Complex Linking in a Nutshell

Axel-Cyrille Ngonga Ngomo[1], Timofey Ermilov[1]

University of Leipzig
AKSW Group
Johannisgasse 26, 04103 Leipzig

Abstract. The Web of Data is growing at a stunning rate. Yet, the percentage of statements that are links between knowledge bases remains remarkably low. This is partly due to the process of specifying links between knowledge bases being so tedious. In this demo, we present the COLANUT interface, an intelligent interface that can suggest link specifications and allows to edit them graphically. We present the different steps from specifying the knowledge bases that are to be linked to a semi-automatically generated link specification.

1 Introduction

The core idea behind the Linked Data paradigm is to facilitate the transition from the document-oriented Web to the Semantic Web by extending the Web with a data commons consisting of interlinked data sources. While the number of triples in data sources increases steadily and has surpassed 28 billions[1], links still constitute less than 3% of the total number of triples available on the Linked Data Web. Two main challenges arise when trying to discover links between knowledge bases. The first challenge is intrinsically related with the link discovery process. The a-priori complexity of a matching task is proportional to the product of the number of instances in the source and target data source, an unpractical proposition as soon as the source and target knowledge bases become large. For example, discovering duplicate cities in DBpedia [1] alone would necessitate approximately 0.15×10^9 distance computations. Time-efficient Link Discovery framework such as LIMES [3] address this drawback.

The second challenge of the link discovery process lies in the selection of an appropriate similarity space (i.e., property mapping and corresponding distance measure). This selection is usually carried out manually, in most cases simply by guessing. Yet, the choice of a suitable link specification decides upon whether satisfactory links can be discovered. Methods for assisting the user during the specification of links can ignite the massive link specification process needed to fully address the fourth Linked Data principle, i.e., the provision of links between knowledge bases.

In this paper, we present the COLANUT (COmplex Linking in A NUTshell) interface. COLANUT resides on top of LIMES and supports the user

[1] http://www4.wiwiss.fu-berlin.de/lodcloud/

during the link specification process by two means: First, it provides a graphical user interface that makes it easier for the user to specify his link specification. Second, it generates suggestions for mapping classes and properties, therewith allowing even lay users who do not possess much schema knowledge to effectively specify links between knowledge bases. By these means, it goes beyond current editors for linking[2] as it does not only allow to edit a link configuration but rather automatically suggests one, which can then be edited by the user. In the following, we present the three steps from specifying a knowledge base to generating a complete link specification by using the Drugbank[3] and Sider[4] knowledge bases as examples. The COLANUT demo can be accessed at `http://limes.aksw.org/colanut`.

2 Overview

In this section, we show how COLANUT supports the user during the creation of link configuration, which we define formally as follows:

Definition 1 (Link Specification). *A link specification consists of three parts: (1) two sets of restrictions $\mathcal{R}_1^S$ ... $\mathcal{R}_m^S$ resp. $\mathcal{R}_1^T$ ... $\mathcal{R}_k^T$ that specify the sets S resp. T, (2) a specification of a similarity metric σ and (3) a threshold τ.*

A restriction $\mathcal{R}$ is generally a logical predicate. Typically, restrictions in link specifications state the `rdf:type` of the elements of the set they describe, i.e., $\mathcal{R}(x) \leftrightarrow x$`rdf:type someClass` or the features the elements of the set must have, e.g., $\mathcal{R}(x) \leftrightarrow (\exists y : x$ `someProperty` $y)$. Each $s \in S$ must abide by each of the restrictions $\mathcal{R}_1^S$ ... $\mathcal{R}_m^S$, while each $t \in T$ must abide by each of the restrictions $\mathcal{R}_1^T$ ... $\mathcal{R}_k^T$.

2.1 Suggestion of restrictions

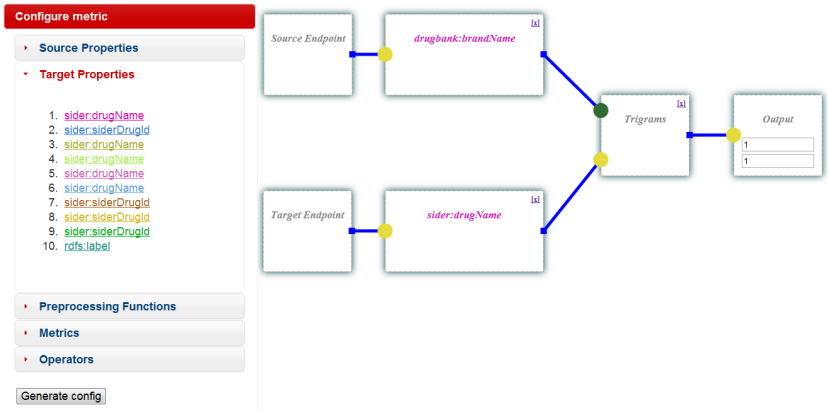
The user begins by setting the source and target knowledge base as shown in Figure 1(a). Clicking on the "OK" button initiates the suggestions of restrictions. This suggestion is carried out by computing a stable matching of the classes from the source knowledge base to the target knowledge base. The current default matching approach implemented by COLANUT regards the mapping of classes between two knowledge bases $\mathcal{K}_S$ and $\mathcal{K}_T$ as a Hospital/Residents ($\mathcal{HR}$) problem [2]. Let R be a set of residents and H a set of hospitals. Furthermore, let the preference functions $\mu : H \times R \rightarrow \{1, ..., |H|\}$ resp. $\gamma : H \times R \rightarrow \{1, ..., |R|\}$ be injective functions that give the degree to which a resident likes a hospital resp. a hospital a resident. A function $s : R \rightarrow H$ is called a stable matching iff for all $(r, r') \in R^2$ and $(h, h') \in H^2$:

$$(s(r) = h) \wedge (s(r') = h') \wedge (\mu(r, h') > \mu(r, h)) \rightarrow (\gamma(r', h') > \gamma(r, h')) \text{ and } (1)$$

[2] See `http://www.assembla.com/spaces/silk/wiki/Silk_Workbench` for example
[3] `http://www4.wiwiss.fu-berlin.de/drugbank/`
[4] `http://www4.wiwiss.fu-berlin.de/sider/`

(a) Specification of knowledge bases

(b) Specification of restrictions



(c) Creation of link specification

Fig. 1. Main windows of COLANUT

$$(s(r) = h) \wedge (s(r') = h') \wedge (\gamma(r, h') > \gamma(r, h)) \rightarrow (\mu(r', h') > \mu(r, h'))\,. \quad (2)$$

COLANUT detects stable matchings of classes by computing μ and γ of the classes C_S of $\mathcal{K}_S$ and C_T of $\mathcal{K}_T$ as follows

$$\mu(C_S, C_T) = \gamma(C_S, C_T) = |\{s \texttt{ type } C_s \wedge s \texttt{ sameAs } t \wedge t \texttt{ type } C_T\}|. \quad (3)$$

If no `sameAs` links exist between $\mathcal{K}_S$ and C_T of $\mathcal{K}_T$, COLANUT uses the following *fallback* definition:

$$\mu(C_S, C_T) = \gamma(C_S, C_T) = |\{s \texttt{ type } C_s \wedge s \texttt{ p } x \wedge t \texttt{ type } C_T \wedge t \texttt{ q } x\}|, \quad (4)$$

where `p` and `q` can be any property.

COLANUT then select the pair (C_S, C_T) with $C_S = s(C_T)$ and the highest $\mu(C_S, C_T) = \gamma(C_S, C_T)$ as the best matching pair and suggests it to the user. In our example, COLANUT detects that `drugbank:drugs` should be mapped to `sider:drugs`. Note that this detection is not based on string similarity. For example, mapping Sider to Dailymed leads to `sider:drugs` being mapped to `dailymed:Offer`. In the event that the restrictions are unsatisfactory, the user can choose to select other restrictions manually by simply typing them in. If the user gives in a class that belong to the domain or range of the stable matching

function s, the corresponding class is suggested to the user, who can choose to either accept the suggestion or override it.

2.2 Semi-Automatic Link Specification

The detection of the best matching pairs of properties is very similar to the computation of the best matching classes. For datatype properties `p` and `q`, we set:

$$\mu(p,q) = \gamma(p,q) = \{s \texttt{ type } C_S \wedge s \texttt{ p } x \wedge t \texttt{ type } C_T \wedge t \texttt{ q } x\}. \quad (5)$$

COLANUT then select the of properties that lead to the highest value of μ and sets it as the default pair of property for the link specification. Depending on their datatype, the properties are either linked with the trigrams (for strings) or the Euclidean (for numbers) similarity measures. By these means, a default link configuration is generated as shown in Figure 1(c). In this example, COLANUT suggest linking Drugbank and Sider via the properties `drugbank:brandName` and `sider:drugName`, which corresponds to the properties selected by an expert manually. The default threshold, i.e., the last part of a link specification, is set to 1, i.e., perfect similarity. The user can choose to change it at will.

In addition to suggesting an initial configuration, the interface supports the user during the process of generating complex link specifications by color-coding matching pairs of properties. In addition, the interface provides all conjunction operations supported by LIMES for the creation of complex configurations. The user can consequently select between several pre-processing functions (including `replace` for replacing characters, `number` for ensuring that the property value is set to a numeric value, etc.) for normalizing the property values before computing the similarity between entities. In addition, several operators such as AND, OR, MIN and MAX can be selected for combining metrics and thresholds.

We applied COLANUT to the five knowledge bases DBpedia (`dbp`), Drugbank (`dbk`), Diseasome (`ds`), Sider (`sd`) and Dailymed (`dm`) and compared the suggested class mappings with those generated by a human expert. In all cases, we generated mappings that were logically equivalent to those created by a human expert. For example, when mapping Sider and Dailymed, COLANUT suggested to map the class `sd:drugs` with `dm:Offer`. Yet, the human expert chose `dbk:drugs` instead of `dm:Offer`. However, since these two classes are linked via `owl:sameAs`, the mappings of the expert and COLANUT are equivalent.

3 Conclusion and Future Work

We presented COLANUT, an intelligent graphical user interface for the assisted and semi-automatic specification of link configurations. COLANUT is based on the LIMES framework and suggest components of configurations to its user. In addition, our interface visualizes the results of the matching process, thus allowing even lay user to specify a link configuration with ease. In future work, we will conduct user tests to measure the time gained by using COLANUT instead of scripting link specifications manually.

Source	Target	Class Mapping	Property Mapping
Drugbank	Sider	dbk:drugs→sd:drugs	dbk:brandName→sd:drugName
Drugbank	Diseasome	dbk:targets→ds:genes	rdfs:label→dbk:geneName
Drugbank	DBpedia	dbk:targets→dbp:Protein	property:name→dbk:name
Drugbank	Dailymed	dbk:drugs→dm:Offer	dm:name→rdfs:label
Sider	Diseasome	sd:sideEffects→ds:diseases	sd:sideEffectName→ds:name
Sider	DBpedia	sd:sideEffects→dbp:Disease	sd:sideEffectName→foaf:name
Sider	Dailymed	sd:drugs→dm:Offer	dbk:name→dm:name
Diseasome	DBpedia	ds:diseases→dbp:Disease	rdfs:label→dm:name
Diseasome	Dailymed	No mapping	No mapping
DBpedia	Dailymed	dbp:Organisation →dm:organization	rdfs:label→rdfs:label

Table 1. Mappings computed by ALOE for ten different pairs of knowledge bases

References

1. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *ISWC*, pages 722–735, 2008.
2. D. Manlove, R. Irving, K. Iwama, S. Miyazaki, and Y. Morita. Hard variants of stable marriage. *Theoretical Computer Science*, 276(1-2):261 – 279, 2002.
3. Axel-Cyrille Ngonga Ngomo and Sören Auer. A time-efficient approach for large-scale link discovery on the web of data. In *IJCAI*, 2011.