Dedalo: looking for Clusters Explanations in a Labyrinth of Linked Data

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. We present Dedalo, a framework which is able to exploit Linked Data to generate explanations for clusters. In general, any result of a Knowledge Discovery process, including clusters, is interpreted by human experts who use their background knowledge to explain them. However, for someone without such expert knowledge, those results may be difficult to understand. Obtaining a complete and satisfactory explanation becomes a laborious and time-consuming process, involving expertise in possibly different domains. Having said so, not only does the Web of Data contain vast amounts of such background knowledge, but it also natively connects those domains. While the efforts put in the interpretation process can be reduced with the support of Linked Data, how to automatically access the right piece of knowledge in such a big space remains an issue. Dedalo is a framework that dynamically traverses Linked Data to find commonalities that form explanations for items of a cluster. We have developed different strategies (or heuristics) to guide this traversal, reducing the time to get the best explanation. In our experiments, we compare those strategies and demonstrate that Dedalo finds relevant and sophisticated Linked Data explanations from different areas.

Keywords: #eswc2014Tiddi, Linked Data, Hypothesis Generation, Knowledge Discovery

1 Introduction

When running a Knowledge Discovery (KD) process, the last step usually consists in interpreting the results (sometimes called "patterns") that have been extracted from data during the data mining step. In most real-world scenarios, those results are given to experts that, with their background knowledge, analyse and give them their own interpretation. However, if given to someone without such expertise, the results would hardly be understandable. Also, additional knowledge from domains that the expert might not be aware of could affect the quality of the interpretation. This makes the interpretation process laborious, manual and time-consuming.

With that said, the Web of Data links datasets of different areas using RDF standards, making sources of knowledge accessible (and interpretable) by machines. With the amount of information shared through Linked Data, it should

therefore be possible to find common characteristics (properties) of the items of a cluster that significantly distinguish them from others, therefore forming hypotheses for the explanation of their grouping. In this scenario, it is clear that the major issue consists of deciding which is the correct piece of knowledge to look at first, in order to quickly find a plausible explanation and not get lost in the Linked Data web.

One of our use-cases consists of coherent clusters obtained through applying Network Partitioning to the co-authorship graph of academic researchers of the same department. While someone familiar with such a department, given those clusters, would explain them saying that each cluster corresponds to a group working on similar topics, someone without such knowledge would find the clusters meaningless. One might require even more background knowledge to state that researchers of the same cluster have worked on projects led by the same person. Our assumption is that, Linked Data can be used to give such explanations. In this scenario, the major issue is to access the right information (research topics of academics, project memberships, etc.) in the Linked Data cloud, assuming of course that such knowledge is herein represented, to find relevant explanations in a short time.

In this paper, we present Dedalo, a framework that, based on a subfield of Machine Learning (Inductive Logic Programming [15]), automatically provides explanations for clusters using Linked Data. When given a set of clusters, Dedalo traverses Linked Data in order to find the best explanation. We used different strategies (or heuristic scoring measures of the properties to inspect) to guide this traversal and in the experiments section we present an evaluation of their performance.

2 Foundations and Related Work

Hypothesis Generation. Hypothesis generation is defined as "*the pre-decisional process by which it is possible to formulate explanations and beliefs regarding the occurrences observed in a specific environment*" [20]. Systems presented in the literature can be classified according to different dimensions: (i) manual or automatic, (ii) domain-specific or domain-independent and (iii) ontology- or Linked Data-driven. In the past, ontologies revealed their usefulness for automatically generating hypotheses; however, this has been mostly shown in specific contexts, e.g. medical computer science or biology, where systems such as Adam, Eira or HyBrow [7, 12, 19] have used OWL reasoning and first-order logic to automatically derive explanations. Hypothesis generation is also the last step of the KD process (sometimes called "data post-processing"), where results obtained from the data mining step are interpreted and refined to start a new iteration on the data. Attempts at using ontologies to produce explanations for KD results (clusters or association rules) can be found in [1, 8, 11]. While [1] describes a domain-specific but automatic process, [8, 11] are domain-independent but require the experts to manually validate the generated hypotheses.

In this context, our first challenge is to produce an automatic, domain-independent process to generate hypotheses.

Assumption 1. *Given a set of clusters* $\mathcal{C}=\{\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_n\}$ *extracted from a set of items* $\mathcal{R}=\{r_0, ..., r_m\}$ *(where* $\mathcal{C}_i \subseteq \mathcal{R}$*), there exists a set of explanations* $\mathcal{H}_i = \{h_0, ..., h_j\}$ *coming from some background knowledge* $\mathcal{B}$ *for each* $\mathcal{C}_i \in \mathcal{C}$.

Linked Data for Hypothesis Generation. The potential of Linked Data for accessing cross-disciplinary linked knowledge is shown in research which uses them to generate hypotheses starting from semi-structured data (such as web tables or statistics) [16, 17, 24]. The importance of selecting the correct background knowledge in order to reduce the computational efforts required by the process emerges from that research line. In the KD field, frameworks for analysing data mining results still select the background knowledge from Linked Data manually, such as in [2, 3, 18, 23]. Following this research line, our second challenge is to automatically select the background knowledge from Linked Data to produce explanations.

Assumption 2. *Given a cluster* $\mathcal{C}_i \in \mathcal{C}$*, Linked Data contains enough connected knowledge to produce a set of explanations* $\mathcal{H}_i$ *for each* $\mathcal{C}_i \in \mathcal{C}$.

Automatic Hypothesis Generation. The automatic generation of hypotheses has been investigated in a field at the intersection of Machine Learning and Logic Programming, called Inductive Logic Programming (ILP, which first appeared in [15]). ILP constructs first-order logic clausal theories (as in Logic Programming) starting from a set of positive and negative examples (as in Machine Learning). To derive those theories, or hypotheses, ILP applies reasoning upon some background knowledge about the examples (both positive and negative).

For example, imagine we want to automatically learn "why someone attends ESWC": `attendsESWC(X)`. In Table 1, the examples show who is participating in ESWC (e^+), and who is not (e^-). In the background knowledge, some more

Table 1: An example of the ILP framework.

	Examples	Background Knowledge
e^+	`attendsESWC(MathieuDAquin).`	`submitted(MathieuDAquin).`
e^+	`attendsESWC(VanessaLopez).`	`submitted(EnricoMotta).`
e^-	`attendsESWC(EnricoMotta).`	`submitted(VanessaLopez).`
		`acceptedPaper(MathieuDAquin,'ESWC').`
		`acceptedPaper(VanessaLopez,'ESWC').`

information about those examples is given. While one can see that all the examples submitted a paper to ESWC, only two of them had their paper accepted.

So, in order to go to ESWC, a person will have to have submitted a paper but also have it accepted:

```
goesToESWC(X) <- submitted(X)∧paperAccepted(X,'ESWC')
```

Lately, ontologies have attracted the interests of several researchers in this area, as they see the formalised knowledge of ontologies as a possible support to build the background knowledge for ILP. A survey of systems exploiting ontologies in ILP is presented in [10]. Similarly, other works have combined Logic Programming and ontologies in the field of Description Logic Programming [6, 13] and Onto-Relational Learning [9].

While no work (other than our first attempt reported in [22]) in the ILP field seems so far to have taken into consideration the Linked Data potential, we consider Linked Data a promising set of resources to help the automatic building of the ILP background knowledge. Given our second assumption, our third challenge is to automatically build the background knowledge of an ILP process using Linked Data.

Assumption 3. *Given a set* $\mathcal{C}^+$ *of positive examples* (*where* $\mathcal{C}^+ \subseteq \mathcal{R}$ and $\mathcal{C}^+ \in \mathcal{C}$) *which we want to find explanations for, a set of negative examples* (*the remaining clusters of* $\mathcal{C}$: $\mathcal{C}^- = \mathcal{R} \setminus \mathcal{C}^+$), *we can use Linked Data as background knowledge* $\mathcal{B}$ *to find explanations about* $\mathcal{C}^+$.

However, using the full Linked Data graph as background knowledge in an ILP process is obviously unfeasible because of the time and computational costs it would imply, while most of this knowledge would certainly be irrelevant. It is then necessary to detect and select only the salient information for our background knowledge. Hence, in our ILP-based framework, we have to focus on finding a clever heuristic to guide the traversal of Linked Data and select relevant background knowledge for generating explanations of the cluster in hand.

3 Dedalo's Framework

Dedalo is conceived as a graph-search process. Here, Linked Data are considered a graph of resources and properties (respectively nodes and edges) and traversed to collect candidate hypotheses about the items $r_i \in \mathcal{R}$, that are used as roots of the graph. Our intuition is that, given a subset of items $r_i \in \mathcal{R}$, there are *paths* (i.e. chains of property assertions) and an end value they have in common: this is how we define a *hypothesis*. We can then assume that when items in the same cluster $\mathcal{C}_i$ share a hypothesis more commonly than items outside the cluster, then that hypothesis constitutes an explanation for $\mathcal{C}_i$.

Path. *A chain of RDF properties defined as* $\mathfrak{p} = \langle p_0 \rightarrow p_1 \rightarrow ... \rightarrow p_n \rangle$.
Hypothesis. *A path* $\mathfrak{p}$ *and an end value* v_i, *defined as* $h_i = \langle \mathfrak{p}.v_i \rangle$.

As our objective is to find the best hypotheses, the graph needs to be iteratively traversed. A complete iteration consists of (see Fig. 1 for an overview):
1. URI Expansion, to resolve a Linked Data entity;
2. Path Extraction, to know which path of the graph leads to a given entity;
3. Path Ranking, to choose the best path to use in the following iteration;
4. Path Values Selection, to select the values of a path that will be further explored;
5. Hypotheses Evaluation, to extract and rank the hypotheses found at the current iteration.

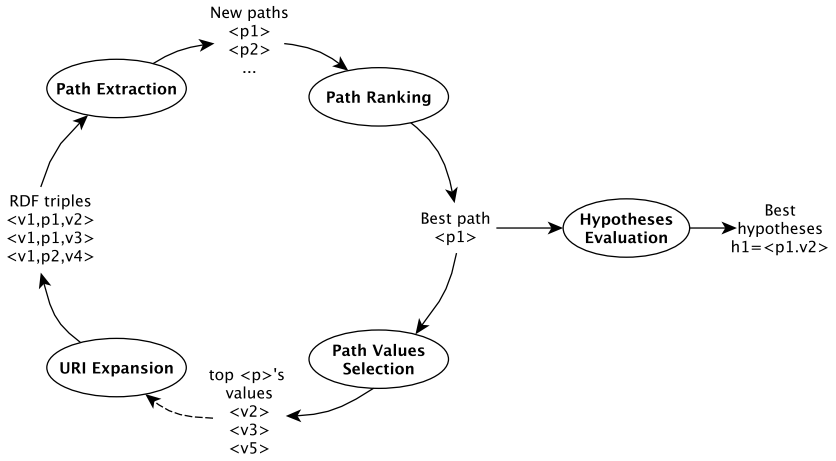


Fig. 1: Overview of Dedalo's structure.

Within a new iteration, two scenarios are possible: (i) we find a better hypothesis, which explains more items $r_i \in \mathcal{C}^+$ than the previous one, or (ii) no better hypotheses are found, and therefore we still consider the current hypothesis as the best one. In other words, by augmenting the time of the traversal, results can only increase in quality. Therefore, Dedalo can be considered an **anytime process**.

As already introduced in the previous section, it becomes clear that, being limited in time and computational resources, a complete graph traversal is not conceivable. Moreover, in an ample space such as Linked Data connected through the `<owl:sameAs>` predicates, the number of paths to follow increases exponentially. This is why we developed several heuristics in order to find the one that was able to predict the most promising path to follow, optimising the process of quickly finding the best hypotheses.

3.1 Algorithm

This section presents the components of Dedalo. For a better understanding of our framework, we will use the graph given in Fig. 2 as an example.
1 – URI Expansion. Given a resource, we resolve its URIs and collect all the property-value pairs $\langle p_i.v_i \rangle$ from the RDF entity description. For instance, we resolve the resource `<ou:MathieuDAquin>` and extract the couples `<ou:memberOf.ou:KMI>` and `<ou:participatedTo.ou:Watson>`.

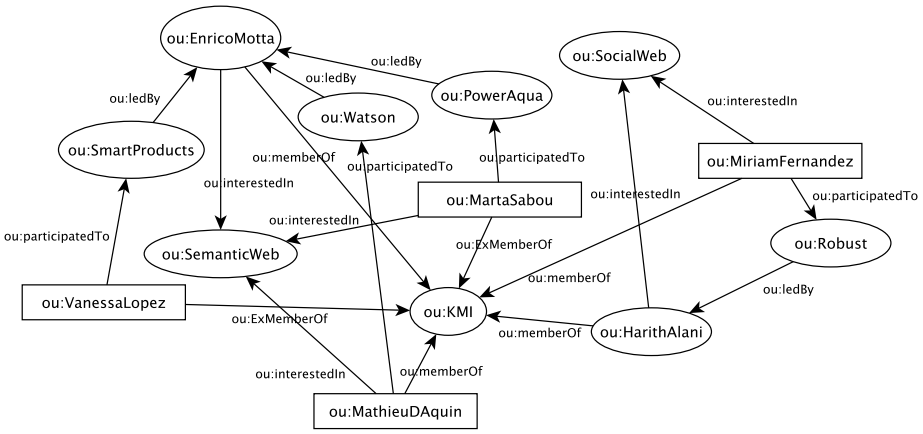


Fig. 2: Graph example using a group of academic researchers. Items in rectangles are the roots of the graph: $r_i \in \mathcal{R}$.

2 – Path Extraction. We detect which is the path that has led us to a given resource (which means, detecting which depth of the graph we have reached). As we already defined a path as a sequence of properties, $\mathfrak{p}=\langle p_0 \rightarrow p_1 \rightarrow ... \rightarrow p_n \rangle$, in the case we reached the resource `<ou:EnricoMotta>`, $\mathfrak{p}=\langle$ou:participatedIn $\rightarrow$ ou:ledBy$\rangle$. A path $\mathfrak{p}$ has also the following properties:

$||\mathfrak{p}||$ – the number of properties composing it, showing how deep we descended in the graph. In the current example, $||\mathfrak{p}||=2$.

***roots*($\mathfrak{p}$)** – the set of roots that share this same path. As the three root items `<ou:MathieuDAquin>`, `<ou:MartaSabou>` and `<ou:VanessaLopez>` are all followed by $\mathfrak{p}$, $|roots(\mathfrak{p})|=3$.

***vals*($\mathfrak{p}$)** – the set of ending values the path can have. In the current example, $|vals(\mathfrak{p})|=2$ because both `<ou:EnricoMotta>` and `<ou:HarithAlani>` are ending values of $\mathfrak{p}$.

Each detected $\mathfrak{p}$ is added to the list of paths to be ranked further ($add(\mathfrak{p}, paths)$).

3 – Path Ranking. To deepen the graph exploration to collect new hypotheses, we need to choose the best path to follow before starting a new iteration. The set of paths discovered at that moment are therefore ranked according to one of the strategies we have defined (presented in the next subsection). In our example, we will have to establish whether we want to follow $\mathfrak{p}_1=\langle$ou:memberOf$\rangle$ or $\mathfrak{p}_2 = \langle$ou:participatedIn$\rightarrow$ou:ledBy$\rangle$.

4 – Paths Values Selection. Once the best path is chosen, its values $vals(\mathfrak{p})$ are expanded (as in step 1), and new (longer) paths are collected (as in step 2). If we chose to follow $\mathfrak{p}_2$, the next values to expand will be `<ou:EnricoMotta>` and `<ou:HarithAlani>`. By iteratively expanding those values and collecting new paths, we deepen the search in the Linked Data graph: this process is defined as "the Linked Data traversal", and it is detailed in the function in Algorithm 1.

Algorithm 1 Linked Data traversal

```
function traverseLinkedData(uris)
    for uri in uris do
        newValues ← expandURI(uri)                    ▷ step 1
    end for
    for value in newValues do
        newPath ← extractPath(value)                  ▷ step 2
        if newPath not in paths then
            add(newPath, paths)
        end if
    end for
end function
```

5 – Hypotheses Evaluation. This step is composed of two parts. At a first stage (called $hypos(\mathfrak{p})$), given the best $\mathfrak{p}$, we extract the hypotheses we can derive from it, by chaining it to its end values $vals(\mathfrak{p})$. In a second phase ($evaluate(h_i)$), each hypothesis in $hypos(\mathfrak{p})$ is evaluated and associated to a score. The score is based on the number of root items $r_i \in \mathcal{C}^+$ sharing the given hypothesis h_i (i.e., the path $\mathfrak{p}$ chained to one of its end values v_i, or $\langle \mathfrak{p}.v_i \rangle$). The best scored will be the best hypothesis $top(\mathcal{H})$ of the current iteration.

The score is calculated according to the hypothesis evaluation measure. The literature includes a wide range of rule evaluation measures [5]. Since the scope of our work is to find the best strategy to traverse Linked Data and get the best hypotheses, we briefly explored them and decided to use the Weighted Relative Accuracy (WR_{acc}). A more complete assessment of evaluation measures is planned for future work. WR_{acc} is part of the probability-based rules classification measures, commonly used to establish the statistical significance of an explanation. It takes into account both the generality (i.e. how big is the portion of $\mathcal{C}^+$ that the hypothesis is matching, compared to the whole dataset) and the reliability (how frequent is the hypothesis in the whole dataset) of a rule. The generality of a hypothesis h_i is defined by the number of roots $r_i \in \mathcal{R}$ matched by h_i, while its reliability is defined in terms of how much h_i matches elements from the cluster $\mathcal{C}^+$ compared to the size of $\mathcal{R}$. WR_{acc} is therefore defined as follows:

$$WR_{acc}(h_i) = \frac{|roots(h_i)|}{|\mathcal{R}|} \left(\frac{|roots(h_i) \cap \mathcal{C}^+|}{|roots(h_i)|} - \frac{|\mathcal{C}^+|}{|\mathcal{R}|} \right) \tag{1}$$

The detailed algorithm is shown in Algorithm 2.

3.2 Driving the Linked Data Search: Heuristics

With a limited time and computational resources, choosing the best strategy becomes the most important factor to obtain the hypotheses. We adapted some existing measures, in order to define the most effective one, where effective means a measure giving the best hypotheses score in the shortest number of cycles.

Algorithm 2 Dedalo's complete algorithm

```
cycle = 0
R ← getRoots(R)                          ▷ R = {r_0, ... ,r_i}
paths ← list()                           ▷ empty list of p
traverseLinkedData(R)                    ▷ steps 1-2 on roots
while (time < limit) do
    rank(paths)                          ▷ step 3
    topPath ← top(paths)
    for hypo in hypos(topPath) do        ▷ step 5
        evaluate(hypo)
        add(hypo, hypos)
    end for
    topValues ← vals(topPath)            ▷ step 4
    traverseLinkedData(topValues)        ▷ step 1-2 on the path values
    remove(topPath, paths)               ▷ new iteration
    cycle++
end while
```

1 – Path Length. Our baseline to compare the other measures is the length of $\mathfrak{p}$. This measure assumes that the best paths are the closest to a root item r_i. $\mathcal{L}en$ counts the number of properties p_i composing a $\mathfrak{p}$, and favours the shortest ones.

$$\mathcal{L}en(\mathfrak{p}) = \frac{1}{||\mathfrak{p}||} \tag{2}$$

Ex. If $\mathfrak{p}_1$=⟨ou:MemberOf⟩ and $\mathfrak{p}_2$=⟨ou:participatedIn→ou:ledBy⟩, then $\mathcal{L}en(\mathfrak{p}_1) > \mathcal{L}en(\mathfrak{p}_2)$.

2 – Path Frequency. $\mathcal{F}q$ estimates the frequency of a path $\mathfrak{p}$ among the dataset $\mathcal{R}$ by counting how many roots r_i share $\mathfrak{p}$. It assumes that the most important paths are the most frequent.

$$\mathcal{F}q(\mathfrak{p}) = \frac{|roots(\mathfrak{p})|}{|\mathcal{R}|} \tag{3}$$

Ex. In Fig. 2, if $\mathfrak{p}_1$=⟨ou:MemberOf⟩ and $\mathfrak{p}_2$=⟨ou:exMemberOf⟩, then $\mathcal{F}q(\mathfrak{p}_1) > \mathcal{F}q(\mathfrak{p}_2)$.

3 – Pointwise Mutual Information. $\mathcal{PMI}$ is used in Information Theory and Statistics to measure the discrepancy of a pair of random variables x and y given their joint distribution $p(x|y)$ and individual distributions $p(x)$ and $p(y)$. In our scenario, we measure the probability that $\mathfrak{p}$ is shared by the root items of the considered cluster $\mathcal{C}^+$.

$$\mathcal{PMI}(\mathfrak{p}) = log\frac{|roots(\mathfrak{p}) \cap \mathcal{C}^+|}{|\mathcal{R}| \times |roots(\mathfrak{p})|} \tag{4}$$

Ex. If `<ou:MathieuDAquin>` and `<ou:MiriamFernandez>` are roots of $\mathcal{C}^+$, while `<ou:MartaSabou>` and `<ou:VanessaLopez>` are not, by comparing $\mathfrak{p}_1$=⟨ou:Mem-

berOf⟩ with $\mathfrak{p}_2$=⟨ou:exMemberOf⟩, then $\mathcal{PMI}(\mathfrak{p}_1)$>$\mathcal{PMI}(\mathfrak{p}_2)$ because $\mathfrak{p}_1$ is only shared by the items of $\mathcal{C}^+$.

4 – Adapted TFIDF. We adapted the very well known TFIDF measure to evaluate the relevancy of a path $\mathfrak{p}$ (the *term*) in a given cluster $\mathcal{C}^+$, compared to its frequency across $\mathcal{C}$ (the set of *documents*).

$$TFIDF(\mathfrak{p}) = \frac{|roots(\mathfrak{p}) \cap \mathcal{C}^+|}{|\mathcal{C}^+|} \times log \frac{|\mathcal{C}|}{|\{\mathcal{C}_i \in \mathcal{C} | roots(\mathfrak{p}) \cap \mathcal{C}_i \neq \emptyset\}|} \tag{5}$$

Ex. If $\mathfrak{p}_1$=⟨ou:MemberOf⟩ and $\mathfrak{p}_2$ = ⟨ou:exMemberOf⟩ and `<ou:MathieuDAquin>` and `<ou:MiriamFernandez>` are in $\mathcal{C}^+$, then TFIDF($\mathfrak{p}_1$)>TFIDF($\mathfrak{p}_2$) as $\mathfrak{p}_1$ is only shared by roots belonging to $\mathcal{C}^+$.

5 – Delta function. We developed a function comparing the number of values of a $\mathfrak{p}$ and the number of clusters in the dataset. Δ assumes that the best $\mathfrak{p}$ is the one having a different end value v_i for each cluster $\mathcal{C}_i \in \mathcal{C}$, so $|vals(\mathfrak{p})| = |\mathcal{C}|$. The closer the cardinality of $vals(\mathfrak{p})$ is to that of $\mathcal{C}$, the better the score is.

$$\Delta(\mathfrak{p}) = \frac{1}{1 + ||vals(\mathfrak{p})| - |\mathcal{C}||} \tag{6}$$

Ex. Given $|\mathcal{C}| = 2$ and $\mathfrak{p}$=⟨ou:participatedIn→ou:ledBy⟩, if $|vals(\mathfrak{p})|$=2 means that there is a different value for each cluster in $\mathcal{C}$ and therefore $\Delta(\mathfrak{p})$ is 1. On the other hand, with $\mathfrak{p}$=⟨ou:MemberOf⟩, $\Delta(\mathfrak{p})$ would be low as the only value of $\mathfrak{p}$ is `<ou:KMi>`. Similarly, if the values of $\mathfrak{p}$ were too sparse (i.e. $|vals(\mathfrak{p})|$>2)), $\Delta(\mathfrak{p})$ would also be very low.

6 – Entropy. Starting with Shannon's theory [21], a broad variety of works have applied the notion of entropy to graphs a networks in different disciplines (see [4, 14]for detailed surveys). Entropy (H, the Greek letter "eta") is a measure analysing the performance of communication channels. According to [14], given a random process $\mathcal{X}$=$\{x_0, x_1, ..., x_n\}$ with n possible outcomes, the amount of uncertainty removed by equiprobable messages increases monotonically with the number of existing messages, meaning that the bigger is n, the less information is gained (and the more $\mathcal{X}$ is uncertain). Considering this, we used a naïve adaptation of Shannon's Entropy, in which the random process $\mathcal{X}$ corresponds to $\mathfrak{p}$, while its n possible outcomes are the values $v_i \in vals(\mathfrak{p})$.

$$\mathsf{H}(\mathfrak{p}) = \sum_{i=1}^{|vals(\mathfrak{p})|} \frac{|roots(\langle \mathfrak{p}.v_i \rangle)|}{|\mathcal{R}|} log \frac{|roots(\langle \mathfrak{p}.v_i \rangle)|}{|\mathcal{R}|} \tag{7}$$

Ex. $\mathfrak{p}_1$=⟨ou:MemberOf⟩ and $\mathfrak{p}_2$= ⟨ou:interestedIn⟩. $\mathfrak{p}_1$ has only one possible outcome, so there is no information gain (also defined as "surprise") when finding it in the graph. The gain of information is much higher with $\mathfrak{p}_2$, as it has more uncertain values and therefore $\mathsf{H}(\mathfrak{p}_2) > \mathsf{H}(\mathfrak{p}_1)$.

7 – Conditional Entropy. Similarly, Conditional Entropy measures the information gain of a random variable $\mathcal{X}$ given the knowledge of a random variable $\mathcal{Y}$. In this scenario, $\mathsf{H}(\mathfrak{p}|\mathcal{C}^+)$ measures how much information gain $\mathfrak{p}$ brings, if we know which items belong to $\mathcal{C}^+$ (i.e. how specific $\mathfrak{p}$ and its values are in $\mathcal{C}^+$).

$$\mathsf{H}(\mathfrak{p}|\mathcal{C}^+) = \sum_{i=1}^{|vals(\mathfrak{p})|} \frac{|root(\langle p.v_i \rangle) \cap \mathcal{C}^+|}{|\mathcal{R}|} log \frac{|root(\langle p.v_i \rangle) \cap \mathcal{C}^+|)}{|root(\langle p.v_i \rangle)|} \tag{8}$$

Ex. If `<ou:MathieuDAquin>`, `<ou:VanessaLopez>` and `<ou:MartaSabou>` are $r_i \in \mathcal{C}^+$, and $\mathfrak{p}_1$=⟨ou:MemberOf⟩ and $\mathfrak{p}_2$= ⟨ou:interestedIn⟩ then $\mathsf{H}(\mathfrak{p}_2) > \mathsf{H}(\mathfrak{p}_1)$ because Semantic Web is specific to $\mathcal{C}^+$ only.

4 Experiments

This section presents the different experiments we ran to evaluate the paths ranking measures, in order to find the best one. The datasets, resulting hypotheses, and evaluations here presented are also available online[1].

4.1 Datasets

As an input for Dedalo's Linked Data traversal, we used three datasets, differing in topic (authors, papers and books), size and clustering methods (see Table 2). While the two first can be seen as test examples in a restricted and well understood area, the third represents a realistically large use case (close to 7,000 root items, leading to the traversal of millions of triples distributed in several datasets). This demonstrates the feasibility of the approach at different scales and using clusters that can be easily understood and evaluated. Future work will focus on increasing the complexity of the use cases.

Table 2: Detailed description of the datasets used for the experiments.

Dataset	Size	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{C}\rvert$	Clustering method
KMiA	small	92	6	Network partitioning clustering
KMiP	medium	865	6	X-KMeans clustering
Huds	large	6969	11	KMeans clustering

KMiA – *The Knowledge Media Institute co-authorship*. A set of researchers have been clustered according to the papers they have written together. We obtained 6 clusters that an expert validated as consisting of people working on the same topics.
KMiP – *The Knowledge Media Institute publications*. Research papers from the department have been clustered according to the words that have been used in the abstract (TFIDF-weighted keywords). In this case, the expert explained that papers about the same topic have been clustered together.
Huds – *The books borrowing observations*. Books borrowed by university students have been clustered according to the Faculty the students belong to. The expert explained that books of the same topics have been clustered together.

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

4.2 Best Hypotheses

Dedalo was run to find the best hypotheses for clusters of each dataset. Some examples of the best hypothesis $top(\mathcal{H})$ automatically found at different iterations are presented in Table 3. For the purpose of the reader's understanding, the second column shows the explanation the experts have given.

Table 3: Examples of $top(\mathcal{H})$ in our experiments. The full URIs are indicated in the online results.

	$\mathcal{C}^+$	$\lvert\mathcal{C}^+\rvert$	$top(\mathcal{H})$	WR_{acc}
KMiA	(1) Semantic Web people	22	h_1=⟨tag:taggedWithTag.ou:SemanticWeb⟩	7.6%
			h_2=⟨org:hasMembership→ox:hasPrincipalInvestigator→ org:hasMembership.ou:SmartProducts⟩	12.8%
	(2) Learning Technology people	23	h_1=⟨org:hasMembership.ou:open-sensemaking-communities⟩	7.3%
			h_2=⟨org:hasMembership→ox:hasPrincipalInvestigator→ org:hasMembership.ou:SocialLearn⟩	12.7%
KMiP	(1) "*learning data, user, technology*" papers	601	h_1=⟨dc:creator→org:hasMembership.ou:StoryMakingProject⟩	3.8%
			h_2=⟨dc:creator→org:hasMembership→ox:hasPrincipal-Investigator→ntag:isRelatedTo.ou:LearningAnalytics⟩	4.2%
	(2) "*ontology, knowledge, system*" papers	220	h_1=⟨dc:creator.ou:EnricoMotta⟩	6.1%
			h_2=⟨dc:creator→ntag:isRelatedTo.ou:SemanticWeb⟩	7.3%
Huds	(1) borrowings of Music Technology students	335	h_1=⟨dc:subject.bl:SoundsRecording⟩	0.2%
			h_2=⟨dc:creator→bl:hasCreated→dc:subject.bl:SoundsRecording⟩	0.4%
			h_3=⟨dc:creator→owl:sameAs→skos:broader skos:broader→skos:broader.lcsh:PhysicalScience⟩	0.5%
	(2) borrowings of Theatre students	919	h_1=⟨dc:subject.bl:EnglishDrama⟩	0.4%
			h_2=⟨dc:creator→owl:sameAs→skos:narrower.lcsh:EnsembleTheatre⟩	0.7%
			h_3=⟨dc:creator→bl:hasCreated→dc:subject.bl:EnglishDrama⟩	1.3%

In KMiA–1, the explanation for h_2 is that people who worked on Semantic Web have been clustered together because they have all been part of a project whose director was someone working himself on the SmartProducts project[2] (with a WR_{acc} score of 12.8%), which is much deeper in the graph than h_1 (those people are associated to the Semantic Web topic, WR_{acc} 7.6%). Also, this kind of explanations could only be given by someone knowing the department well enough to affirm that those people worked in projects under the same director. Typically, this is an example in which explanations are hidden and only an expert with the right background knowledge could provide it.

Those results also demonstrate that Dedalo is agnostic to the process used to obtain the cluster, as well as the topic of the dataset, as by changing them, we obtained satisfactory hypotheses.

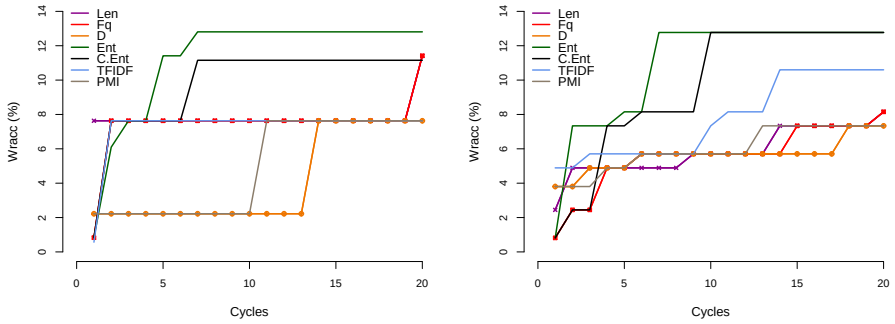
Finally, we also remark how, by using the connections between datasets in the Linked Data cloud, we can also get better explanations. In Huds–1, while

[2] http://www.smartproducts-project.eu/

first we get explanations from the British Library dataset[3] ("books borrowed by students of the Music Technology faculty are about sound recording", WR_{acc} 0.2%), when descending the graph we reach the Library Of Congress dataset[4] and find a better explanation that "those borrowed books are about a topic referenced in the LCSH dataset as a narrower topic of Physical Science" (WR_{acc} 0.5%). Although it is an intuitively easier explanation to make, it shows that more accurate explanations can be found using Linked Data connections among datasets and domains.

4.3 Results and Discussion

We compared the measures presented in section 3 on our examples, to see which was the fastest at reaching the best hypotheses given a fixed number of iterations. In Fig. 3–5, the X axis represents the cycles the process has gone through, and the Y axis represents the WR_{acc} score (in %) of the $top(\mathcal{H})$ found at that given iteration. As we explained, each improvement of the WR_{acc} score means that new $top(\mathcal{H})$ have been found by Dedalo.



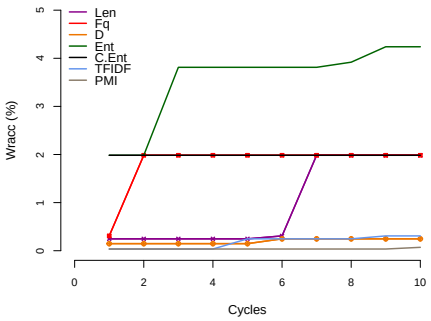
(a) KMiA–1. Semantic Web people. (b) KMiA–2. Learning Analytics people.

Fig. 3: KMiA results. Dedalo ran 20 iterations.

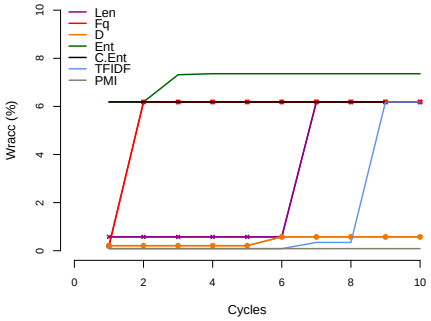
Our experiments show that Entropy outperforms the other measures. The Entropy method reduces redundancy (i.e. following wrongs paths) and allows Dedalo to directly detect the most promising paths to follow. The Conditional Entropy measure, showing a very good performance as well, is the second best performing in 5 of out 6 experiments. In Fig. 5b, Conditional Entropy even finds better explanations of the cluster. The reason is that items of that cluster had hypotheses specific enough when compared to the rest of the dataset.

The $\mathcal{PMI}$, TFIDF and Δ measures have the worst performances, possibly because our use-cases were homogeneously composed and each entity, regardless which cluster it belonged to, had approximately the same properties. For instance, TFIDF works relatively well in the case illustrated in Fig. 3b. In that case, the experts explained that we were dealing with a more heterogeneous cluster of data. $\mathcal{L}en$ and $\mathcal{F}q$ are good in finding an explanation in the first cycles,

[3] http://bnb.data.bl.uk/
[4] http://id.loc.gov/authorities/subjects.html

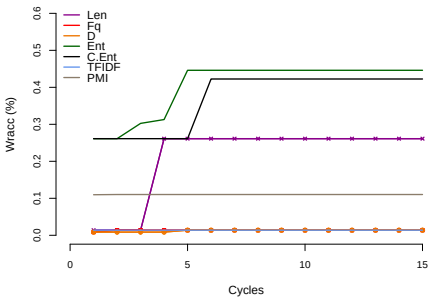


(a) KMiP–1. Learning analytics topic.

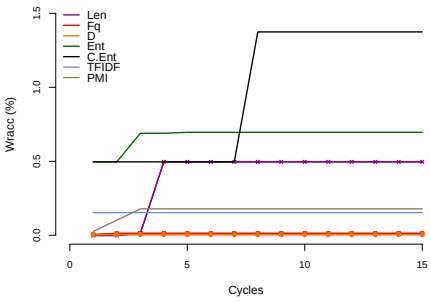


(b) KMiP–2. Semantic Web topic.

Fig. 4: KMiP results. Dedalo ran 10 iterations.



(a) Huds–1. Music Technology.



(b) Huds–2. Theatre.

Fig. 5: Huds results. Dedalo ran 15 iterations.

but then they plateau and take time before getting any improvement. They are not able to follow the correct path, until it finally shows up in the queue of paths to further analyse. $\mathcal{L}en$ seems to have a better performance on big clusters with smaller numbers of properties, as shown in Fig. 5a and 5b.

The experiments also showed an apparent phenomenon that the bigger the dataset, the lower is WR_{acc}. This can probably be explained by the fact that it is harder to find strong explanations in a larger population.

In Fig. 6, we compared the time the measures need to reach the same hypothesis. We choose as $top(\mathcal{H})$ the last and most common hypothesis after a fixed number of iterations (20th, 10th and 15th). In most of the examples, relatively to the scale of the dataset, Entropy is among the fastest measures also in time, while Conditional Entropy appears slightly slower.

5 Conclusions and future Work

In this work we presented Dedalo, an ILP-inspired approach that automatically produces explanations for clusters using Linked Data as background knowledge. We have shown not only that hidden explanations for clusters can be extracted from Linked Data, and that this can come from the different domains connected

in the Linked Data cloud, but also that it is important to correctly choose the direction in the graph in order to save computational effort and time. We developed and evaluated different measures to traverse Linked Data to access the explanation in the shortest time. The Entropy and Conditional Entropy measures performed best in all test cases.

In our future work, we intend to pursue three main lines: (i) exploring different hypothesis evaluation measures, other than WR_{acc}, to detect if the best explanation or the heuristic are affected by changing the measure; (ii) refining the discovery of paths, using inverse properties, and of hypotheses, combining the best hypotheses to obtain a better score; and finally (iii) deal with the issue of the lack of connections between datasets. In fact, we are aware that Dedalo works as far as Linked Data sources (and therefore, domains) are interconnected. In another example, in which students have been clustered according to the region they come from, it turned out that in certain regions, some faculties attract more students than others (for instance, a lot of students have enrolled in the Health&Social Care Faculty in the East-Midlands, while the Law&Business Faculty attracts students from regions around London). While we know that there is a possibly eco-demographic explanation to this, and that Linked Data contain datasets to give us such information, at the current stage we cannot obtain it because of the lack of connections between these datasets. Our future work will be focused on this issue.

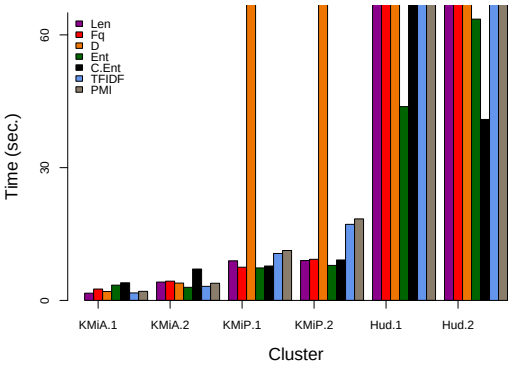


Fig. 6: Time (in seconds) the measures needed to reach $top(\mathcal{H})$. The process assumes that the data have been cached locally, as the times to retrieve entities from different datasets are not comparable.

References

1. Brisson, L., Collard, M., & Pasquier, N. (2005, November). Improving the knowledge discovery process using ontologies. In Proceedings of the IEEE MCD international workshop on Mining Complex Data (pp. 25-32).
2. Brisson, L., & Collard, M. (2009). How to Semantically Enhance a Data Mining Process?. In Enterprise Information Systems (pp. 103-116). Springer Berlin Heidelberg.
3. d'Aquin, M. & Jay, N. (2013) Interpreting Data Mining Results with Linked Data for Learning Analytics: Motivation, Case Study and Direction, LAK 2013.
4. Dehmer, M., & Mowshowitz, A. (2011). Generalized graph entropies. Complexity, 17(2), 45-50.

5. Geng, L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR), 38(3), 9.
6. Grosof, B. N., Horrocks, I., Volz, R., & Decker, S. (2003, May). Description logic programs: Combining logic programs with description logic. In *Proceedings of the 12th international conference on World Wide Web* (pp. 48-57). ACM.
7. King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., & Clare, A. (2009). The automation of science. Science, 324(5923), 85-89.
8. Lavrač, N., Vavpetič, A., Soldatova, L., Trajkovski, I., & Novak, P. K. (2011). Using ontologies in semantic data mining with SEGS and g-SEGS. In Discovery Science (pp. 165-178). Springer Berlin Heidelberg.
9. Lisi, F. A. (2010). Inductive Logic Programming in Databases: From Datalog to DL+log. *Theory and Practice of Logic Programming*, 10(03), 331-359.
10. Lisi, F. A., & Esposito, F. (2009). On ontologies as prior conceptual knowledge in inductive logic programming. In Knowledge discovery enhanced with semantic and social information (pp. 3-17). Springer Berlin Heidelberg.
11. Marinica, C., & Guillet, F. (2010). Knowledge-based interactive postmining of association rules using ontologies. Knowledge and Data Engineering, IEEE Transactions on, 22(6), 784-797.
12. Moss, L., Sleeman, D., Sim, M., Booth, M., Daniel, M., Donaldson, L., & Kinsella, J. (2010). Ontology-driven hypothesis generation to explain anomalous patient responses to treatment. Knowledge-Based Systems, 23(4), 309-315.
13. Motik, B., & Rosati, R. (2006). Closing semantic web ontologies. Technical report, University of Manchester, UK.
14. Mowshowitz, A., & Dehmer, M. (2012). Entropy and the complexity of graphs revisited. Entropy, 14(3), 559-570.
15. Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. The Journal of Logic Programming, 19, 629-679.
16. Mulwad, V., Finin, T., Syed, Z.,& Joshi, A. (2010). Using Linked Data to Interpret Tables. In COLD2010.
17. Paulheim, H., Generating Possible Interpretations for Statistics from Linked Open Data (2012). In Proceedings of ESWC 2012, (560-574). Springer, Berlin.
18. Paulheim, H. Exploiting Linked Open Data as Background Knowledge in Data Mining. In: CEUR workshop proceedings DMoLD 2013 collocated with ECMLP-KDD 2013; 1-10. RWTH, Aachen, 2013.
19. Racunas, S. A., Shah, N. H., Albert, I., & Fedoroff, N. V. (2004). HyBrow: a prototype system for computer-aided hypothesis evaluation. Bioinformatics, 20(suppl 1), i257-i264.
20. Roos, M., Marshall, M. S., Gibson, A., Schuemie, M., Meij, E., Katrenko, S., & Adriaans, P. (2009). Structuring and extracting knowledge for the support of hypothesis generation in molecular biology. BMC bioinformatics, 10(Suppl 10), S9.
21. Shannon, C. (1948). A Mathematical Theory of Communication. Bell System Technical Journal 27 (3): 379–423.
22. Tiddi, I., d'Aquin, M., & Motta, E. (2013). Explaining Clusters with Inductive Logic Programming and Linked Data. 12th International Semantic Web Conference. Sydney.
23. Tiddi, I. (2013). Explaining data patterns using background knowledge from Linked Data, ISWC 2013 Doctoral Consortium, Sydney, Australia.
24. Zapilko, B., Harth, A., & Mathiak, B. (2011): Enriching and analysing statistics with Linked Open Data. In: Eurostat (Hrsg.): NTTS - Conference on New Techniques and Technologies for Statistics. S8 Paper 1, Brüssel.