ISSN 1346-5597

National Institute of Informatics

NII Technical Report

ArtViz: A Web Platform for Artist Data Visualization and Exploration.

Oana Balaceanu, Hideaki Takeda

NII-2016-002E
Feb. 2016

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

ArtViz: A Web Platform for Artist Data Visualization and Exploration.

Oana Balaceanu
Vienna University of Technology
Vienna, Austria

Hideaki Takeda
National Institute of Informatics
Tokyo, Japan

***Abstract*—Browsing through the available data nowadays can be a difficult task for the unexperienced users. Every day, the Internet is flooded by unstructured information that sometimes needs time to be extracted and analyzed. When it comes to art data, there aren't any concrete sources that offer consistent information about artists in an interactive way. In this paper we describe the prototype of ArtViz, a web platform that displays art linked data to unexperienced users and helps them discover new information through interactive visualization. Furthermore we will present additional useful features like natural language querying or recommending similar entities (artists).**

***Keywords*—Linked Open Data, DBPedia, Europeana, ULAN, Natural Language Processing, Machine Learning, Data Visualization**

I. INTRODUCTION

Similar to the situation of data in general, art data also faces an exponential growth that makes its retrieval and analysis very difficult. The Internet is filled everyday with more and more unstructured data that is hard to process and use in a fast, useful and automatic way.

When unstructured data is all that is available within a specific field or subfield, technologies like natural language processing has to be used in order to structure it. On the other side, small but consistent steps are made in the direction of structuring the available data. Semantic technologies like OWL, RDF, SPARQL or Linked Open Data sources like DBPedia, Europeana or Getty are making way for the structured data to be usefully used in applications. Unfortunately, all the previously mentioned sources offer a rather large quantity of information, but would be more consistent if used together. In other words, there are lacks in ontology alignments [1].

Furthermore, art data offers potential and good possibilities when it comes to data visualization and offering the users an interesting experience. Everything can be presented in encyclopedia-like style with attractive diagrams and concrete information.

In this paper we will describe how ArtViz managed to use the potential of structured art data and present it in an attractive way for any user, including the unexperienced ones.

II. MOTIVATION AND VISION

The primary goal of the ArtViz prototype is to offer users an easy way to browse art-related information. Along with this comes the need to use a consistent data source. On a simple search through the linked open data sources we can notice that they offer large amounts of data, but none complete. This means that the need to aggregate the multiple sources will appear when preparing or retrieving the datasets. In the present case, we took into consideration 4 endpoints: DBPedia as the main data endpoint, Getty (with the ULAN vocabulary), Wikidata and Europeana. The second goal was finding a way to outline the useful data through easy to use visualization means. This would attract the users and offer them a fast way of answering their own questions. At the present moment, the most popular way of displaying linked open data is under a graph form [2] or as a friendlier text-based interface [3]. As a result, another goal is experimenting with other form of visualization and diagrams. The last goal is enhancing the experience by offering functionalities like a question answering system and a recommender system. All the previously mentioned goals add up in the vision of the project which is an encyclopedia-like platform that accesses a considerable dataset and returns answers in an attractive form.

III. RELATED WORK

When it comes to the museum and library archive field, there are several applications that have used LOD sources datasets, but haven't made visualization a priority task.

One of the most stable and concrete application that deals with both LOD and visualization is LODView,[1] a Java web application implemented by the LODLive team and launched in December 2014. LODView comes with additional functionalities to LODLive[2] which is a graph-like RDF browser. LODView offers IRI dereferenciation that complies with the

[1] http://lodview.it/

[2] http://en.lodlive.it/

W3C standards and allows the users the publish RDF data in an easy and flexible way. This way the main goal of the project is fulfilled: helping semantic web spread. Among the main features of LODView we can mention multiple language compact view, a widget area that contains multimedia elements like images and maps and other data published in the LOD cloud regarding the same topic. Compared to ArtViz, LODView also uses data from multiple sources (DBPedia and the Smithsonian) but doesn't put very much accent on its visualization. As mentioned before, this is done by a separate application and only under a graph form.

Another application, eCultureMap[3] is sustained by Europeana's partners in order to gather cultural content in one single geographical knowledge map. Basically, it is a very efficient re-use of the Europeana data published under the Creative Commons Zero license. Launched in 2008 through the initiative of the European Union, Europeana contains now more than 30 million media objects (images, texts, audio recordings, etc.) from a large range of European museums, libraries and collections. This large amount of data allowed the map to grow to a total number of 2 million digital cultural heritage objects. Locations are a very important piece of information which can enhance the searching and visualization process of data. Besides offering this, eCultureMap also shows cultural heritage objects in an international context (through linking to the Europeana portal) and national context (through linking to the national portals). When compared to ArtViz, eCultureMap shows focus only upon a type of visualization (map view). The data is provided by Europeana and its partners.

LODStories[4] on the other hand is a web application that offers the users the possibility of exploring art linked data in a story-like manner. Basically users start with a topic of their choice and create a path formed by subjects and their connecting properties. After the path is formed, the user can create videos using the Smithsonian database, Google and YouTube API. The generated videos can then be shared with other users on the platform and the social media. The goal of LODStories is to create materials that can be used in a variety of educational settings [4].

One final mention is represented by the official Rijksmuseum[5] site. When browsing through their art collection, users can use a tag-based search which allows them to select an art object and then choosing the object's tags as keywords another set of related objects is populated. A category-like browsing system is also available. Similar to the Rijksmuseum site there is the Cooper Hewitt[6] art collection in which users can browse through art objects by color and category.

IV. ARTVIZ PLATFORM

The ArtViz platform consists of six static pages each with a specific type of diagram, dataset and art relationship:

- Similarity Diagram
- Dependency Wheel
- Faceted Browser
- Timeline
- Concept Map
- Artist Bar chart

Each of the pages contains a search box where users can ask artist related questions in a natural language manner, a diagram area, user interface controls and a result area that is populated according to the selected item (either the response to the search question or a label clicked on the diagram). The navigation between the pages is done through the menu available on the top side, under the header. All the diagram types and additional functionalities will be explained in the following subchapters.

A. Similarity Diagram

The similarity diagram was implemented in order to show the results of the recommender system and an overview on the similarity degree between top artists. The results of this system are also used more briefly in other types of visualization like the Dependency Wheel, but in the Similarity Diagram we can see how the system works. When entering the diagram's page, a user sees the generated diagram and a set of checkbox controllers. When the mouse is hovering over the diagram, the diagram's chord that belongs to the hovered artist will be highlighted and the similarity scores between the artists and the rest of the artist dataset will appear. The best match will appear when hovering on the artist and the rest of the scores will appear when hovering over each chord. (Fig. 1) The user can control which combination of similarity type he wants to visualize by checking the correct checkboxes (location, movement, field or year) (Fig. 2). Also, the number of artists that are displayed can be changed (Fig. 3). The artist data is fetched straight from DBPedia through SPARQL as JSON format, parsed and processed further in order to calculate the similarity scores using the cosine distance and a custom formula. The formulas involved are the following:

$$sim(A,B) = 2 * yearSim(A,B) + 2 * locationSim(A,B) + 3 * fieldSim(A,B) + 4 * movementSim(A,B) \quad (1)$$

The year and the location functions uses the cosine distance and have the following form:

$$movSim(A,B) = \frac{|A_{mov}| + |B_{mov}|}{|\cup mov|} \quad (2)$$

$$yearSim(A,B) = \frac{birth(A)*birth(B)+death(A)*death(B)}{\sqrt{birth(A)^2+death(A)^2}+\sqrt{birth(B)^2+death(B)^2}} \quad (3)$$

where:

- birth(A) - birth year of artist A
- birth(B) - birth year of artist B
- death(A) - death year of artist A
- death(B) - death year of artist B

$$locationSim(A,B) = \frac{lat(A)*lat(B)+long(A)*long(B)}{\sqrt{lat(A)^2+long(A)^2}+\sqrt{lat(B)^2+long(B)^2}} \quad (4)$$

where:

- lat(A) – latitude of birth location for artist A
- long(A) – longitude of birth location for artist A

[3] http://eculturemap.eculturelab.eu/eCulture14m/map.html
[4] http://lodstories.isi.edu/LODStories/html/linkeddataeduapp.html
[5] https://www.rijksmuseum.nl/en/explore-the-collection
[6] https://collection.cooperhewitt.org/

- lat(B) – latitude of birth location for artist B
- long(B) – longitude of birth location for artist B

The movement and field similarities are calculated by dividing the number of common movements/fields between two artists and the total number of movements/fields.

$$movementSim(A,B) = \frac{movements(A) \cap movements(B)}{allMovements} \quad (5)$$

where:

- movements(A) – all distinct movements under which artist A has worked
- movements(B) – al distinct movements under which artist B has worked
- allMovements – all distinct movements presents at all the artists in the dataset

The weights for each type of similarity were chosen according to the potential importance of each criteria when deciding what makes two artist more alike. As a result, two artists might be more similar if they created under the same movement than if they were born in the same country.

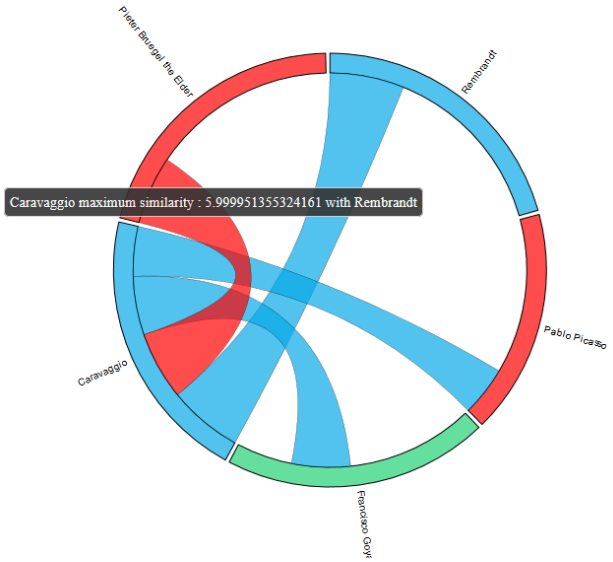

Fig. 1. Similarity Diagram best match. Once the mouse is hovering over Caravaggio, we can see that the most similar artist with him is Rembrandt.

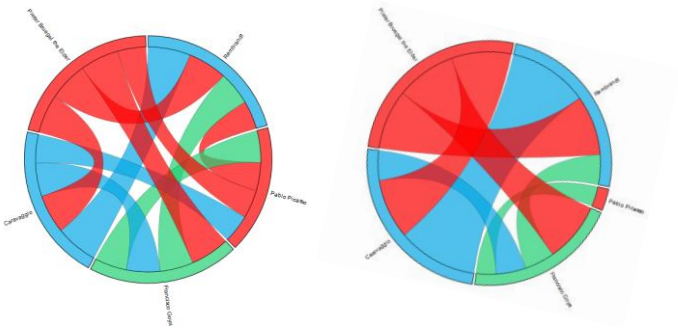

Fig. 2. Similarity Diagram difference among the selected criteria: first diagram all criteria selected (year, location, movement, field), second diagram movement and field criteria selected.

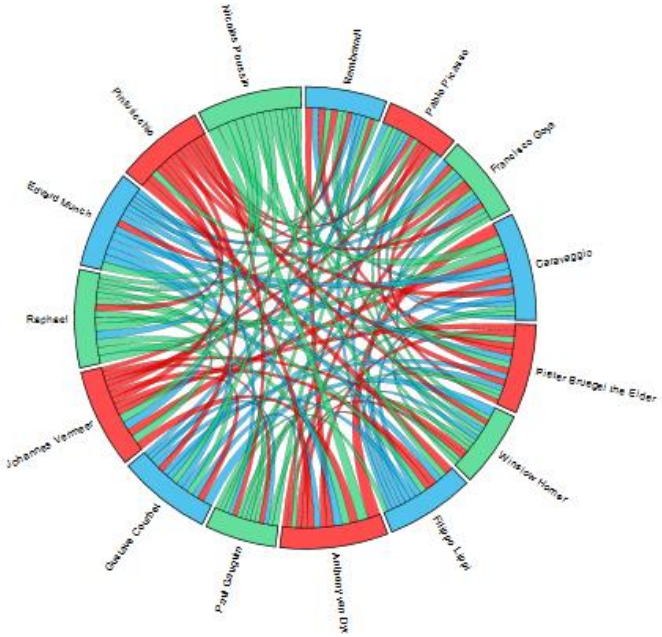

Fig. 3 Similarity Diagram for top 15 artists with all the criteria selected (year, location, movement, field)

B. Dependency Wheel

The dependency wheel diagram gathers multiple functionalities that show the relationships that artists have according to different criteria (geographic, movement and relationship with other artists). All the datasets start from the same sets of artists and then their data is enriched with additional information (the artist's movement, location and other artists that influenced/were influenced). This explains the discrepancy between the number of entries in the movement and geography datasets and the influenced-influences dataset (some artists might have influenced other 10 artists). The data is fetched as JSON and structured in the following way. (Fig. 4)

```
[{ "name": "source",
  "influence": ["target#1", "target#2"] }, … ]
```

Another important functionality of this visualization form is showing additional information once an artist is selected. If the label selected is not an artist or there is no information available, a message will be displayed ("No information found"). Last but not least, another important functionality on this type of diagram is the final form of the recommender system. When an artist label is selected, the same algorithm and formulas used in the Similarity Diagram will calculate the most similar artist for the Dependency Wheel. (Fig. 5)

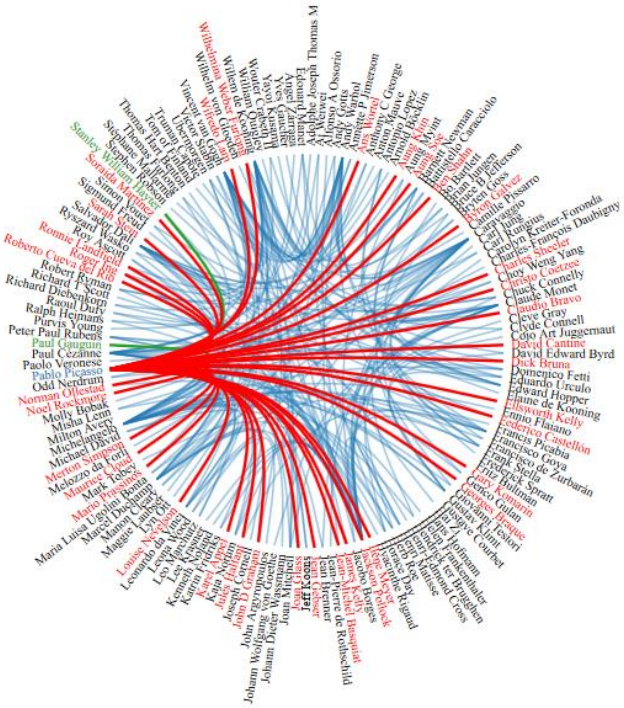

Fig. 4. Dependency Wheel – Influenced criteria for Pablo Picasso. The green chords are the people who influenced him and the red chords are the people that he influenced.

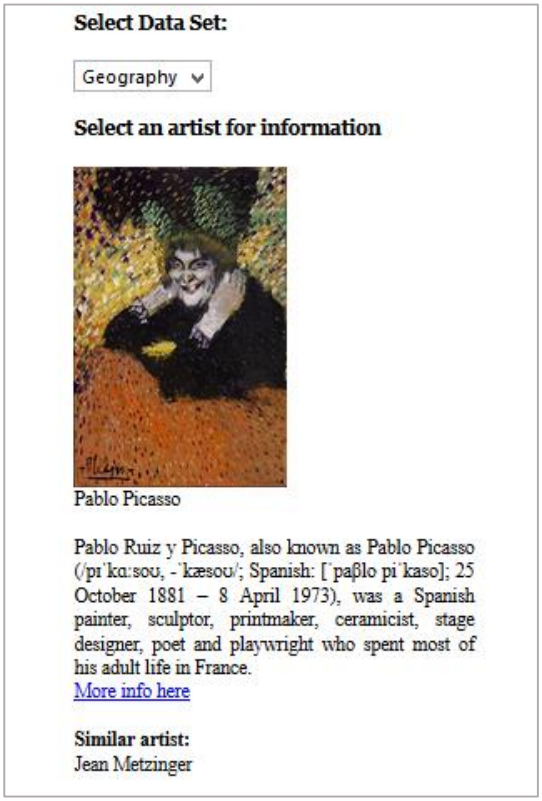

Fig. 5. Dependency Wheel – Pablo Picasso additional information and recommended artist

C. Timeline

The artist timeline is a visualization tool in which the user can see certain artist events (birth, death and art creation) that occurred between two points in time (Fig. 6). Another variable that can be controlled by the user is the number of artists to be displayed (Fig. 7). Their order is taken automatically according to the number of work of arts. Once the mouse is over a node in the timeline, additional information like the location of birth, location of death, artwork name and links to the source appear.

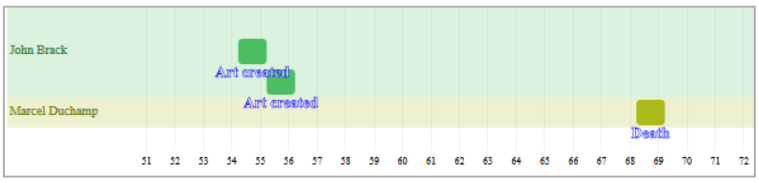

Fig. 6. Timeline snippet of events for the top 5 artists according to number of created work of arts between 1950 and 2000

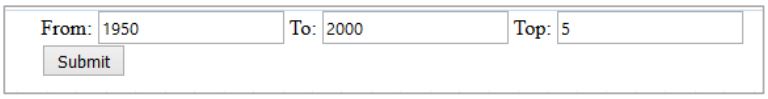

Fig. 7 Timeline controllers

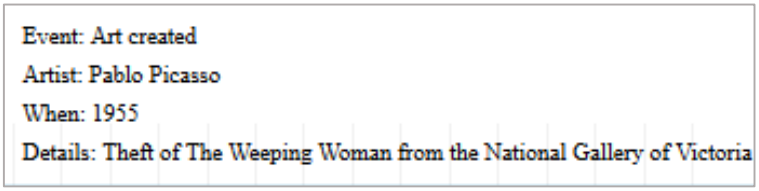

Fig. 8. Timeline additional information when hovering over an event

D. Faceted Browser

The Faceted Browser functionality offers the user a way to visualize artists according to several criteria (birth/death date, nationality, movement, field, gender). The page is formed by two parts: the faceted browser itself displayed as a table, where the columns are the criteria and on the rows the different criteria options (e.g. for the birth criteria there is 1267, 1284, etc.) (Fig. 9) and the results component that shows the results of the selected options under the form of artist name, birth and death year, description, movement and field (Fig. 10). In other words, the user can select several criteria options like birth year 1864 and French as nationality and see how many artists would pass the selected filters. The result for the given dataset would be "*Henri de Toulouse-Lautrec: 1864-1901. French painter Movement::Art Nouveau, Post-Impressionism, Field::Drawing, Illustrator, Printmaking*". Some other interesting observations can be made on the different comparisons between the female-male ratios for example. For the current dataset, where all the artists that contain an ULAN ID were fetched, we can see that the female-male-unspecified ratio is 18:196:11 which shows on one hand that within the artist field, women make only 8% out of the total number of artists and 4.8% profiles don't have the gender submitted.

Fig. 9. Faceted Browser

Fig. 10 Faceted Browser – the results component

E. Concept Map

The concept map has the same purpose as the Dependency Wheel, but under a different visualization style. Its implementation was done in order to acquire D3.js knowledge and to experiment with different types of D3.js attributes and functionalities. (Fig. 11)

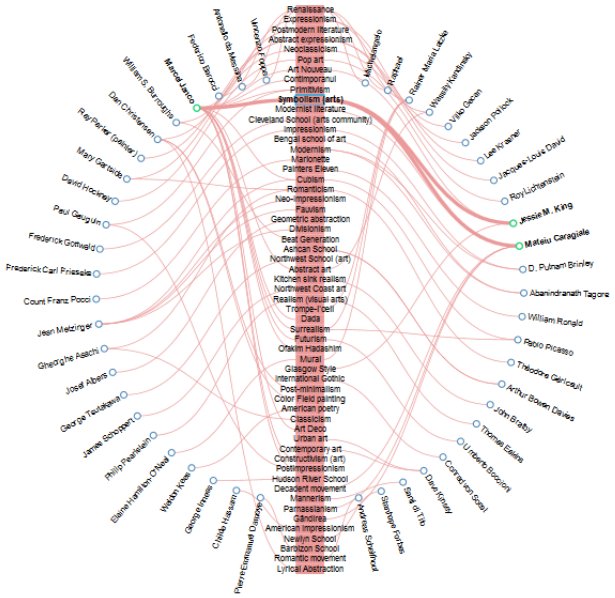

Fig. 11. Concept Map

F. Artist Bar Chart

The Artist Bar Chart brings a quantitative view to the artist data. This platform page was implemented in order to display the number of artworks per artists, but it can be customized to show other metrics as well. (Fig. 12)

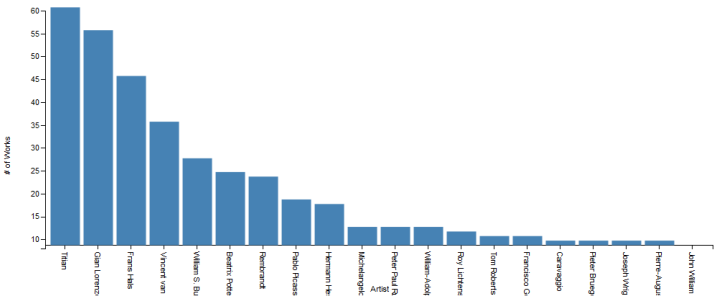

Fig. 12. Artist Bar Chart

G. Natural Language Search Engine

Every static page has a search box where the user can query artist data in a natural language manner (Fig. 13). The technology used is based on Python: Bottle as a server framework, NLTK for Natural Language Processing and Quepy for Natural Language Querying. The current possible questions have the following form:

- Who is Pablo Picasso?
- Where is Pablo Picasso from?
- When was Pablo Picasso born?
- When did Pablo Picasso die?
- Where did Pablo Picasso die?

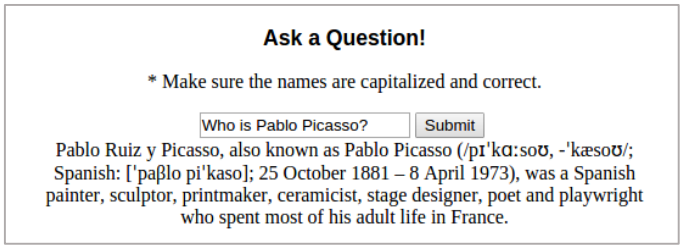

Fig. 13. Question answering system

More technical details will be presented in the next section.

V. IMPLEMENTATION AND TECHNOLOGIES USED

The platform design and implementation went through several essential steps that eventually shaped the final form of the application. We will present next the processes of data retrieval, the creation of use cases, data visualization, implementing the question answering system and an overview of the technologies used.

A. Data retrieval

The first step was analyzing the available data and finding a way to connect multiple sources in order to enrich the information. From the 4 analyzed data sources, none had a complete ontology that contained all the information about art and artists. For example, although DBPedia contains the largest dataset among the 4, the property of movement for an artwork,

cohabitant relationships or gender for an artist was only contained by Wikidata. ULAN vocabulary from Getty contained events, relationships and Europeana contained the biggest number of artworks. Although the number of artworks was bigger than the other data sources, the level of information detail is quite small. Also, there is a language issue, since the data is not structured according to it and some information might be available only in language used by the partner that inserted the data (museum, archive, institute etc.).

Despite these issues, a reasonable solution was to aggregate all the sources and use them in order to enrich the information when needed. This can be done simply by using the ULAN and Europeana IDs. Although the link between ULAN and DBPedia is not complete, there are many artists that contain an ULAN ID. On the other hand, there aren't any mismatches between Wikidata and ULAN. As a result, the connection can be made by matching the ULAN ID in all the 3 sources and then the label of the needed keyword. This brings an intersected dataset of about 225 artists, but the number is continuously changing since DBPedia is updated through Wikipedia.

When it comes to the link between Europeana and the other data sources (Getty, DBPedia and Wikidata), the connection is done through the Europeana ID present in Wikidata and Europeana. Europeana is problematic at the moment since there are language issues, but some information can be used in order to enrich the dataset.

In this manner, all the LOD sources are connected and we can extract all the available information about a specific artist without having to compare their names or other identifying methods that can differ from one source to another and could be error prone.

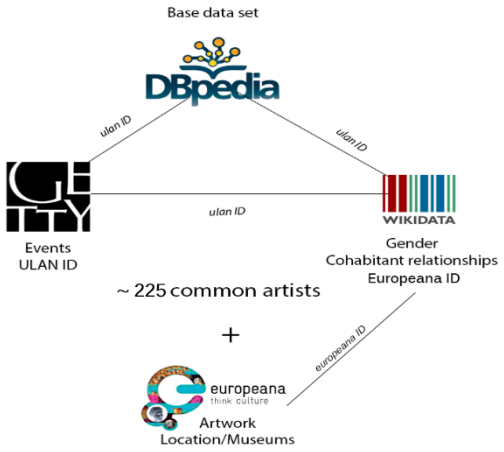

Fig. 13. Connections between all the LOD sources analyzed/used in ArtViz

An additional step in this stage was analyzing other methods of gathering artist data and relationships which meant diving into the Natural Language Processing technologies. This started with searching the best available NLP libraries and seeing how they can be used on a Wikipedia text. Among those analyzed were Python's NLTK, GATE and Stanford CoreNLP. Although not satisfactory, the best results were given by Stanford CoreNLP after applying the coreference component and then relationship component. The reason for this is that the Wikipedia text is rather complex with very long sentences and this causes problems when the coreference component is applied. Although the relationships clearly followed a *subject property object* pattern, the results cannot be used in application without being error prone. [7]

The conclusion of the data gathering and analysis was a dataset containing 225 artists that have a ULAN ID and contain the base data from DBPedia (see Appendix A). This number was registered in October 2015 and is continuously changing as new ULAN IDs are added on Wikipedia. Once this has been set, the next step was finding a way to visualize all the data provided. The initial solution was using tools and libraries like Gephy, JGraph or JUNG. All the tools were analyzed and since the dataset at the time contained around 225 common artists, the relationships between them and the relationship with the people they influenced/were influenced by, a tool that could handle large sets was needed. As a result Gephy was used and this offered some information on the data arrangement[8]. As it can be seen in the screenshot, there is a cluster of very influential artists in the middle, the less-popular artists around the center cluster and isolated nodes around.

B. Creating Use Cases

The platform targets unexperienced users that want to enrich their current knowledge in art or that have little or no background whatsoever. Thinking and creating use cases was also a part of the development so as a result a few scenarios were made.

- Use Case #1: John, the student. The first major user category would be represented by the students that need to do homework and essays on a specific topic (e.g. movement, artist). If they have no previous background in the topic they are researching, they can enter the platform and check different relationships and information about it. This would help them understand and write about the subject from multiple perspectives. In other words, let's say that the student, John, has to write an essay about the movement Cubism. He enters the Dependency Wheel diagram, selects the movement option from the drop-down list and hovers over the Cubism label. Next he can see the major artists that created in this movement and information about them.
- Use Case #2: Jane, the teacher. The second user category treats the situation of art pedagogues and how they can teach and present their art lessons in a more interactive way. Teachers and professors can create screenshots of the diagrams and include them in their art and art history lessons in order for students to understand them easier.
- Use Case #3: Mark, the businessman. The third use case refers to the situation of a businessman who wants to explore the local artwork during a business trip. He can verify artists by geography relationships or ask location-related questions.

C. Data Visualization

Once the purpose of the application was set, the next step was using the provided data and implement the visualization

[7] https://github.com/oana906/ArtViz/blob/master/NLP_relationships.pdf

[8] https://github.com/oana906/ArtViz/blob/master/Gephi_results.pdf

according to it. The suggestion of using D3.js, came as a perfect fit since it offers high flexibility when working with large data sets and it lets the developer control the final form of the design. D3.js or Data-Driven Documents currently supports 3 data formats (CSV, JSON and geoJSON), but since it is a JavaScript library, other functions can be written for other data formats. As a step-by-step process, working with D3.js means the selection of an HTML element, the creation of a SVG object within it (or the bound of datasets to SVG objects) and the appliance of styles, transitions and tooltips elements.

Another intermediate stage that was necessary at an incipient point was filtering the large data set in order to retrieve the most important artist entries. Initially we used Networkx, a Python library for network and graph analysis but due to the fact that the client language was different (JavaScript), we ended up fetching directly the artists that created the most artworks. The Networkx source code used is available on GitHub[9]

D. Natural Language Querying

The last step within the project was implementing a question answering system. The initial experience with NLP was gained when the algorithm for relationship extraction was implemented. Now, every page of the platform has a search box where users can query artist data in a natural language manner. The technology used is based on Python: Bottle as a server framework, NLTK for Natural Language Processing and Quepy for Natural Language Querying. Quepy can be easily customized in order to detect different questions. The mechanism basically transforms a question into a SPARQL query, sends the request with the query to the chosen SPARQL endpoint and gets the response back. The first step, the transformation from natural language to SPARQL is done by using a form of regular expressions. For example, for the question "Who is Pablo Picasso?" the regular expression has the following form:

```
person_name = Group(Plus(Pos("NNP")),"person_name")
regex = Lemma("who") + Lemma("be") + person_name + Question(Pos("."))
definition = DefinitionOf(match.person)
```

where the DefinitionOf method sets the RDF property and additional settings:

```
class DefinitionOf(FixedRelation)
  relation = "rdfs:comment"
  language="en"
```

E. Technology Overview

Architecture wise, the project can be seen as formed on 2 levels: the client side and the server side. The client side deals with all the visualization operations and functionalities while the server side appeared when the question answering system had to be implemented. Technology-wise, the client side uses the expected web languages (i.e. HTML, CSS and JavaScript) and a series of JavaScript libraries. Among the libraries the most important is D3.js. D3.js or Data-Driven Documents had its initial release in early 2011 and since then its popularity grew exponential due to its flexibility and pre-built elements that makes implementing visualization components an easy task. Other used libraries are jQuery, Bootstrap.js and Underscore.js for common tasks, SPARQL.js for fetching data directly from the DBPedia endpoint and Ajax in order to receive and send requests from/to the server side. On the server side, the main technologies used are gravitating around Python. After analyzing a few possibilities like implementing a question answering system from scratch using Stanford CoreNLP or Python library NLTK and due to the short time period, we decided to use an already implemented library called Quepy. Quepy uses Python, NLTK and Regular Expressions in order to identify questions, transforms the question into a SPARQL query, sends the query through a request to the LOD server and returns an answer. In order to transform Quepy into a web service, we have used Bottle, another Python library. At the moment, besides the question answering code, on the server side there is also the data used for the client side visualization. These files (JSON format) underwent through parsing and modifications in order to be easily used by the client side.

Other technologies that need to be mentioned is SPARQL, the query language for RDF used in order to retrieve data from the LOD endpoints and Networkx, a Python library needed for network and graph analysis, used for very large datasets that needed to be filtrated. Java was used in the early stages, when I retrieved data for multiple LOD sources in order to analyze it and Stanford CoreNLP was needed for early NLP processing and relationship extraction. As for the recommender system, it was implemented using JavaScript and two formulas in order to calculate the similarity between two artists or painters: the cosine distance and a custom formula that is presented later on in the similarity diagram.

VI. CONCLUSION AND FURTHER WORK

The results of this project is represented by a platform prototype that gives users the possibility to browse through artist data and gather information in a different way that doesn't revolve around plain reading. Other functionalities like question answering and artist recommendations bring additional value and help in the discovery of new artists and information. Different aspects of the art data like the male-female ratio, geographical distribution, most productive artists or the world's greatest influencers have been underlined.
The project and the source code [10] are available on GitHub along with the presentations and other auxiliary materials and files resulted from the research and project.

REFERENCES

[1] Jain P., *Ontology alignment for Linked Open Data* http://www.knoesis.org/pascal/resources/publications/BLOOMS.pdf [18.08.15]

[2] Micsik A., *LODmilla: a Linked Data Browser for All* http://ceur-ws.org/Vol-1224/paper8.pdf [18.08.15]

[3] Lukovnikov D., *DBpedia Viewer - An Integrative Interface for DBpedia Leveraging the DBpedia Service Eco System* http://events.linkeddata.org/ldow2014/papers/ldow2014_paper_05.pdf [18.05.15]

[4] Chen J., LODStories*: Learning About Art by Building Multimedia Stories,* http://lodstories.isi.edu/LODStories/paper/LODStories_LNCS.pdf [18.08.15]

[9] https://github.com/oana906/ArtViz/tree/master/utils

[10] https://github.com/oana906/ArtViz

APPENDIX A

name	ulanid
Fiona Margaret Hall	500268195
George James Coates	500019332
John Hassall (illustrator)	500001686
Laurie Simmons	500077698
Jerome and Evelyn Ackerman	500299822
Jerome and Evelyn Ackerman	500299823
Emery Walker	500278471
Jacques Androuet II du Cerceau	500033557
Anthony of Padua	500342642
Richard Dadd	500014877
Henri Laurens	500010480
Bernard Accama	500084911
Francis Greenway	500085992
Frederick Scott Archer	500003843
Rosemarie Trockel	500033164
Taddeo di Bartolo	500029513
Alfred Leete	500013277
Hippolyte Destailleur	500017698
Gilbert Jackson	500024408
Edward Bulwer-Lytton	500292901
Alexandre Debelle	500184006
Olivier van Deuren	500031289
Gerard Krefft	500100528
Stanley Grinstein	500353387
Bernardino Castelli	500004774
Clara Southern	500124047
Lorenzo Veneziano	500030488
Pierre-Alexis Delamair	500016506
Philip Richard Morris	500022016
Lawrence Carmichael Earle	500029948
Adolf Dehn	500009264
Colin Colahan	500067294
Ellis Rowan	500069008
John Bettes the Younger	500015018
Thomas Hill (painter)	500021255
Juste de Juste	500099815
Claes Oldenburg	500029735
George Earl Ortman	500000936
Monique Prieto	500329962
Samuel W. Rowse	500031230
John Montresor	500081798
Jean-Gabriel Domergue	500023858
Placido Costanzi	500028648
Thomas de Leu	500032747
Nell Blaine	500016457
Vaughan Grylls	500102717
Doyle Lane	500122183
Reginald Uren	500070854
Belmiro de Almeida	500118117
Antoine Samuel Adam-Salomon	500037061
Ben Ormenese	500340839
Jean Baptiste Androuet du Cerceau	500014087
Ettore Cercone	500046206
Lawrence Alma-Tadema	500008100
Marilyn Monroe	500342163
Tako Hajo Jelgersma	500027457
Guillaume Dubufe	500030836
John William Brown (artist)	500010267
Adriaen van Cronenburg	500014687
Arthur Streeton	500020718
Jane Sutherland	500124033
Pro Hart	500093200
Thomas Francis Dicksee	500019516
Mortimer Menpes	500015549
Wybrand de Geest	500024157
Erwin Olaf	500116847
Maria Sibylla Merian	500009826
H. M. Bateman	500008840
Leslie Garland Bolling	500034510
Ibrahim Kodra	500240266
Kenny Meadows	500018528
Dudley Hardy	500014364
Constance Mayer	500014284
Paolo da San Leocadio	500007663
David Eduard Steiner	500005247
Michael Powolny	500107071
Edward Vernon Utterson	500086172
Subhaprasanna	500122885
Arthur Wallis Mills	500031030

Pierre-Antoine Quillard	500000636
Alfred Jensen	500031242
John Gorrie	500102926
James Sowerby	500128681
Cowan Dobson	500064125
Jacques Boyceau	500103607
Elisabeth Murdoch (philanthropist)	500277785
Jonas Umbach	500022818
Gerard Edema	500011623
Max Hermann Maxy	500030192
Hans Vredeman de Vries	500006358
Olafur Eliasson	500116131
Giannicola di Paolo	500005722
Pierre-Antoine Demachy	500024480
Coppo di Marcovaldo	500115281
Deodato Orlandi	500007771
Henry Walton (painter)	500010151
Michael Goldberg (painter)	500002151
Robert Ingpen	500058103
Wigerus Vitringa	500003483
Albert Tucker (artist)	500041112
Chan Canasta	500024173
Jules Joseph Lefebvre	500013504
Tom Roberts	500028066
Dora Meeson	500032442
Jimmy Pike	500124123
Johnny Bulunbulun	500330870
Domenico da Cortona	500115865
Lucien Clergue	500115463
Sandra Goldbacher	500101857
Guy de Gisors	500087364
Letitia Byrne	500032842
Armand Laroche	500000681
Albert Laprade	500032667
Albert Rigolot	500053468
Frederick Thomas Dalton	500103137
Henri Zuber	500021494
Bernard Perlin	500024499
Giovanni Francesco Maineri	500020981
Maitland Armstrong	500002573
Paul Androuet du Cerceau	500041510
Leonardo da Vinci	500010879

Jordan National Gallery of Fine Arts	500304974
Lucian Freud	500116243
Ivor Hele	500026702
Giulio Rosati	500092889
Vittorio Sgarbi	500288963
William Ellis (engraver)	500032823
Thomas Davies (British Army officer)	500124785
Giovanni da Milano	500012251
Joseph Simon Volmar	500075144
Richard Saul Wurman	500222770
Sampson Strong	500010458
Hill & Adamson	500041217
Oluf Braren	500090410
Abraham Hirsch (architect)	500235100
Catalina Parra	500061642
Colette Whiten	500063760
Jakab Marastoni	500020076
Jan Kamphuysen	500102082
Abraham Lincoln	500344436
Albert Einstein	500240971
Alexander von Humboldt	500023604
D. H. Lawrence	500005716
Henry David Thoreau	500229765
Joseph Dalton Hooker	500004899
Washington Irving	500231645
Ralph Steadman	500022057
Thomas Phillips	500020549
National Museum of Korea	500308894
David Haines (artist)	500355630
William Charles Ross	500028240
Elizabeth Nourse	500009983
Henri Sauval	500283015
John Bettes the Elder	500030213
City Museum of Ljubljana	500301610
Alphonse de Gisors	500233612
William Heath (artist)	500018507
C. Y. Lee	500110349
Agnes Martin	500024489
John Pollard Seddon	500000430
Madeline Gins	500112518
Burr H. Nicholls	500082708
Emily Sartain	500092505

Gilles Le Breton	500089255
Carroll Dunham	500110733
Flinders University Art Museum	500307577
Kanuty Rusiecki	500121212
Henri-Gabriel Ibels	500016306
Jan Stolker	500029097
John Munsterhjelm	500065175
Louisa Anne Meredith	500029259
Jean Androuet du Cerceau	500032925
Constance Stokes	500174158
Jogen Chowdhury	500122703
Margaritone d'Arezzo	500025562
Antoni Viladomat	500124473
Charles Wild	500119661
Giuseppe Pellizza da Volpedo	500028941
Georgiana Burne-Jones	500030326
Nicolaas Baur	500032684
Bill Brandt	500026943
Exene Cervenka	500128102
Peter Eisenman	500025316
Phil May (caricaturist)	500023489
Janet Cardiff	500116193
Robert De Niro, Sr.	500002287
Jean-Baptiste Belin	500002361
John Hungerford Pollen (senior)	500009806
Nicolas Edelinck	500098256
Fateh Moudarres	500124620
Elliott & Fry	500116795
Robert Beauchamp	500018294
Carl Guttenberg	500031904
Jane Freilicher	500017408
Juan Davila (artist)	500105779
Antonio da Vendri	500045912
Benedetto Gennari II	500027649
Erastus Salisbury Field	500021230
Coosje van Bruggen	500032593
Jeffrey Smart	500016674
Marco d'Oggiono	500019724
William F. Cogswell	500016038
Modesto Faustini	500003583
Pyotr Petrovich Sokolov (painter)	500120644
Pieter Feddes van Harlingen	500029166
Hendrick Van Cleve	500031503
Moses van Uyttenbroeck	500017388
Claude Raguet Hirst	500019148
Sukumar Ray	500122894
Jan Frans van Dael	500007049
Claudio Castelucho	500035809
Hjalmar Munsterhjelm	500065178
Ralph Eleaser Whiteside Earl	500022904
Ursula Reuter Christiansen	500084964
Aachen Cathedral Treasury	500304388
Florence Fuller	500073416
Pere Moragues	500039303
Walter Runeberg	500097870
C. H. Collins Baker	500008062
Lodewijck van Ludick	500004647
Carl Ludwig Jessen	500018033
Austin W. Lord	500098316
Suermondt-Ludwig-Museum	500264335