Walking Linked Data: a graph traversal approach to explain clusters

Ilaria Tiddi, Mathieu d'Aquin, Enrico Motta

Knowledge Media Institute
The Open University, United Kingdom
{ilaria.tiddi, mathieu.daquin, enrico.motta}@open.ac.uk

Abstract. Link traversal is one of the biggest advantages of Linked Data, as it allows the serendipitous discovery of new knowledge thanks to the natural connections between data of different sources. Our general problem is to understand how such a property can benefit the Knowledge Discovery process: in particular, we aim at using Linked Data to explain the patterns of data that have been extracted from a typical data mining process such as clustering. The strategy we propose here is Linked Data traversal, in which we explore and build on-the-fly an unknown Linked Data graph by simply deferencing entities' URIs until we find, by following the links between entities, a valid explanation to our clusters. The experiments section gives an insight into the performance of such an approach, in terms of time and scalability, and show how the links easily gather knowledge from different data sources.

Keywords: Linked Data, Graph Traversal, URI Dereferencing

1 Introduction

Almost ten years passed since Tim Berners-Lee presented the Linked Data principles for the first time[1]:

1. Use URIs to denote things.
2. Use HTTP URIs so that these things can be referred to and looked up ("*dereferenced*") by people and user agents.
3. Provide useful information about the thing when its URI is dereferenced, leveraging standards such as RDF and SPARQL.
4. Include links to other related things (using their URIs) when publishing data on the Web.

Ever since, there has been much effort from both the academia and the industry to create a multi-domain, shared knowledge graph today defined as "the Web of Data" (sometimes referred to as the Linked Data Cloud, too). Following those principles, datasets of multiple formats, sources and domains have been published and connected, in order to aggregate fragmentary information into a more complete one and facilitate automatic data reuse.

[1] http://www.w3.org/DesignIssues/LinkedData.html

Interlinking data allows the Linked Data graph to be blindly navigated, as one would usually do with the Web of documents: “blindly”, because by looking up URIs, new resources can be discovered on-the-fly, possibly belonging to unknown datasources, and therefore new knowledge can be serendipitously discovered. If it is true that new fields have emerged in the Semantic Web area, that try to leverage this link traversal feature as well as datasources interconnections, most of their applications still rely on data known in advance. They lose, therefore, one of the major benefits of Linked Data: the serendipitous discovery of knowledge that, in real world applications, is yet to be reached.

Our research finds its place at the intersection between Knowledge Discovery and Linked Data or, in other words, we consider that Linked Data can benefit a field of long tradition such as Knowledge Discovery. What we aim at exploiting is the Linked Data shared knowledge, to derive explanations about Knowledge Discovery patterns (more precisely, clusters). The main assumption is that items are clustered together because of common characteristics, that can be explained by (possibly cross-domain) background knowledge, that is usually provided by experts that analyse and understand those patterns. Assuming those items are represented as Linked Data, we can then exploit this interconnected knowledge to derive explanations about their grouping, by looking for Linked Data information that such items have in common. To this end, can the link traversal be beneficial to derive those explanations, and how?

Based on the previous work presented in [13], we propose in this paper an A* process to derive Linked Data-based explanations for groups of items behaving in the same way. To produce those explanations, we apply a graph search process relying on link traversal and resources dereferencing. Link traversal allows us to navigate and span from datasource to datasource throughout Linked Data, without knowing those in advance nor in their entirety, with the ultimate scope of finding commonalities among the items of the cluster we want to explain. The main contributions of this paper are a reformulation of the process in [13] as an A* strategy based on Linked Data traversal, the extension of the existing process to generate explanations out of datatype (and mostly numerical) properties and a real world use-case in which we demonstrate that by following the links between data we can gather new unrevealed knowledge from different datasources.

2 Problem definition

The scenario we use to illustrate our problem involves the educational domain. The map of Figure 1 shows a dataset $\mathcal{D} = \{c_1, \dots, c_j\}$ of j world countries grouped according the rate of female and male literacy over the last decade (enrolment in secondary and tertiary school from the UNESCO Linked Data statistics[2]). Countries where female are more educated than men are in blue (we will define it as cluster $\mathcal{B} = \{c_i, \dots c_m\}$, where $\mathcal{B} \subset \mathcal{D}$); countries where men are more educated than women in yellow (cluster $\mathcal{Y} \subset \mathcal{D}$); finally, countries where the education rate is on average equal are in green (cluster $\mathcal{G} = \mathcal{D} \setminus \mathcal{B} \cup \mathcal{Y}$).

[2] http://uis.270a.info/.html

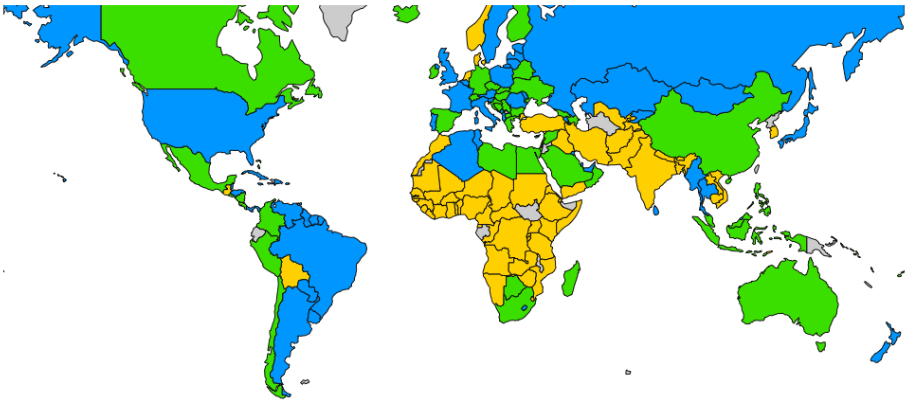

Fig. 1. World countries grouped by literacy rate.

Explaining a cluster. In our example, countries are grouped together if they have a common characteristic, that is, based on the difference between women's literacy rate and the men's one. For each country $c_i \in \mathcal{D}$, we state that:

```
if $literacy(\text{male}, c_i) - literacy(\text{female}, c_i) > 2\%$: then $c_i \in \mathcal{Y}$
else if $literacy(\text{male}, c_i) - literacy(\text{female}, c_i) < 2\%$: then $c_i \in \mathcal{B}$
else: $c_i \in \mathcal{G}$
```

Our first assumption is that countries do not happen to be together by pure luck, but an underlying reason will make them appearing in the same group $\mathcal{C}_i$. Finding this underlying reason is defined as `explain`$(\mathcal{C}_i)$. If one looks at the map, this underlying reason will be clearly visible. In fact,

`explain`$(\mathcal{Y})$ = "*least developed countries*"
`explain`$(\mathcal{B})$ = "*developed countries*"

What one does to deduce so is using his own background knowledge (knowledge about the countries' geopolitical, economical or social situations) to infer that the countries belonging to $\mathcal{Y}$ correspond to societies living on older standards, where women are less educated as their education is not considered useful.

Here, the challenge is, can we exploit Linked Data as the source of such background knowledge, and automatically reproduce the process of explaining a cluster, e.g. `explain`$(\mathcal{Y})$?

Extracting an explanation from Linked Data. Our second assumption is that Linked Data connect enough knowledge to derive the explanation for the items in a cluster, e.g. that countries with less educated women are the least developed countries. This, of course, assumes that such an information is somehow described in some (accessible) Linked Data sources.

The main idea is that the items share in the Linked Data graph the same path, or *walk*, to a specific and unique entity e_i. This walk has length l, corresponding to the distance in number of RDF properties between the observed items we want to explain, and the given entity e_i.

In summary, given:

- a RDF graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ is the set of URI entities and $\mathcal{E}$ the set of RDF properties;
- the set of items $\mathcal{D}$, where $\mathcal{D} \subseteq \mathcal{V}$;
- the cluster we want to explain, $\mathcal{C}^+$, where $\mathcal{C}^+ \subseteq \mathcal{D}$;
- the items that do not belong to $\mathcal{C}^-$, where $\mathcal{C}^- = \mathcal{D} \setminus \mathcal{C}^+$;

there exists

- a set of items $\mathcal{I} = \{c_1, \ldots, c_k\} \subseteq \mathcal{D}$ sharing the same walk $\overrightarrow{w_i}$ of length l to an entity e_i, where $\overrightarrow{w_i}$ is a sequence of l RDF properties $p_i \in \mathcal{E}$ in the form of $\overrightarrow{w_i} = \{p_1, \ldots, p_l\}$ and e_i is an entity in $\mathcal{V}$.

Given the items $c_i \in \mathcal{I}$, some of them would belong to $\mathcal{C}^+$, and some others will belong to $\mathcal{C}^-$. The objective is then to find the best walk $\overrightarrow{w_i}$ to an entity e_i maximising the number of $c_i \in (\mathcal{I} \cup \mathcal{C}^+)$ and minimising the number of $c_i \in (\mathcal{I} \cup \mathcal{C}^-)$. This can be defined as an explanation exp_i for a cluster.

Figure 2 shows a toy example that uses a RDF graph of countries. Here, $\mathcal{D}$ is the set of 5 countries `uis:Somalia`, `uis:Ethiopia`, `uis:India`, `uis:UK` and `uis:US` from the UNESCO dataset. What we know from the clusters is that `uis:Somalia`, `uis:Ethiopia`, `uis:India` belong to $\mathcal{Y}$ ($\mathcal{Y} = \mathcal{C}^+$), while `uis:UK`, `uis:US` to $\mathcal{B}$ ($\mathcal{B} = \mathcal{C}^-$). As one can see, the three `uis:Somalia`, `uis:Ethiopia`, `uis:India` are connected to the DBpedia entity $e_1 =$ `dbpedia:category:Least_developed_Countries` by a walk of length $l = 3$, i.e. $\overrightarrow{w_1}$=$\{$*owl:sameAs*, *dc:subject*, *skos:relatedMatch*$\}$, while `uis:UK`, `uis:US` do share the same $\overrightarrow{w_1}$, but to a different entity $e_2 =$ `dbpedia:category:developed_Countries`. Because items in $\mathcal{Y}$ share the same walk $\overrightarrow{w_i}$ to the entity e_i, while items outside the cluster do not, then this can considered an explanation to it, i.e. `explain`($\mathcal{Y}$).

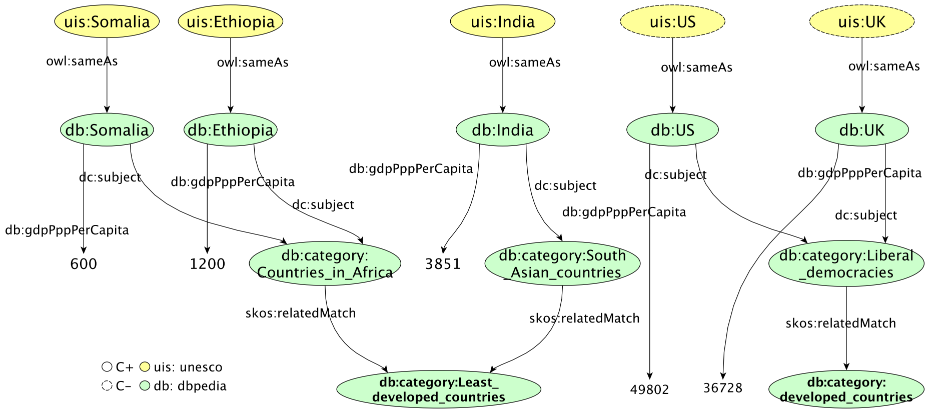

Fig. 2. Linked Data graph about countries.

The process of explaining a cluster is therefore:

$\texttt{explain}(\mathcal{C}^+)$
$exp_1 = \langle \overrightarrow{w_1}.e_1 \rangle$ $\dots$ $exp_k = \langle \overrightarrow{w_k}.e_k \rangle$

finding all the explanations exp_i, with $\overrightarrow{w_i}$ being the common walk and e_i the entity that is common to a set of initial items $\mathcal{I}$, where $|\mathcal{I}| \approx |\mathcal{C}^+|$. In our example,

$\texttt{explain}(\mathcal{Y})$
$exp_1 =$ $\langle$*owl:sameAs*, *dc:subject*, *skos:relatedMatch*. `db:category:Least_developed_Countries`$\rangle$.

Here is the second issue: how to perform such a search for a common entity? In other words, where do we find `db:category:Least_developed_Countries`, and how?

Traversing Linked Data. The interconnection of Linked Data can be easily exploited for this purpose. Looking for a common entity can become a graph search process, in which a graph is iteratively built by traversing entities and following their links to other entities. In such a manner, there is no need to have any a priori knowledge about data sources, nor taking care of data indexing or crawling. Each entity can be dereferenced in order to find connections to other entities (therefore, datasets), allowing the discovery of new knowledge, until an entity common to enough items of the cluster is found.

The link traversal process relies on the fact that if data are connected (through *owl:sameAs*, *skos:exactMatch*, *rdfs:seeAlso* or simply by vocabulary reuse), then we can easily and naturally span datasources and gather new, unknown knowledge. If we refer again to our example, the UNESCO data (defined by the *uis* namespace) are connected to their DBpedia correspondent via the walk $\overrightarrow{w_1}$=$\{$*owl:sameAs*$\}$ of length $l = 1$. So, in only one traversal, we already accessed knowledge within a new datasource. As DBpedia entities are also linked to other datasets, we can expect to go across new datasets within few traversals.

As the link traversal can be only be applied to URIs, our last challenge is: how can we build explanations out of literals and numerical values?

Reasoning over datatype properties. So far we have considered as valid explanation for a group of items $\mathcal{I}$ a walk $\overrightarrow{w_i}$ from them to one common entity e_i. If we look again at our graph example, we will notice that `uis:Somalia`, `uis:Ethiopia`, and `uis:India` have the same walk $\overrightarrow{w_2} = \{$*owl:sameAs*, *dbp:gdpPppPerCapita*$\}$, and the three numerical values they are walking to are similar if compared to the ones of items in cluster $\mathcal{B}$. Again, our human expert would say:

$$\texttt{explain}(\mathcal{Y}) = \text{“\textit{countries with a GPD per capita lower than 4k\$}”}$$

In the case of incomes, it is unlikely that two countries will have the same one, so we cannot expect that the walk will take to a common value. It is necessary to refine the definition of an explanation for a cluster, by including this similarity between numerical values, as well as literals:

1. `explain`($\mathcal{C}_i$): $\langle \overrightarrow{w_i}.v_i \rangle$
   if the last property p_l of the walk $\overrightarrow{w_i}$ is an object property
2. `explain`($\mathcal{C}_i$) $=\langle \overrightarrow{w_i}.\ [\leq \mid \geq].v_i \rangle$
   if the last property p_l of the walk $\overrightarrow{w_i}$ is a datatype property

To conclude, we now focus on creating a process to generate those explanations, that exploits the Linked Data traversal and interconnections between datasources.

3 Proposed Solution

3.1 Dedalo, an A* process for Linked Data

In [13] we presented Dedalo, an automatic approach to derive Linked Data explanations out of clusters. As said, the current work presents an extension of such a process.

Dedalo is an A* process considering Linked Data as a graph in which nodes are the RDF entities and edges are the properties connecting them. Many algorithms have proven being more efficient than the A* in pathfinding, as they pre-process the graph to perform better. Those approaches, however, cannot be applied in our context, for two main reasons: (1) a retrieval of the entire Linked Data graph is not conceivable considering the huge amount of data sources and (2) most of the information would actually be not relevant for our explanation (we might not care about movies, when looking for an explanation about countries, unless those movies are connected to the countries for some reason).

The A* is a best-first search aiming at finding the least-cost path from a given initial node (the source) to one other node (the goal) according to a given heuristics [3]. The graph traversal is held by following the path with the lowest cost, while the new paths are collected and kept into a queue. The cost of a path x is estimated using a heuristic measure $f(x)$, which defines the order the paths in the queue. $f(x)$ is the sum of :

- $g(x)$, the past path-cost function, which is the known distance from the starting node to the current node;
- $h(x)$, the future path-cost function, which is an estimate of how likely the path is to be a good one to reach the goal.

This idea is then applied to Linked Data. Items in $\mathcal{D} = \{c_1, \ldots, c_j\}$ are the graph sources, while the entity e_i of each explanation $exp_i = \langle \overrightarrow{w_i}.e_i \rangle$ is the goal. In [13], we demonstrated how the *entropy* of a path is a valid cost function $f(x)$ for our purpose. Entropy [12] focuses on the frequency of a given path (corresponding to $g(x)$) and the distribution of its values (corresponding to $h(x)$). For a detailed discussion around other possible cost functions, please refer to [13].

The problem here is that we do not know what is the goal in advance, nor we can know how good it is for our cluster. Moreover, our graph is build iteratively: each time we dereference new entities, $\mathcal{V}$ increases in size. For this reason, the goal of our traversal is any entity $e_i \in \mathcal{V}$ at a maximum distance j from the sources, where j is the length of the graph at the jth given iteration. Iteration is intended as how many times a new (first) path is the queue has been chosen. When this happens, a new part of the graph $\mathcal{G}$ is revealed, and new goals e_i are added to $\mathcal{V}$. Finally, for each of the discovered goals, we introduced a second function $f^2(exp_i)$, to assess the explanation $exp_i = \langle \overrightarrow{w_i}.e_i \rangle$ for the given cluster.

3.2 The Linked Data traversal process

The Linked Data traversal is composed of three different steps: (i) URI dereferencing, (ii) Path collecting and (iii) Explanation building.

URI dereferencing. Initially, the graph we have is a graph of length $j = 0$, where $\mathcal{V} = \mathcal{D}$ and $\mathcal{E} = \emptyset$. As explained, we chose to use the URI dereferencing process to be consistent with the Linked Data principles. For each of the items, we use the HTTP protocol to obtain all the RDF properties and values the entity is related to, by collecting all the triples $<e_i, p_i, v_i>$. For example, given the entity `uis:Ethiopia`, we collect p_0=`owl:sameAs` and v_0=`dbpedia:Ethiopia`. The discovered values v_i are added to $\mathcal{V}$, while the properties to $\mathcal{E}$. As one can see, some of the discovered values are part of new datasets, that we have found following the natural links of the described resource. In case the entity has no equivalent values, we select equivalent instances using the sameAs.org service[3], by processing the new triples $<e_i$,`owl:sameAs`,$v_i>$ and adding its components to the graph.

Path collecting. Each new walk $\overrightarrow{w_i}$ is built starting by adding to the existing first walk of the pile, the new properties p_i of each triple extracted from the URI dereferencing. The new $\overrightarrow{w_i}$ are evaluated according to the entropy function $ent(\overrightarrow{w_i})$ and queued in the pile of possible walks to follow in the graph accordingly. When the new first walk in the queue will be chosen, a new $j+1$th iteration will start.

For instance, if the last first walk in the pile was of length $l = 1$ such as $\overrightarrow{w_1}$ =$\{owl{:}sameAs\}$ and from the dereferencing of the entity `dbpedia:Ethiopia` we have collected the triples:

t_1=<`dbpedia:Ethiopia,dc:subject,db:category:Countries_in_Africa`>
t_2=<`dbpedia:Ethiopia,dbp:gdpPppPerCapita,"1200"`>

we will form two new walks of length 2, such as $\overrightarrow{w_2}$=$\{owl{:}sameAs,\ dc{:}subject\}$ and $\overrightarrow{w_3}$=$\{owl{:}sameAs,\ dbp{:}gdpPppPerCapita\}$. We then evaluate their costs with $ent(\overrightarrow{w_2})$ and $ent(\overrightarrow{w_3})$ and add them to the queue of paths to follow.

[3] http://sameas.org/

All the entities, the first $\overrightarrow{w_i}$ in the queue walks to, are the ones further expanded by deferencing within the following iteration. If we assume the walk with the least cost is $\overrightarrow{w_2}$, all the entities this one takes to (in our case `db:category:Countries_in_Africa`, `db:category:South_Asian_countries`, and `db:category:Liberal_democracies`) are dereferenced. Subsequently, new walks are found and build of out of this new traversal, e.g. $\overrightarrow{w_4} = \{$*owl:sameAs, dc:subject, skos:relatedMatch*$\}$, and so on.

Explanations building. Before starting a new iteration, we build and evaluate the new explanations. Explanations are built by chaining the walks $\overrightarrow{w_i}$ to the entities $\overrightarrow{e}_i$ that have been discovered at the current iteration. The length of the new explanations, which corresponds to the length of the walk $\overrightarrow{w_i}$ first in the queue, gives an insight of how much the graph has been traversed, i.e. how far we have gone from the sources. If we take $\overrightarrow{w_4}$, we will build the following explanations:

$exp_1 = \langle$ *owl:sameAs,dc:subject,skos:relatedTerm*.`db:category:developed_countries` $\rangle$
$exp_2 = \langle$ *owl:sameAs,dc:subject,skos:relatedTerm*.`db:category:Least_developed_countries`$\rangle$

To evaluate how accurate a new explanation exp_i is for the cluster we are explaining, we chose as $f^2(exp_i)$ the F-Measure $= 2 * \frac{P*R}{P+R}$, and adapted it by defining precision and recall as follows. Given an explanation $exp_i = \langle \overrightarrow{w_i}.e_i \rangle$:

$$P = \frac{sources(exp_i) \cap \mathcal{C}^+}{sources(exp_i)} \quad (1) \qquad R = \frac{sources(exp_i) \cap \mathcal{C}^+}{|\mathcal{C}^+|} \quad (2)$$

where $sources(exp_i)$ is equivalent to $|\mathcal{I}|$, the number of sources walking to e_i through the walk $\overrightarrow{w_i}$, and $\mathcal{C}^+$ is the cluster we want to explain. For instance, the explanation exp_2 has three sources walking to it, and the three of them are part of $\mathcal{C}^+$ $(= \mathcal{Y})$, while none from outside the cluster is. So we consider it as the most valuable explanation for the cluster.

In the case the walk $\overrightarrow{w_i}$'s ending property p_l is a datatype property and v_i is a numerical value, we create two alternate explanations:

$$exp_1 = \langle \overrightarrow{w_i}. \geq .v_i \rangle$$
$$exp_2 = \langle \overrightarrow{w_i}. \leq .v_i \rangle$$

and check, for each of the sources that have that same walk $\overrightarrow{w_i}$, whether the value v_j they are walking to is greater or less than the value v_i, and subsequently estimate the F-measure of both exp_1 and exp_2. Let us consider the walk $\overrightarrow{w_2}$ again. The entity `uis:Ethiopia` walks to the value v_1="1200", `uis:Somalia` to the value v_2="600" and `uis:India` to v_3= "3851". For each of the values v_i, we create the two alternate explanations and then evaluate them (see Table 1), keeping only the one with the best score with respect to the cluster $\mathcal{Y}$.

4 Explaining the map – experiments

Data preparation. The UNESCO Institute for Statistics publishes most of its data under Linked Data principles. Following the cube model[4], they pro-

[4] http://www.w3.org/TR/vocab-data-cube/

Table 1. Example of the production of explanations for numeric values.

	exp_i	$f^2(exp_i)$
$e_1 =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≥.600⟩	75%
$e_2 =$	⟨owl:sameAs, dbp:gdpPppPerCapita.≤.600⟩	50%
$e_3 =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≥.1200⟩	57%
$e_4 =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≤.1200⟩	80%
$e_5 =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≥.3851⟩	33%
$e_6 =$	⟨***owl:sameAs, dbp:gdpPppPerCapita.*****≤.3851**⟩	**100%**
$e_7 =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≥.49802⟩	0%
$e_8 =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≤.49802⟩	75%
$e_9 =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≥.36728⟩	0%
$e_{10} =$	⟨*owl:sameAs, dbp:gdpPppPerCapita.*≤.36728⟩	85%

vide statistical observations about countries in a wide range of domains such as economics, food, agriculture, finance and so forth. To select data and build the dataset $\mathcal{D}$ of items to use as source of the graph, we used the provided SPARQL endpoint. We selected, for each country, the percentage of females enrolled in the secondary and tertiary education since the year 2000 and accordingly derived the male one. We thus compared the two percentages: if the absolute difference of the two groups was less than 2%, the country was considered part of the $\mathcal{G}$ cluster, comprehending countries where the education is on average equal. As already presented, the map of Fig. 1 shows the results. All those data, as well as the results and maps, are publicly available online[5].

4.1 Evaluation and discussion

In our experiments we aim at evaluating how fast the process `explain`$(\mathcal{C}^+)$ performs.

We are interested in knowing how much time it takes to reach the same explanation exp_i that a human would naturally give, how much it fits the cluster $\mathcal{C}^+$, how far it is from the sources, as well as how big is the graph at the moment of the discovery. This is a preliminary step for a broader evaluation to be held on a long term perspective, in which we aim at manually evaluating explanations obtained automatically and the ones given by human experts.

Table 2 shows the results we had for each cluster after 10 iterations. Time is evaluated in terms of seconds taken to reach the explanation exp_i; the quality of the explanation for the cluster $\mathcal{C}^+$ is evaluated in F-Measure. In 10 iterations, our graph has 3.742.344 triples, 671 walks $\overrightarrow{w_i}$ have been built and are queueing in the pile.

As one can remark, the process found very good explanations (in F-measure score) with very little cost. Dedalo's A* process is actually able to produce explanations involving knowledge from different datasources (from the UNESCO statistics to DBpedia), by following the natural links between data and by cleverly detecting the correct walk to follow into the big Linked Data graph.

To get the best explanation for $\mathcal{Y}$, the process requires less than 200". The explanation shows that the 87.8% of the countries in $\mathcal{Y}$ are ranked below the 126th

[5] http://linkedu.eu/dedalo/

Table 2. Summary of the best explanations found for each group of countries, the time it has taken to get to exp_i and its F-Measure.

`explain`($\mathcal{Y}$): countries where males are more educated		
exp_i	F(%)	Time”
⟨*skos:exactMatch, dbp:hdiRank.*≥.“126”⟩	87.8	197”
⟨*skos:exactMatch, dc:subject.* `db:Category:Least_developed_countries`⟩	74.7	524”
⟨*skos:exactMatch, dbp:gdpPppPerCapitaRank.*≥.“89”⟩	68.3	269”
⟨*skos:exactMatch, dc:subject skos:broader.* `db:Category:Countries_in_Africa`⟩	67.1	540”
⟨*skos:exactMatch, dbp:populationEstimateRank.*“76”⟩	61.9	201”
⟨*skos:exactMatch, dbp:gdpPppRank.*≥.“10”⟩	59.1	235”

`explain`($\mathcal{B}$): countries where females are more educated		
exp_i	F(%)	Time”
⟨*skos:exactMatch, dbpedia:hdiRank.*≤.“119”⟩	63.4	198”
⟨*skos:exactMatch, dbp:gdpPppRank.*≤.“56”⟩	62.3	236”
⟨*skos:exactMatch, dbp:populationEstimateRank.*≥.“128”⟩	56.9	203”
⟨*skos:exactMatch, dbp:gdpPppPerCapitaRank.*≤.“107”⟩	56.3	267”
⟨*skos:exactMatch, dbp:gdpPppPerCapitaRank.*≥.“100”⟩	54.5	267”
⟨*skos:exactMatch, dc:subject, skos:broader.* `db:Category:Latin_American_Countries`⟩	49.3	542”

`explain`($\mathcal{G}$): countries where education is on average equal		
exp_i	F(%)	Time”
⟨*skos:exactMatch, dbprop:gdpPppRank.*≥.“64”⟩	62	234”
⟨*skos:exactMatch, dbprop:gdpPppPerCapitaRank.*≥.“29”⟩	61	268”
⟨*skos:exactMatch, dbprop:areaRank.*≥.“18”⟩	57	254”
⟨*skos:exactMatch, dbprop:populationDensityRank.*≤.“148”⟩	52	238”
⟨*skos:exactMatch, dbprop:populationEstimateRank.*≥.“25”⟩	49	201”

country in the *Human Development Index*[6] (HDI) ranking. Based on statistics on life expectancy, education and income, the HDI ranks countries from the most developed to the least one. The lower the country is in the rank, the less developed it is. Similarly, the best explanation for $\mathcal{B}$ is that the 63.4% of its countries are among the 119 most developed countries. It is important to recall that such an explanation would have not been found without any reasoning upon numerical values. Other good explanations involve an object property, which confirms our assumption that items of the same cluster share walks to common values. In fact, the second good explanation for $\mathcal{Y}$ is that the 74.7% of the cluster is labeled in DBpedia as least developed countries, which means that they all have a common walk $\vec{w_i}$={*skos:exactMatch, dc:subject*} to the common entity `db:Category:Least_developed_Countries`.

[6] http://en.wikipedia.org/wiki/Human_Development_Index

5 Related Work

Works discovering new knowledge in Linked Data can be grouped into bottom-up and top-down approaches.

Bottom-up approaches are focused on coping with data diversity. Generally, those approaches present data services allowing the exploration, navigation and reasoning on billions of triples from different datasets: among them, we can cite Factforge [1], including DBpedia, Freebase, Geonames, the CIA World Factbook, MusicBrainz, WordNet and the New York Times; the LODatio framework [5], a platform to search instances over Linked Data, using a *Google-like* approach based on RDF types and properties; but also indexes such as the OpenLinks LOD cache[7] or the Semantic Web index Sindice [2]. The main objective of those works is to keep a large, up-to-date coverage of the Web of Data as well as a fast and efficient response time of the service. As already mentioned in [8], those objectives have been partially met using technical expedients (e.g. distribution techniques, index optimisation, data synchronisation), but they still require a local data management that goes beyond the principles of the Web of Data.

The second category comprehends top-down techniques traversing Linked Data as graph and exploiting the connections between sources for an on-the-fly knowledge discovery. Some works such as the ones of [4, 11] focus more on the navigation functionalities providing query languages, while recent approaches to automatically traverse links between entities to gather data live and from unknown sources can also be found in the Link Traversal Based Query Execution field (LTBQE), such as the ones of [6, 7, 14]. After obtaining the query results, the URIs are looked up following the data links in order to improve the SPARQL answer with information from unknown sources. Similarly, we use the entities dereferencing to gather unknown data and produce meaningful explanation for clusters.

Finally, the idea of applying graph search algorithms to Linked Data has been exploited in the literature for users recommendation. In the works of [9, 10] users are suggested items that are considered similar, when similar means Linked Data items sharing the same path to a specified entity. Those work only take into consideration a singular graph (such as DBpedia) and do not consider the knowledge that might be connected in external datasources. Moreover, they rely on SPARQL endpoints to retrieve information rather than URI lookup.

6 Conclusion and Future Work

In this work we presented an extension to Dedalo, a process to explain Knowledge Discovery clusters using Linked Data. To achieve this, we redefined Dedalo as an A* search in the Linked Data graph aiming at finding the best walk(s) of RDF properties between a set of initial sources (the items in the cluster to be explained) and a specific value in the graph, that can be either a URI resource or a numerical value. Those explanations are built using the links between data (and datasources), simply exploiting URI dereferencing. Without having any a priori

[7] http://lod.openlinksw.com/

knowledge about the datasources, we find meaningful explanations gathering knowledge from different datasets.

The future direction we might want to take are currently focusing on the noise and bias introduced by the owl:sameAs links. In fact, explanations might be biased if information in the datasets is missing or not homogeneous. Other future directions might concern the traversal of incoming links, as our process currently only takes into account the outgoing ones.

References

1. Bishop, B., Kiryakov, A., Ognyanov, D., Peikov, I., Tashev, Z., & Velkov, R. (2011). Factforge: A fast track to the web of data. Semantic Web, 2(2), 157-166.
2. Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., & Tummarello, G. (2008). Sindice.com: a document-oriented lookup index for open linked data. IJMSO 3(1), 3752.
3. Delling, D., Sanders, P., Schultes, D., & Wagner, D. (2009). Engineering route planning algorithms. In Algorithmics of large and complex networks (pp. 117-139). Springer Berlin Heidelberg.
4. Fionda, V., Gutierrez, C., & Pirró, G. (2014). The swget portal: Navigating and acting on the web of linked data. Web Semantics: Science, Services and Agents on the World Wide Web, 26, 29-35.
5. Gottron, T., Scherp, A., Krayer, B., & Peters, A. (2012). Get the google feeling: Supporting users in findingrelevant sources of linked open data at web-scale. Semantic Web Challenge, Submission to the Billion Triple Track.
6. Hartig, O., & Langegger, A. (2010). A database perspective on consuming linked data on the web. Datenbank-Spektrum, 10(2), 57-66.
7. Hartig, O. (2013, June). SQUIN: a traversal based query execution system for the web of linked data. In Proceedings of the 2013 international conference on Management of data (pp. 1081-1084). ACM.
8. Ladwig G. & Tran, T. (2011). SIHJoin: Querying remote and local Linked Data. In ESWC 2011.
9. Ostuni, V. C., Di Noia, T., Mirizzi R. & Di Sciascio E. (2014). A Linked Data Recommender System using a Neighborhood-based Graph Kernel. EC-Web2014; to appear.
10. Ostuni, V. C., Di Noia, T., Di Sciascio, E., & Mirizzi, R. (2013, October). Top-N recommendations from implicit feedback leveraging linked open data. In Proceedings of the 7th ACM conference on Recommender systems (pp. 85-92). ACM.
11. Pérez, J., Arenas, M., & Gutierrez, C. (2010). nSPARQL: A navigational language for RDF. Web Semantics: Science, Services and Agents on the World Wide Web, 8(4), 255-270.
12. Shannon, C.(1948).A Mathematical Theory of Communication. Bell System Technical Journal 27 (3): 379-423.
13. Tiddi, I., d'Aquin, M. and Motta, E. (2014) Dedalo: looking for Clusters Explanations in a Labyrinth of Linked Data, 11th Extended Semantic Web Conference, ESWC 2014, Crete.
14. Umbrich, J., Hogan, A., Polleres, A., & Decker, S. (2014). Link Traversal Querying for a diverse Web of Data. Semantic Web Journal.