# Mining Linked Datasets

## Jorge Martinez Gil

## Abstract

Linked Open Data is a paradigm for linking the data available on the Web in a structured format in order to make it accessible for computers and people. Unfortunately, not many tools for mining hidden information from existing linked datasets have been proposed in the literature yet. This paper addresses the design and development of TheMa which is an Application Programming Interface (API) for making explicit, information contained in a Linked Dataset. To do this, we have designed and implemented a set of useful functions based on network theory and demonstrated its usefulness when mining implicit information from real world linked datasets. Moreover, we have deployed a web application so that interested users can freely use our software for their own tasks.

**Keywords:** Linked Open Data, Graph Mining, Network Theory

# Introduction

Linked Open Data (LOD) is an emerging paradigm for connecting the data available on the World Wide Web (WWW) in a well defined way so that it becomes accessible for computers and people (Bizer, 2009). Nowadays, the amount of linked data available on the WWW is growing very quickly. The data reflect a wide range of resources such as institutional data, user-generated content, and so on. With more and more sources publishing their content in the form of linked data, the amount of data is exploding, and therefore it is becoming very difficult to process it. Therefore, the development of new techniques and tools for facilitating this task is a big challenge for the research community.

Therefore, dealing with this huge amount of linked data content can be seen to constitute a challenge for many analysts. For example, exploiting implicit information from the linked datasets can lead to improved effectiveness of information retrieval operations, or to extract very useful information that is not explicit in its current form which can lead to important competitive advantages (Auer & Lehmann, 2010). Unfortunately, not many software tools for mining information from these kinds of datasets have been reported in the literature or are not yet available.

In this paper, we present TheMa (derived from Thessaloniki and Malaga which are the affiliations of the authors) which is an Application Programming Interface (API) suitable for calculating useful information related to the Linked Datasets available on the WWW. This information includes the identification of the most prestigious nodes, the key predicates, the link cycles or the reachability for the nodes among other many statistics concerning the linked datasets. Only simple but useful methods from the field of the network theory have been included in this first version of our API. However, to the best of our knowledge, this API is one of the first reported attempts to provide a set of useful methods for analyzing and mining Linked Datasets available on the Web.

The remainder of this paper is organized as follows: Section 2 presents the state-of-the-art related to the data mining techniques for dealing with Linked Data. Section 3 describes our contribution, including the preliminary notions that are necessary to understand our work, the design decisions and the development details taken into account when developing TheMa. Section 4 evaluates our implementation using a linked

dataset extracted from DBPedia[1]. In section 5, we extract the main conclusions drawn from this work and suggest future lines of research. Finally, we describe a web application deployed so users can test our API.

# State-of-the-art

The great amount of linked data in the form of RDF triples which is out there on the Web can be considered an important step forward on the way to reaching a truly structured web which may allow not only humans, but also computers to understand the data, information or knowledge available on the WWW. In fact, a number of software applications benefit from billions of triples in the repositories such as the current DBPedia (Bizer et al, 2009). At the same time, experts belonging to such domains such as finance, medicine or bioinformatics, are demanding more formal and expressive knowledge models for dealing with their data.

Therefore, a very important challenge for linked data mining consists of dealing with the problem of mining structured datasets, where the entities are linked in some way. Links among entities belonging to the dataset may demonstrate certain patterns, which can be helpful for many mining tasks and are usually very hard to capture with traditional statistical models over other kinds of databases.

This kind of problem has been traditionally studied by the link mining community which refers to this situation as "data mining techniques that explicitly consider these links when building predictive or descriptive models of the linked data" (Getoor, 2003).

Commonly addressed link mining tasks include object ranking, group detection, collective classification, link prediction and sub graph discovery (Chakrabarti & Faloutsos, 2006). Therefore, mining Linked Data can be useful in many analog situations, some of which are explained below.

## Use Cases

The problem of mining linked data on the Web is becoming relevant as more and more information is made available online. Some of the most popular mining tasks concentrate on

**Finding cycles in citation data** (White & McCain, 1989). Scientific citation indexes are very important in order to know the productivity of scientists. However, the citation systems can be easily altered by forming citation cycles which do not reflect exactly how scientific production has been influenced. Finding citation cycles can help experts to more exactly determine the scientific productivity of researchers.

**Finding cycles in web pages** (Gibson et al, 1998). There are many algorithms for ranking websites or resources which are based on the prestige of said websites or resources. This prestige is computed taking into account the number of input links that directly or indirectly point to the given resources. Taking into account the link cycles in this context can prevent owners of these kinds of resources increasing the prestige artificially.

**Identification of customer networks** (Hopcroft et al, 2003). Mining Linked Datasets can help us to provide a better understanding for the inclusion of individuals in a particular market segment. This can be very useful from the point of view of those organizations who wish to cluster people based on a given common profile.

[1] http://www.dbpedia.org

**Identification of crime or fraud networks** (Baker & Faulkner, 1993). Mining Linked Datasets can help experts to identify possible fraud scenarios by discovering fraud indicators and connections between nodes. Obviously, it is supposed that criminals are not going to publish their data on the WWW, but for example, institutional data about public funds is usually published and many misuses can be discovered using computer algorithms.

These are only a few examples, but we are confident that over time, further areas of application for this type of software will be proposed by users and practitioners.

# Contribution

Firstly, we are going to provide the formal aspects of our model. Then, we are going to explain the design and evaluation of our API.

A Linked Dataset is a set of triples LD = {(S, P, O)} where

- S is a concept which is called subject
- O is a concept or a literal data which is called object
- P is an ordered pair which includes S and O. It is called predicated

A predicate p = (S, O) is always directed from S to O; O is also called the head and S is called the tail of the predicate; O is said to be a direct successor of S, and S is said to be a direct predecessor of O. If a path leads from S to O, then O is said to be a successor of S and reachable from S, and S is said to be a predecessor of O.

So, a Linked Dataset LD is called symmetric if, for every predicate in LD, the corresponding inverted predicated also belongs to LD. For the rest of this section we are going to explain the design and development details of TheMa.

## Design of TheMa

We found inspiration in network theory in order to design our API. The reason is that network theory provides the foundations concerning the study of graphs as a representation of relationships between discrete objects which exactly coincides with our proposed data model.

In this first version of our API, we have decided to implement such basic operations as: computation of the prestige for the nodes, discovering the bridges, computation of the reachability for a given node, cycle finding or closure computation. All these methods are overloaded, thus, are offered under different versions so that users can select the most appropriate function for each situation. We are going to explain the details for these operations now.

### Prestige

Prestige relates to the reputation/importance a node has in a dataset. To represent this measure we count the links that converge on or originate from this node. The more links that converge on or originate from this node, the more prestigious the node. We divide the prestige into two categories, the out-going prestige and in-coming prestige.

With the out-going prestige we measure those links in which this node is a subject and points to others. A very prestigious node is a node that appears as a subject in many triples in the dataset. We can say that we

can get lots of information for this node in our dataset, so if we process our dataset the result for this node will be more reliable than the others.

With the in-going prestige we measure the links that point to a specific node and this node appears as an object in these links. A node with a very high in-going prestige seems to be a very important node for the dataset because it represents useful information for most of the nodes in the dataset and leads us to obtain more information about this node if we follow the graph, the information of which we did not know, and this could help us process the data more efficiently to find the hidden information in the dataset. More formally,

**Definition 1 (Input Prestige).** *The input prestige of a node n is the number of predicates terminating at n.* Figure 1 shows an example for the Input Prestige.

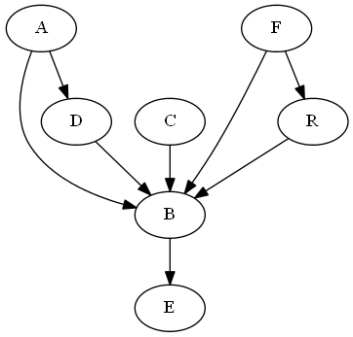


*Figure 1: The input prestige for node B is 4.*

**Definition 2 (Output Prestige).** *The output prestige of a node n is the number of predicates beginning at n.* Figure 2 shows an example for the Input Prestige.

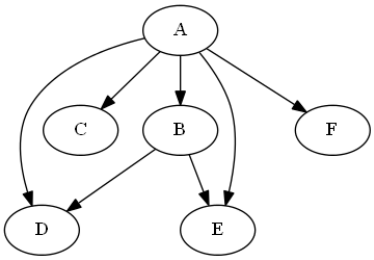


*Figure 2: The output prestige for node A is 5.*

## Bridges

A bridge of a given dataset is a predicate connecting two nodes (the subject node and the object node) the deletion of which means there no longer exists an alternative path from the given subject node to the given object node. Equivalently, the removal of a bridge from a dataset results in the splitting of the latter into two datasets To distinguish in which cases this predicate is a bridge and in which it is not, because this predicate can appear several times in a dataset, we keep as a key the whole triple in which this predicate appears and connects the two nodes that otherwise would be disconnected. More formally,

**Definition 3 (Bridge)**. *A bridge is a predicate whose removal disconnects a Linked Dataset. (For example, a dataset with the form of a tree is made up entirely of bridges.) A disconnected Linked Dataset is a set of predicates whose removal increases the number of components.* Figure 3 shows an example of a bridge.

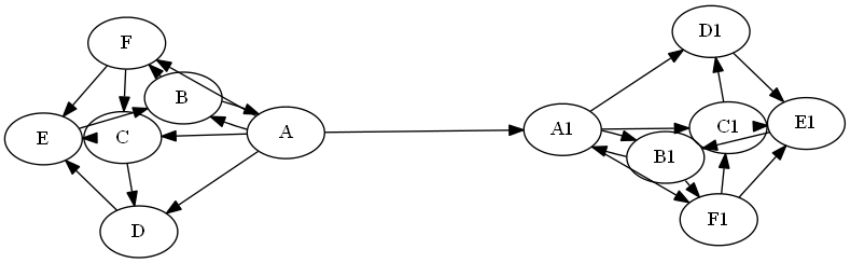


*Figure 3: The connection from node A to node A1 is a bridge in the dataset.*

The statements which include those predicates which are bridges are the "weak" points of the dataset because we can disconnect the dataset if we delete these statements and as a result we will lose important information.

If we collect all the bridges in the dataset we can create a critical path inside the graph and use it in order to evaluate the importance of the result we get from another procedure or we can use it to discover the importance of the connections that we are interested in.

## Reachability

**Definition 4 (Reachability).** *Reachability is a measure that counts how many nodes we can reach from a specific node.*

Mean reachability is the mean value of the nodes that can be reached in the whole dataset. Combining these two measures we gain useful information about the connections in our dataset and how we can navigate through our dataset. Moreover if we add and process the results from the Bridge/Statements percentage method in parallel, we will have a clearer view of our dataset and its cohesion.

## Cycles

**Definition 5 (Cycle)**. *Cycle in the dataset is a path which starts from a specific node in the dataset and if we follow it we return to this same node. .*

This method can take a node as a start and return to it if there is a cycle around it. Also we can return to all the cycles that appear in the dataset. Moreover these methods are overloaded and we can choose which predicates we are interested in to find out if there is a cycle that includes them only. Figure 4 shows an example explaining the notion of cycle.

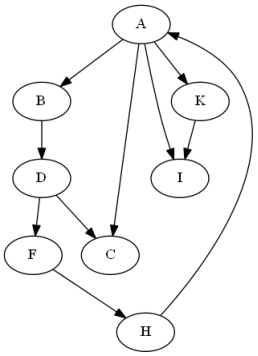


*Figure 4: The path from A -> B -> D -> F -> E -> A is a cycle in the dataset.*

### Closure

Closure is the set of nodes that can be reached from a given one. More formally,

**Definition 6 (Closed set of vertices)**. A set *S* of nodes is said to be *closed* if each successor of each node *S* is also in *S*. Equivalently, *S* is closed if it has no outgoing predicate.

We assume that *LD* is a directed acyclic graph. For, if it is not acyclic, each of its strongly connected components must either be entirely contained in or entirely disjointed from any closed set. Therefore, the closures of *LD* are in one-to-one correspondence with the closures of the condensation of *G*, a directed acyclic graph that has one node for each strongly connected component of *G*.

# Development of TheMa

In order to develop these algorithms we have used Jena[2] as a supporting software framework. Jena is a framework for building semantic applications. It is very useful for us because it provides a programmatic environment for RDF, RDFS and OWL, SPARQL.

### Prestige

As we commented earlier, these two methods are overloaded and we can get the in-going/out-going prestige for the whole dataset or we can create a new smaller dataset by choosing the predicates we are interesting in. The result in both cases is a Map data structure. We have the name of the node as the key in the Map, and the count of the prestige of the node as the value.

### Bridges

We offer two versions of this method for determining the bridges: One that computes the bridges in the whole dataset, and another that computes the bridges only between the resources (we do not include the literals in the results). These methods are (also) overloaded and can be performed on more specific models by creating them by means of the predicates interesting for us. However, we may lose some information in this case.

### Bridge/Statements percentage

The percentage of statements that comprise bridges is an useful measure regarding to the cohesion of our dataset. Applying the classic graph technique about the cohesion in directed graphs is not useful because by deleting statements we are losing information. This has not sense in linked data since the information in linked data is in the links that appear within the nodes. So a better approach is this statistic measure that tells us the percentage of the statements which are bridges. If this percentage is high that means that our dataset is not strongly connected and we can disconnect it very easy. If it is low, we have that most of our nodes are connected more than once and we can say that our dataset is strongly connected and it hard to split it and lose information.

### Reachability/Mean Reachability

Reachability is a statistic measure that counts how many nodes can be reached from a given one. Mean reachability is the mean value of the nodes that can be reached in the whole dataset. By combining these two measures we can get useful information about the connections in our dataset, and how we can navigate

[2] http://jena.sourceforge.net

through our dataset. Moreover, if we add and process the results from the Bridge/Statements percentage method in parallel, we will have a more clear view about our dataset and its cohesion.

### Cycles

This method can take a node as a start and return if there is a cycle around it. It is also possible to return all the cycles that appear in the linked dataset. Moreover, these methods are overloaded (most of times, it has not sense to detect cycles with different predicates), and therefore, we can choose the predicates we are interesting in to know if there is a cycle that include them.

### Closure

This method computes the closure for a node given by the parameter node. Once again, we offer the option to only compute the closure for a node taking into account the predicate types defined in a list.

# Results

In order to evaluate our proposal, we have created a dataset on South American countries from dbpedia.org (Data can be downloaded from the website http://khaos.uma.es/thema). Randomly we chose some predicates from the dataset and we created a list from them, then we selected some of them and we created a second list in order to implement the algorithms in which you have to choose which predicates you are interested in. We implemented the algorithms in this dataset and the results are summarized below.

**Input prestige:**

a) The most prestigious node is “http://dbpedia.org/resource/ Argentina” which has a value 2001.That means that the node “http://dbpedia.org/resource/ Argentina” appears as an object in the dataset’s statements 2001 times, more times than all the other nodes.
b) The most prestigious node in the dataset which was created with the statements which have as predicate one from the first list that we created earlier, is “http://dbpedia.org/resource/Colombia” with value 1377 which means that there are 1377 statements which have this node as object.
c) The most prestigious node in the dataset which was created with the statements which have as predicate one the first list that we created earlier and where we choose only the statements with predicate from the second list, is “http://dbpedia.org/resource/ Colombia” with value 1289. That means that there are 1289 statements which have this node as a subject.

**Output prestige:**

a) The most prestigious node is “http://dbpedia.org/resource/Suriname” which has a value 182. That means that the node “http://dbpedia.org/resource/Suriname” appears as a subject in the dataset’s statements 182 times, more times than all the other nodes.
b) The most prestigious node in the dataset which was created with the statements which have as predicate one from the first list that we had previously created, “http://dbpedia.org/resource/System_of_Cooperation_Among_the_American_Air_Forces” with value 8 which means that there are 8 statements which have this node as a subject.
c) The most prestigious node in the dataset which was created with the statements which have as predicate one from the list that we had previously created, and after, we chose only the statements with predicates from the second list, “http://dbpedia.org/resource/System_of_Cooperation_Among_the_American_Air_Forces” with value 8. This means that there are 8 statements which have this node as a subject.

**Bridges/Statements Percentage:**

a) In the dataset we have 16038 statements and from among them, 12418 are bridges. The percentage of the statements which are also bridges in the dataset is 0.7742861.
b) In the dataset we have 16038 statements and from them we find the number of bridges, without computing the bridges that point to a literal is 11735. The percentage of the statements which are also bridges without pointing to a literal in the dataset is 0.7316997.
c) In the dataset that we created with the statements which include a predicate from the first list, we have 9898 statements and from them 9198 are bridges. The percentage of the statements which are also bridges in the dataset is 0.9292787.

**Reachability:**

a) The nodes that we can reach from "http://dbpedia.org/resource/ Brazil" node as a start are 8.
b) The number of nodes that we can reach from "http://dbpedia.org/resource/ Brazil" node as a start and pass only through predicates which are in the first list that we created is 1.

**Mean rechability:**

a) The mean reachability for all the nodes in the dataset is 257.15933
b) The mean reachability, for all the nodes in the new dataset that we created from statements which have as predicate one, from the first list is 4.7225914.

# Conclusions

In this paper, we have presented an API, namely a set of computer algorithms, for extracting implicit information from Linked Datasets. This API is inspired by the foundations concerning the study of graphs as a representation of relationships between discrete objects, and includes methods for identifying the most prestigious nodes, the key predicates, the link cycles or the reachability for the nodes among other many statistics. The results have shown us that this set of algorithms is very useful to extract implicit information from data which is offered currently in the Web of Data.

As a future line of research, we propose to extend our API with a richer set of functions. The main idea is to include not only basic operations from network theory, but other more complex functionalities. For example, we wish to add a function to compute the similarity between two paths from two different Linked Datasets or to predict new links between nodes. For this reason, we keep working towards a second and improved version of TheMa.

# References

[1] S. Auer, J. Lehmann (2010), Creating knowledge out of interlinked data, Semantic Web, Vol. 1, pp. 97-104.

[2] W. Baker and R Faulkner (1993) The social organization of conspiracy: illegal networks in the heavy electrical equipment industry. Am. Social. Rev., Vol. 58, pp. 837-860.

[3] C. Bizer (2009), The Emerging Web of Linked Data, IEEE Intelligent Systems, Vol. 24(5), pp. 87-92.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann (2009), DBpedia - A crystallization point for the Web of Data, J. Web Sem., Vol. 7(3), pp. 154-165.

[5] D. Chakrabarti, C. Faloutsos (2006), Graph Mining: Laws, Generators, and Algorithms, ACM Comput. Surv., Vol. 38(1).

[6] L. Getoor (2003), Link mining: A new data mining challenge, SIGKDD Exploration, Vol. 5.

[7] D. Gibson, Jon M. Kleinberg, and P. Raghavan (1998) Inferring Web Communities from Link Topology, Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, pp. 225-234.

[8] J. Hopcroft, O. Khan, and B. Selman (2003), Tracking evolving communities in large linked networks, Proceedings of the SIGKDD International Conference on Knowledge Discovery and Data Mining, Vol. 101, pp. 5249-5253.

[9] Jorge Martinez-Gil, Jose Francisco Aldana-Montes: Evaluation of two heuristic approaches to solve the ontology meta-matching problem. Knowl. Inf. Syst. 26(2): 225-247 (2011).

[10] Jorge Martinez-Gil, Jose Francisco Aldana-Montes: Optimizing Ontology Alignments by Using Genetic Algorithms. NatuReS 2008.

[11] H. White and K. McCain (1989) Bibliometrics. Annual Review of Information Science and Technology, Vol. 24, pp. 119-186.