-link, infobox
RelFinder		✓	✓	node-link, infobox
gFacet		✓		list, node-link
DBpedia Viewer			✓	infobox
LodView			✓	infobox
Spacetime		✓	✓	geomap, timeline, infobox
Linked Data Query Wizard		✓		table, node-link, various
LOD Visualization	✓	✓		treemap, tree
DBpedia Atlas	✓	✓	✓	*map-like visualization, infobox*

Table 1: This table shows a comparison of our proposal with eight applications found in literature. Most of the applications represent a subset of a given Linked Data set and give a view of single instances. Only *LOD Visualization* provides a visualization of the whole dataset but it does not represent single instances.

gating LOD. However, they are focused on the exploration of single entities or a small group of them, neglecting to show an effective overview of the whole data source. This aspect is one of the key points of Shneiderman's Mantra. Other works present some kind of overview: LODVisualization[3] is a prototype based on the Linked Data Visualization Model [4], and offers different diagrams such as an interactive treemap and an indented tree representing class hierarchies. The former shows a compact overview of a data set, but it does not provide the detailed information about the resources within it. In the latter, the ontology is clearly visualized but no overview is shown, since the number of classes makes the diagram too long to be displayed in a single view.

2. DESIGN

DBpedia Atlas is designed as an interactive, web-based visualization that allows different kinds of users to understand and benefit from a complex RDF dataset such as DBpedia. The application is primarily meant for those users who are not proficient in semantic web technologies but are interested in learning, researching, or developing applications specifically on DBpedia. To a lesser extent, casual users interested in doing some research about a given subject could benefit from the map as a complementary way of accessing Wikipedia content.

Our primary goal is to provide these users an overview. Hence, we first define some high-level tasks that they should be able to perform by looking at the visualization at a glance: i) get a feel of the size of the dataset; ii) see the main aspects of its structure; iii) approximately compare different parts of its structure in terms of both size and complexity. Secondly, we define more specific tasks, to characterize the user's wish to get detailed information by interacting with the visualization space: i) locate a class; ii) search for or locate an instance; iii) consult its properties; iv) browse the list of its connections; v) explore to find the location of its related instances; vi) discover which are the classes to which it is more connected; vii) compare its connections with the ones of other instances.

2.1 Data abstraction

Since hierarchical ontologies are often the structure upon which Linked Data sets are based [6], we consider the set of RDF triples of DBpedia to form a *compound network*, i.e., a structure defined by a graph with an associated tree. In our

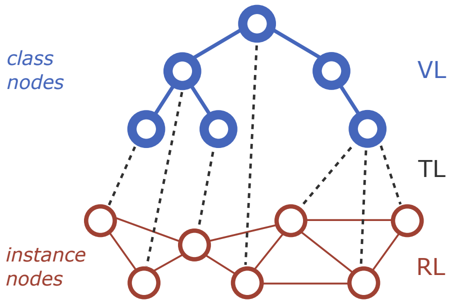


Figure 2: A graph and an associated tree define a compound network. In our case, it is composed by class nodes, instance nodes, vocabulary links (VLs), relationship links (RLs) and type links (TLs).

case (Figure 2), it comprises two kind of nodes: *class nodes*, which define the hierarchical structure, and *instance nodes*, which are the nodes of the graph. More precisely, we define an instance node for each distinct URI found as subject or object of an RDF triple. In order to avoid to take external resources into account, we filter out URIs not prefixed by `http://dbpedia.org/resource/`. Three kinds of links are also defined [7]: *vocabulary links* (VL) are derived from the DBpedia infobox ontology (i.e., `rdfs:subClassOf`), *relationships links* (RL) express various types of connections between two instances (e.g., `dbpedia-owl:birthPlace` for *Galileo Galilei* and *Pisa*), and *type links* (TL) connect class nodes to instance nodes, describing the membership of an instance to a class (i.e., `rdf:type`). Of the many TLs that a single instance could feature (e.g. *Scientist*, *Person*, *Agent* and *Thing* for *Galileo Galilei*), we consider only the one leading to the most specific class in the ontology (e.g. *Scientist* for *Galileo Galilei*), since the other ones can be inferred by walking up the ontology tree. We ran an ad-hoc

[3] http://lodvisualization.appspot.com/

script that verified that no instance node is connected to multiple class nodes belonging to different branches (i.e, no entity has incompatible classes). In the resulting compound network, 476 class nodes constitute the tree, while 4,232,628 instance nodes and 15,077,186 RLs compose the graph. We do not consider all the 721 class nodes currently included in the DBpedia ontology tree[4] because we prune the tree branches to which no instances are connected. Since the automatic attribution of a class to a DBpedia entity from the corresponding Wikipedia infobox may lead to errors [13], our compound network is characterized by large amounts of instance nodes connected to very generic class nodes (e.g., *Leonardo da Vinci* is classified simply as *Person*, while it could have been more specifically typed as *Artist* or *Scientist*). It is also worth noticing that about 500,000 instance nodes in our network have no associated class node. Such entities may have a URI but still lack their own Wikipedia page (i.e., the “red links” appearing in Wikipedia articles), or be the result of an error of the aforementioned automatic classification.

2.2 Interactive Visualization

The spatialization process upon which our visualization is based adopts a treemap approach [11], following the results of Auber et al. on Gosper treemaps [2]. Treemaps are in general able to represent big and complex trees in a small amount of space, trading the explicit representation of hierarchical links for compactness. Gosper treemaps have the additional feature of being able to represent each leaf of the tree as a hexagonal tile with a specific position, at the expense of some compactness and simplicity. In both cases, internal nodes of the tree are implicitly represented as a hierarchy of regions contained into one another.

Gosper treemaps come with the additional benefit of producing geographic-like regions, which helps users to instinctively read the visualization as they would with a geographic map. Thus, in our approach, each instance node (i.e., each entity from DBpedia) is given a position into a hexagonal tiling. Entities belonging to the same class are placed near one another, and positioned in the same region. Unfortunately, though, two entities that are neighbors in the tiling do not necessarily belong to the same class. By construction, the size of a region corresponding to a class node is proportional to the amount of instance nodes having that class or a subclass of it (e.g., *Person* takes *Galileo Galilei* into account, even if its most specific type is *Scientist*).

The layout algorithm of Gosper treemaps is also order-preserving and stable, i.e., a small modification of the dataset would cause only a small change in the map[5], making it ideal for an ever-changing Linked Data set like DBpedia. It would in fact be confusing for users to explore a newer map of the same dataset and see a very different spatial arrangement.

The interface of the application (Figure 1) comprises three main components that work together in order to provide overview, zoom and filter and details on demand.

1. *Map.* It initially provides the overview of all the instances and classes in DBpedia, allowing the user to zoom and pan at will. The main island represents *owl:Thing* (i.e., the root of the ontology) while the colored regions identified by the uppercase labels represent its direct children (e.g., *Agent*, *Place*, *Work*, *Species* and so on). Instances with missing types are shown in the smaller island at the bottom left. Regions having an area of suitable size show a label from the beginning, while labels of minor regions are loaded when zooming in. The zoom behaviour allows to filter out certain regions and to focus the attention to other ones. Some notable instances have been manually identified and have been given a label that is always visible, in order to provide the users with additional, city-like landmarks to get orientation in the map and to identify some basic categories. Selecting an instance on the map loads its details in the infobox (on the right). All the instances connected to it are also depicted in the map as a distribution of red dots. Two thematic maps can also be loaded: one showing the depth of the classes in the DBpedia ontology hierarchy, and the other showing the average outdegree of instances contained in each class (Figure 6).

2. *Search box.* This component (top left of the interface), allows to perform a text search about a specific instance by using the DBpedia lookup service [3]. The selection of one of the resulting instances triggers the displaying of its position and distribution of connected entities on the map, and the loading of its details in the infobox;

3. *Infobox.* Shows the title, classes, data properties, incoming and outgoing relations of an instance. Links to DBpedia online and Wikipedia are also provided. Data is loaded within this container when the user selects an instance from the map or from the search box. Moreover, by clicking on an outgoing or incoming property, it is possible to follow the connection to another instance.

3. PRELIMINARY EVALUATION

To asses the usefulness of our approach and get an early feedback, we carried out a preliminary formative evaluation of our prototype. We briefly presented the purpose of DBpedia Atlas to five users with different backgrounds: three technical users without a specific expertise on Semantic Web technologies, and two lay users with no scientific or technical background. Then, we observed their free interaction with the system, and asked them to answer some questions to assess their ability to perform the tasks introduced in Section 2. Finally, we asked them to compare the application with other solutions and to complete a short questionnaire.

Participants found DBpedia Atlas easy to read and to operate with, giving it an average score of 4 in a scale from 0 to 5. They also found it useful (3.6/5 on average), especially to get a general feel of the dataset. Two of them were skeptical about the level of detail of the map, expressing the need to see more information as they progressed with the zoom. All of them reported to prefer DBpedia Atlas over LodLive [5] and RelFinder [8] as an entry point for the exploration of the dataset, but RelFinder was pointed out to be more useful for a specific task unsupported by our map (i.e., to find paths between two instances).

[4] http://mappings.dbpedia.org/server/ontology/classes/

[5] This is true only when both the original and the modified tree are ordered by following the same criterion. In order to ensure this and be able to keep a similar map for future updates of the dataset, we transform the tree from our compound network into its *canonical ordering* form [17].

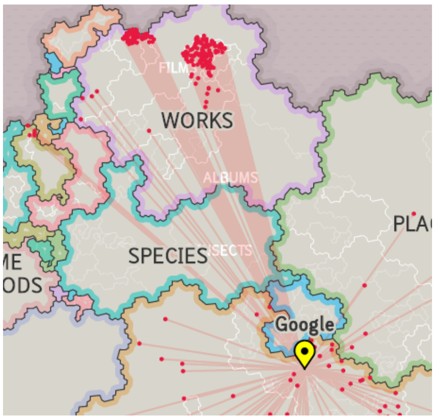


Figure 3: The distribution of instances (red dots) connected to the entity *Google* (yellow placemark). A large number of dots gathers in the *Website* region (top left) and in the *Software* region (top middle).

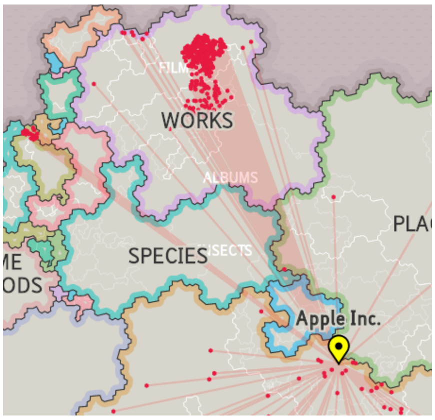


Figure 4: When *Apple Inc.* is selected, the amount of websites decreases significantly, while *Software* becomes much more prominent. This is especially true for the lowest part of the region (*Video Game*). An interesting conglomerate appears on the left (*Device*).

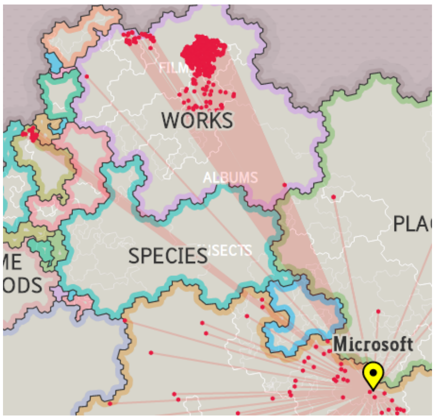


Figure 5: The distribution for the instance *Microsoft* is more similar to the one for *Apple Inc.* than it is for *Google*. However, with regards to both *Device* and *Website*, it seems that Microsoft falls somewhere in between the other two.

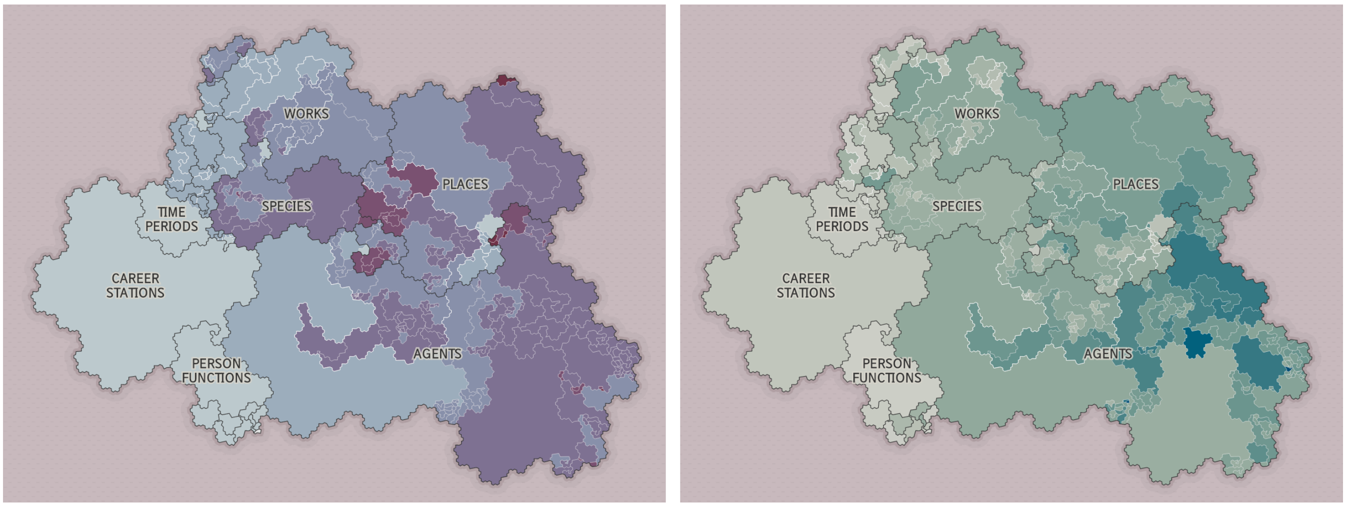


Figure 6: Two examples of thematic maps. The first one shows the depth of the classes in the DBpedia ontology hierarchy (the darker, the deeper). The second one shows the average outdegree of instances contained in each class (the darker the color, the higher the average outdegree). By inspecting the interactive maps, it can be seen that the deepest level of the ontology corresponds to the small *Diocese* class (top right), and that the highest average outdegree is found in *Soccer Manager*, *Jockey* and *Horse Trainer* (bottom right). Conversely, *CareerStation*, *PersonFunction* and *TimePeriod*, while vast, have the lowest depth and the lowest average outdegree.

When asked to estimate the amount of instances in the map, almost all the participants replied with a number greater than a few millions, proving to get a feel of the vastness of the dataset. All the participants showed no difficulties in interpreting the regions as more and more refined classifications of the entities composing the map, nor in relating the size of regions to the amount of instances of that class. The largest classes of the ontology (e.g., *Agent*, *Place*, *Work* and *Species*) were quickly identified from the initial overview, while minor ones were inspected by zooming in. In one case, a user reported to give more importance to detailed regions (i.e., with many subdivisions) rather than to big ones. Three participants got curious about the big and flat *CareerStation* class, and tried to understand its meaning by selecting random entities from the region (discovering that it contains information about the career of people, mostly athletes).

Users selected various instances and compared their dot distributions of connected entities, sometimes noting a steep difference in the amount of connections. Some interesting patterns were also found, as in the case of the comparison between *Google*, *Apple Inc.* and *Microsoft* (see Figures 3, 4 and 5 for more details). Uncommon connections sometimes popped to the eye of participants when a selection showed a dot in an unexpected region. For example, when one of them selected the instance *Dog* from the *Species* class, he noticed a lone connection in the *Food* region, revealing that *Saksang* is an Indonesian dish made of dog and pork. Thematic maps (Figure 6) got mixed reactions from users, which described them as very informative but harder to read than the base map, especially because of difficulties in the interpretation of label-region correspondence.

4. CONCLUSIONS AND FUTURE WORK

We presented DBpedia Atlas, a web application for exploring instances, relations and classes of DBpedia. By using this application, users can obtain a grasp of the fundamental properties of the dataset, browse it, and get several interesting insights, without the need to be experts of Semantic Web technologies. The underlying approach we propose, based on cartography and information visualisation techniques, can be reused for visualizing and exploring other LOD sets with hierarchical ontologies. Several improvements can be introduced to the current prototype. Data can be updated to reflect the current status of DBpedia online[6]. A formal user study with a greater number of participants can be carried out to better validate the approach and to get more feedback. Specific improvements can be made to the map visualization, in order to increase its expressive power. In particular, a ranking factor (based for example on the degree of an instance node, or on the length or the popularity of the corresponding Wikipedia article) could be adopted to display the most important instances (i.e., "cities") at each zoom level. Moreover, a concept of *distance* between instances can be introduced to complement the treemap approach. We are currently investigating an ontology-independent similarity measure that would pack similar entities together regardless of their class. This approach could prove to be useful to define a meaningful spatialization for vast regions of entities having the same class or no class at all, and it would open our approach to datasets without a hierarchical ontology.

5. REFERENCES

[1] M. Abrate. *Data Cartography: atlases and maps for non-geographical data.* PhD thesis, 2014.

[2] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sallaberry, and A. Saulnier. Gospermap: Using a gosper curve for laying out hierarchical data. *IEEE Trans. on Visualization and Computer Graphics*, 2013.

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 2009.

[4] J. M. Brunetti, S. Auer, and R. García. The linked data visualization model. In *International Semantic Web Conference*, 2012.

[5] D. V. Camarda, S. Mazzini, and A. Antonuccio. Lodlive, exploring the web of data. In *Proc. of the International Conference on Semantic Systems*, 2012.

[6] A.-S. Dadzie and M. Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2011.

[7] T. Heath and C. Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 2011.

[8] P. Heim, S. Hellmann, J. Lehmann, S. Lohmann, and T. Stegemann. Relfinder: Revealing relationships in rdf knowledge bases. In *Semantic Multimedia.* 2009.

[9] P. Heim, J. Ziegler, and S. Lohmann. gfacet: A browser for the web of data. In *Proc. of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web*, 2008.

[10] P. Hoefler, M. Granitzer, E. Veas, and C. Seifert. Linked data query wizard: A novel interface for accessing sparql endpoints. In *Proc. of Linked Data on the Web at WWW*, 2014.

[11] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *IEEE Proc. of Conference on Visualization*, 1991.

[12] D. Lukovnikov, C. Stadler, D. Kontokostas, S. Hellmann, and J. Lehmann. Dbpedia viewer-an integrative interface for dbpedia leveraging the dbpedia service eco system. In *Proc. of Linked Data on the Web at WWW*, 2014.

[13] H. Paulheim and C. Bizer. Type inference on noisy rdf data. In *Internatioan Semantic Web Conference.* 2013.

[14] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, 1996.

[15] A. Skupin. From metaphor to method: Cartographic perspectives on information visualization. In *IEEE Symposium on Information Visualization*, 2000.

[16] F. Valsecchi and M. Ronchetti. Spacetime: a two dimensions search and visualisation engine based on linked data. In *The Eighth International Conference on Advances in Semantic Processing*, 2014.

[17] R. A. Wright, B. Richmond, A. Odlyzko, and B. D. McKay. Constant time generation of free trees. *SIAM Journal on Computing*, 1986.

[6] Our work is based on the latest available DBpedia dump (2014). Subsequent updates are not included in our map.