Graph Visualization of the Characteristics of Complex Objects on the Example of the Analysis of Politicians

Mikhail Ulizko[1[0000-0003-2608-8330]], Evgeniy Antonov[1,2[0000-0003-1498-9131]],
Alexey Artamonov [2[0000-0002-9140-5526]] and Rufina Tukumbetova [1[0000-0002-1976-1390]]

[1] Plekhanov Russian University of Economics, 36 Stremyannyy per., 115093 Moscow, Russia
mulizko@kaf65.ru
eantonov@kaf65.ru
rrtukumbetova@kaf65.ru
[2] National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Kashira Hwy, 31, 115409 Moscow, Russia
aartamonov@kaf65.ru

Abstract. The paper considers the task of analyzing complex interconnected objects using graph construction. There is no unified tool for constructing graphs. Some solutions can build graphs limited by the number of nodes, while others do not visually display data. The Gephi application was used to construct graphs for the research. Gephi has great functionality for building and analyzing graphs.
The subject of research is a politician with a certain set of characteristics. In the paper an algorithm that enables to automate data collection on politicians was developed. One of the main methods of data collecting on the Internet is web scraping. Web scraping software may access the World Wide Web directly using the HTTP, or through a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a software agent.
The data was necessary for constructing graphs and their analysis. The use of graphs enables to see various types of relationships, including mediate. This methodology enables to change the attitude towards the analysis of multidimensional objects.

Keywords: Web Development, Graph, Visualization, Politicians.

1 Data collection and processing

In the modern world there is a large amount of information that can be represented in the form of objects and the relationship between them. For example, objects can be ыindividuals, the relationship between which is described by a certain characteristic. In this case, graphs are often used for analysis [1].

Political forces often play an important role for the development of scientific and social activities within individual states and the world community. Decisions made by political forces may depend on the individual characteristics of the person (gender, age, fraction) and the relationship between each other.

Information about political figures is also posted on the Internet. It enables to obtain data in sufficient quantities. It is unreasonable to collect data on politicians manually, since there is a lot of information. Therefore, algorithms are being developed that enable to collect data from information sources of the Internet.

To build graphs for political leaders, it is necessary to determine the nodes of the graphs and describe the relationship between them.

Graph plotting is an unnatural process for the computer. With existing software it is rarely possible to construct large graphs or have the ability to interact. However, with the correct construction, the graphs enables a qualitative data analysis and identify both explicit and hidden relationships [2-4].

2 Methodology

2.1 Collecting data about policies on the Internet

One of the main methods of data collecting on the Internet is web scraping. The general principle of web scraping can be represented as follows: the program code sends a request to the target source and receives a response in the form of HTML code, after which it searches for the required information using the XPath query language.

Requests are most often generated using the HTTP protocol; they use the GET request method to receive data.

The response is an object that is predisposed to receive any data. In data processing, an HTML document is extracted from the response, after which a search is performed, and the result is converted to the required format. This process is called parsing.

Web scraping can be done in two ways:

- development of software;
- use of third-party software (including API).

If it is necessary to use several information sources at the same time for data extraction, the second method is usually not used. Automate data collection during the development of software can be through the use of agent technologies (software agents) [5, 6]. Software agent is a computer program that acts on behalf of a user on demand or according to a schedule. Additional software packages can be also used to interact with the browser. When developing software to solve this problem, the programming languages Python and JavaScript are most often used.

Information about politicians is available on various Internet sources. As an example, information from such sources as VoteSmart and Govtrack is considered [7, 8]. They have a complex structure, so it is impossible to use third-party software. For the development, we used the Python programming language and software packages of the language, such as selenium, lxml.

The data collection algorithm is divided into three subtasks:

- GET request to an information source;
- extracting data from the response body (HTML document) using Xpath;
- saving the received data with the possibility of visualization.

Thus, the algorithm is as follows (see Fig. 1).

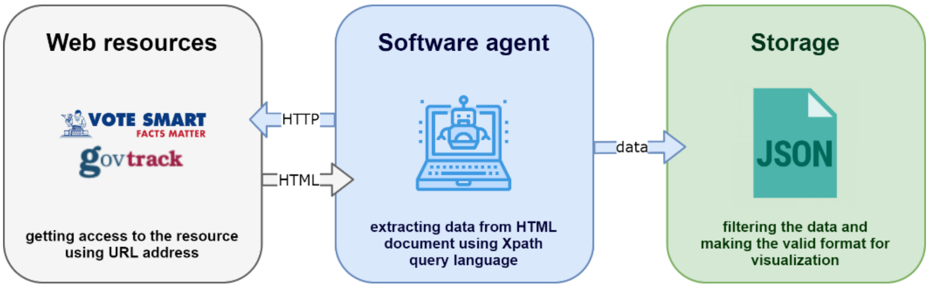

Web resources
VOTE SMART
FACTS MATTER
govtrack
getting access to the resource using URL address
HTTP
HTML
Software agent
extracting data from HTML document using Xpath query language
data
Storage
JSON
filtering the data and making the valid format for visualization

Fig. 1. Data collection algorithm

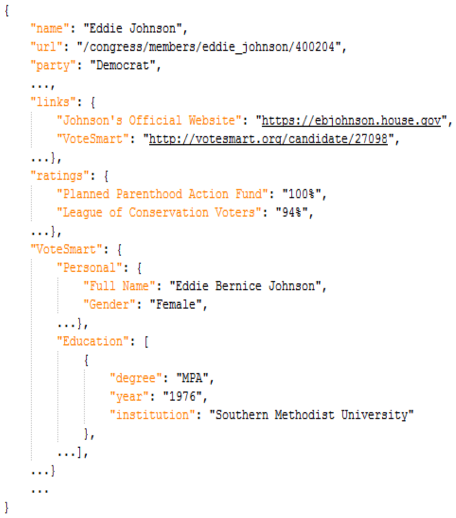
JSON is selected as the data presentation format. The choice is due to the complex nested structure of the raw data, which is difficult to process when stored in CSV format. Meanwhile, the XML format is redundant [9].

2.2 Graph representation of personality

For the analysis of interrelated objects graphical representation of the data are used. A graph consists of vertices (nodes) and edges that form a connection between vertices. In order to determine the nodes and edges, we select the characteristics of the person obtained during data collection:

- Name;
- Party;
- District;
- Ratings;
- Bills;
- Religion;
- Education;
- Political experience;
- Current Legislative Committees
- Former Committees;
- Professional experience;
- Other organizations;
- Additional information (awards, favorite quotes, etc.).

The figure (see Fig. 2) shows an example of the data collected by a politician in json.



```
{
    "name": "Eddie Johnson",
    "url": "/congress/members/eddie_johnson/400204",
    "party": "Democrat",
    ...,
    "links": {
        "Johnson's Official Website": "https://ebjohnson.house.gov",
        "VoteSmart": "http://votesmart.org/candidate/27098",
    ...},
    "ratings": {
        "Planned Parenthood Action Fund": "100%",
        "League of Conservation Voters": "94%",
    ...},
    "VoteSmart": {
        "Personal": {
            "Full Name": "Eddie Bernice Johnson",
            "Gender": "Female",
        ...},
        "Education": [
            {
                "degree": "MPA",
                "year": "1976",
                "institution": "Southern Methodist University"
            },
        ...],
    ...}
    ...
}
```

Fig. 2. Example of politician data

Due to the fact that a person is characterized by many characteristics, when constructing a graph for politicians, two approaches arise:

- the node of the graph is a person, an edge - some of the properties;
- several types of nodes of a person are selected, edges - the correspondence of nodes in the raw data.

In the first approach, for the trivial case, nodes [7, 8] can be connected by an edge of the same type (for example, by belonging to the same party). In this case, between the nodes either there is or there is no a node. The approach can be expanded by introducing edges for each of the considered properties and introducing markers (color, weight) to distinguish between the edges. For clarity, the different types of edges should be no more than 5.

For the second approach, the nodes are unique values of characteristics; edges are the relationship between these characteristics in the raw data. To construct such a graph, it is required to consider in what form the information is presented.

The raw data are separate JSON files, which contain information about representatives of some of the largest US states: Texas, California and Florida. To construct a graph it is necessary to explicitly distinguish nodes and edges. We divide the nodes into categories, while all nodes can be connected by edges only through the node identity.

In the typical case a graph is defined by an adjacency matrix, but other methods are often used. In particular, if there are many “zeros” in the adjacency matrix, which indicate that there is no connection between nodes, the graph is specified with a list. The list consists of entries of the following form (1):

$$< v_i, v_j, w_{i,j} >, \text{ where} \tag{1}$$

$$v_i - \text{starting node}, v_j - \text{end node}, w_{i.j} - \text{ edge weight}$$

In this case the graph will be undirected. The edge only indicates the connection between the nodes, not the nature of the connection. We distinguish the following categories of nodes:

- Person;
- University;
- Religion;
- Membership in a civil organization;
- Party.

Because of the initial data representation one of the nodes of each edge will be a ‘person’ node.

The graph may be weighted and unweighted. Weight can be introduced if there is a one-to-many relationship between objects. In particular, such a relationship is formed in such node as ‘person – university’ and “person - membership in a civil organization”. In this case, the weight of the edge will be specified and show the relative share of the university / organization for a particular person. Mathematically this is given by a formula (2):

$$\sum_{i=1}^{k_j} w_{j,i} = 1 \; for \; \forall j, \; where \tag{2}$$

$$j - \text{"person" vertex}, i - \text{institutions connected with vertex j}$$

The Python programming language is used to obtain the graph structure. The developed algorithm converts the raw JSON data into nodes and edges, which are stored in separate CSV files.

2.3 Visualization and analysis

There is no unified tool for constructing graphs. Some solutions can build graphs limited by the number of nodes, while others do not visually display data [10, 11]. One commonly used application is Gephi [12].

Gephi has great functionality for building and analyzing graphs. To build a graph we want to colorize the nodes depending on the type of a characteristic. For drawing graphs, several algorithms are proposed, we choose Force Atlas. The algorithm is based on minimizing energy (the nodes are iteratively attracted or repelled from each other in the visualization space, depending on their relative position and the presence of connections), which allows graphs to be constructed with a high degree of interaction. First we colorize the nodes according to the type of a considered characteristic.

We construct a graph for the raw data by adjusting the display of output. The resulting graph will have the following form (see Fig. 3).

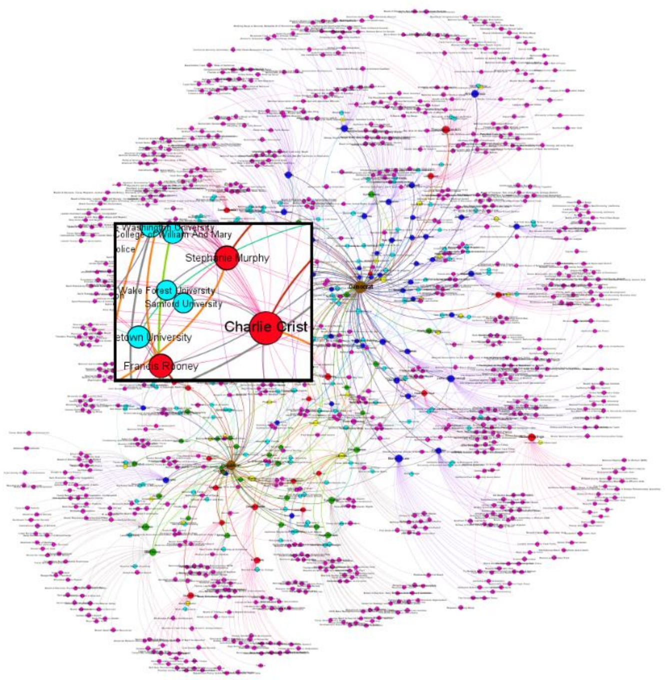


Fig. 3. The resulting graph

In the figure, the size of a vertex is directly related l to the number of edges adjacent to it, and the thickness of an edge is directly related to its weight. The following colors are used in the figure:

- purple – organization;
- brown – party;
- yellow – religion;
- blue – university;
- red – a politician from the state of Florida;
- green – a politician from the state of Texas;
- blue – a politician from the state of California;

Due to the layout method, the graphs were divided into two clusters: Democrats and Republicans, but due to the large number of nodes, the data cannot be analyzed in detail.

Filters can be used to refine the information. For example, the following filtering result will show which university representatives of the states most often graduate from. (see Fig. 4).

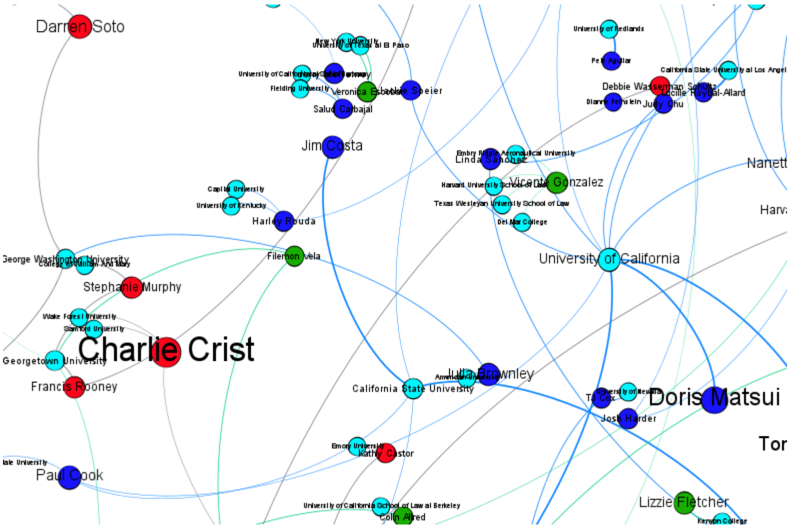


Fig. 4. Connection between representatives and educational organizations

You can also view information about the relationship of an individual object or group of objects. The figure shows the links between the «Henry Cuellar» and «Darren Soto» nodes (see Fig. 5).

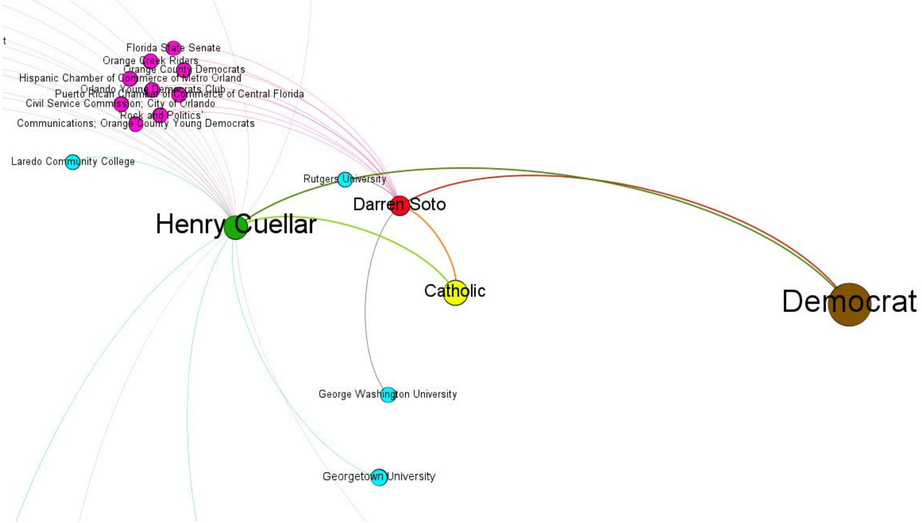

t
Florida State Senate
Orange Creek Riders
Orange County Democrats
Hispanic Chamber of Commerce of Metro Orland
Orlando Young Democrats Club
Puerto Rican Chamber of Commerce of Central Florida
Civil Service Commission; City of Orlando
'Rock and Politics'
Communications; Orange County Young Democrats
Laredo Community College
Rutgers University
Darren Soto
Henry Cuellar
Catholic
Democrat
George Washington University
Georgetown University

Fig. 5. Graph filtered by personality

These persons are close to each other and it can be assumed that their views are similar. As we can see from the graph, despite the fact that they are from different states, they have a connection with the same organizations. Both are Catholic by religion and belong to the Democratic Party.

3 Conclusion

The task of visualizing linked multidimensional objects is relevant in the field of data analysis. The described technology offers a working approach to constructing graphs for an object such as a politician.

The division of one object of an entity of different nature allows you to build relationships between objects, to draw certain conclusions about them. In particular, the most graduating educational institution is the University of California, graduates of this university become democrats. The construction of several graphs allows us to analyze in more detail and find internal relationships.

It seems that for the sake of completeness, it is necessary to supplement the data for all US congressmen, which truth will lead to a significant increase in relations and the need to consider the graph to solve practical problems, whether it be an analysis of educational institutions or lobbying organizations, or religions depending on the states, etc.

The use of such graph models seems extremely promising from the point of view of analyzing a large amount of information not only on such a complex object as a person, but also on organization technology, etc.

References

1. Kulik, S., Shtanko, A.: Using convolutional neural networks for recognition of objects varied in appearance in computer vision for intellectual robots. Procedia Computer Science 169, 164-167 (2020).
2. Onykiy, B., Artamonov, A.A., Tretyakov, E.S. Ionkina, K.V.: Visualization of large samples of unstructured information on the basis of specialized thesauruses. Scientific Visualization 9(5), 54-58 (2017).
3. Tretyakov, E.S., Tukumbetova, R.R., Artamonov, A.A.: Methodology of Analysis of Similar Objects with the Use of Modern Visualization Tools. Mechanisms and Machine Science 80, 113-119 (2020).
4. Artamonov, A.A., Leonov, D.V., Nikolaev, V.S., Onykiy, B.N., Pronicheva, L.V., Sokolina, K.A., Ushmarov, I.A.: Visualization of semantic relations in multi-agent systems. Scientific Visualization, 6 (3), 68-76 (2014).
5. Kulik, S.D., Shtanko, A.N.: Experiments with Neural Net Object Detection System YOLO on Small Training Datasets for Intelligent Robotics. Mechanisms and Machine Science 80, 157-162 (2020).
6. Kulik, S.D.: Neural network model of artificial intelligence for handwriting recognition. Journal of Theoretical and Applied Information Technology 73(2), 202-211 (2015).
7. Votesmart, https://justfacts.votesmart.org, last accessed 2020/06/10.
8. Govtrack, https://www.govtrack.us, last accessed 2020/06/10.
9. Grigorieva, M.A., Aulov, V.A., Golosova, M.V., Gubin, M.Y., Klimentov, A.A.: Data knowledge base prototype for modern scientific collaborations. In: Selected Papers of the 7th International Conference Distributed Computing and Grid-technologies in Science and Education, pp. 26-33. CEUR, Dubna (2016).

10. Galkin, T., Popov, D., Pilyugin, V., Grigorieva, M.: The visualization method pipeline for the application to dynamic data analysis. In: Proceedings of the 27th Symposium on Nuclear Electronics and Computing, pp. 295-299. CEUR, Budva (2019).
11. Galkin, T.P., Grigorieva, M.A., Klimentov, A.A., Korchuganova, T.A., Milman, I.E., Pilyugin, V.V., Titov, M.A.: Visual cluster analysis for computing tasks at workflow management system of the ATLAS experiment at the LHC. In: GraphiCon 2018 - 28th International Conference on Computer Graphics and Vision, pp. 111-114. GraphiCon Scientific Society, Tomsk (2018).
12. Gephi, https://gephi.org, last accessed 2020/06/28.