SOFYA: Semantic on-the-fly Relation Alignment

Maria Koutraki
University of Paris-Saclay
45 Av. des États Unis
78000, Versailles
France
kom@prism.uvsq.fr

Nicoleta Preda
University of Paris-Saclay
45 Av. des États Unis
78000, Versailles
France
preda@prism.uvsq.fr

Dan Vodislav
University of Cergy-Pontoise
2 Av. Adolphe-Chauvin
95302, Cergy-Pontoise
France
dan.vodislav@u-ergy.fr

ABSTRACT
Recent studies have shown the rise of Web data, in particular Linked Data, with up to now, more than 1000 datasets in the Linked Open Data Cloud (LOD). These datasets are mostly of entity-centric nature and are highly heterogeneous in terms of domains, language, schema, etc. Hence, the vision of uniformly querying such resources in the LOD has a long way to go. While equivalent entity instances across datasets are often linked by \texttt{sameAs} links, relations from different datasets and schemas are usually not aligned.

In this paper, we propose an on-line \textit{instance-based} relation alignment approach. The alignment may be performed during query execution and requires partial information from the datasets. We align relations to a target dataset using association rule mining approaches. We sample for equivalent entity instances with two main sampling strategies. Preliminary experiments show that we are able to align relations with high accuracy, even if accessing the entire datasets is impossible or impractical.

1. INTRODUCTION

As of April 2015, the publicly accessible part of the LOD project counts more than 1000 datasets, which together store more than 30 billion facts. The datasets span across different domains, such as social Web, government data, geographic data, or the life sciences. Moreover, the datasets are highly heterogeneous in terms of schemas, of quality of the data, and only 2% of the schemas are aligned across different datasets [6]. Many of these datasets are accessible through \texttt{SPARQL endpoints}, yet uniformly querying them remains a long way to go.

Motivation. Successful examples include well known knowledge bases (KB) like DBpedia, YAGO, and Freebase, which comprise factual statements about real world entities. These facts are typically stored as triples \langle \textit{subject}, \textit{relation}, \textit{object} \rangle (e.g \texttt{Frank_Sinatra, wasBornIn, USA}). Yet, even for such KBs, the same entity can have different identifiers (e.g. \texttt{Frank_Sinatra, (Singer) or Sinatra}). Similarly, equivalent relations across KBs use different names (e.g. \texttt{wasBornIn} and \texttt{bornInCountry}), hence makes them non-interoperable, such that queries cannot join information across KBs.

Challenges. Several approaches have been proposed to align relations across datasets [9, 7, 3], but in all these cases alignment is performed on the entire KB snapshot. In the real world, however, one may not always have access to the entire dataset. First, KBs are typically quite large (e.g. YAGO, requires 100GB of space on disk), and it is rather impractical to download several entire KBs just to answer a single query. Second, performing relation alignment on KB snapshots, may miss out KB updates. For time-sensitive data, it is better to query the data dynamically. Finally, not all KBs can be freely downloaded. Some providers allow users to issue a limited number of queries to KB via a SPARQL endpoint, but do not allow them to download the entire dataset. In this line, [5] focus on discovering schema alignment on data streams, however, this does not represent any guarantee that one can align any relation given the stream of data.

Contributions. In this paper, we propose an \textit{instance-based on-the-fly} approach for relation alignment between two KBs. Our method requires only a \texttt{SPARQL endpoint} for each dataset. Given a relation name in a source KB, e.g. coming from a query on that KB, our method automatically finds corresponding relations in the target dataset, without any need to download the data. Since our method works with few queries, it could be used at query time.

The main idea behind our approach is to use samples of data from both KBs in order to identify candidate relations, then rely on inductive logic programming (ILP) to validate them. Existing works [1, 8], use ILP to mine rules in order to align hierarchies of entities. We go beyond this goal, and want to express more complex mappings, by mining logical rules such as \texttt{k1: wasBornIn(x, y) \Rightarrow k2: bornInCountry(x, y)}.

In particular, we perform two types of alignments, \textit{subsumption} and \textit{equivalence}, which can be expressed as logical rules. However, as we will show below, such rules cannot be solely mined with standard ILP approaches from small samples of instances. Hence, we develop smart sampling methodologies that are geared to this type of problems. Experiments with real-world datasets show that we can align relations with more than 90% precision, based on only very small samples.

2. APPROACH

2.1 Rule Mining

Given two KBs \( K \) and \( K' \), a relation \( r \) in \( K \) and the set \( E \) of \texttt{sameAs} entity equivalences, we want to find rules \( r' \) in \( K' \) subsumed by \( r \), i.e. \( r' \Rightarrow r \). Candidate relations \( r' \) may be found by sampling \( r(x, y) \), then considering all \( r' \) such that \( r'(x, y) \) for some sample. Equivalence of relations is expressed as a double subsumption: \( r' \Rightarrow r, \text{iff } r' \Rightarrow r \text{ and } r \Rightarrow r' \).

In this work we use two ILP techniques to validate subsumption between relations. A vanilla association rule mining approach [2] could simply regard all absent data as counter-examples (closed world assumption), which yields the following confidence measure:

\[
cwaconf(r' \Rightarrow r) := \frac{\#(x, y) : r'(x, y) \land r(x, y)}{\#(x, y) : r'(x, y)} \tag{1}
\]
where \( \#(x,y) : A \) is the number of pairs \((x,y)\) that fulfill \( A \).

The second technique\cite{4}, works under a open world assumption and considers that a KB knows either all or none of the \( r \)-attributes of some \( x \). In this case, we count as counter-examples for a rule \( r'(x,y) \Rightarrow r(x,y) \) only instances \((x,y)\) such that \( x \) has \( r \) relations, but not \( r(x,y) \). The confidence measure is:

\[
\text{precision}(r' \Rightarrow r) := \frac{\#(x,y) : r'(x,y) \land \neg r(x,y)}{\#(x,y) : r'(x,y) \land \neg r(x,y) \lor \neg r'(x,y)}
\]

### 2.2 Instance Sampling

**Simple Sample Extraction.** We propose a baseline solution that computes a (pseudo-)random set of samples to check if a candidate relation \( r_{sub} \) from \( K' \) satisfies \( r_{sub} \Rightarrow r \). First, we extract from \( K' \) a set of samples that are entities that are subjects in \( r_{sub} \) facts:

\[
S_{r_{sub}} = \{ x_1 \mid r_{sub}(x_1, y_1) \in K', \forall x_2 \forall y_2 \in K : x_1 \equiv x_2 \land y_1 \equiv y_2 \}
\]

The same query extracts the actual \( r_{sub} \) facts where the sample entities occur. More precisely, it extracts the set:

\[
K_{r_{sub}}^{S} = \{ r_{sub}(x_1, y_1) \mid x_1 \in S_{r_{sub}} \land r_{sub}(x_1, y_1) \in K' \}
\]

The actual SPARQL queries that are used to extract the two sets depend on the nature of the relation \( r_{sub} \). For entity-entity relations, we select for a subject \( x_1 \) all the facts \( r_{sub} \) for which there are same\( x_1 \)a\( s \) links to entities in \( K \) for both the subject and the object. Since we do not want to punish the score of the alignment because of incomplete information, we ignore the \( r_{sub} \) facts where the same\( x_1 \)as\( x_2 \) links to entities in \( K \) are.

In the next step the subject and the object of a \( r_{sub} \) are translated to the equivalent entities in \( K \) and create the set:

\[
P_{r_{sub}}^{S} = \{ (x_2, y_2) \mid \exists x_1 \forall y_1 : x_1 \equiv x_2 \land y_1 \equiv y_2 \land r_{sub}(x_1, y_1) \in K_{r_{sub}}^{S} \}
\]

Then corresponding \( r \) instances are extracted:

\[
K_{r_{sub}}^{S} = \{ r(x_2, y_2) \mid r(x_2, y_2) \in K, \exists y_2' : (x_2, y_2') \in P_{r_{sub}}^{S} \land r(x_2, y_2') \}
\]

Note that if for some pair \((x_2, y_2')\) from \( P_{r_{sub}}^{S} \) a fact \( r(x_2, y_2') \) is discovered in \( K \), then we need to select all the other facts \( r(x_2, y') \) of \( x_2 \). This is required by the \( pcaconf \) measure. For simplicity, in this presentation we assumed that the inverse relations have been added to the two KBs. This is why we only consider direct relations.

If \( r_{sub} \) is an entity-literal relation, we retrieve from \( K \) facts of the samples \( S_{r_{sub}} \) and apply string similarity functions to align the literals. Once the sets \( K_{r_{sub}}^{S} \) and \( P_{r_{sub}}^{S} \) are retrieved, we can run the \( pcaconf \) and the \( cwconf \) scorers on the coalesce of the two sets.

**Unbiased Sample Extraction (UBS).** The random selection of the samples is a fair objective approach, but several cases require a more careful selection of unbiased samples when using \( pcaconf \).

**Mining subsumptions that are not equivalences.** Consider the example of a mined subsumption \( K' : \text{composerOf} \Rightarrow K : \text{creatorOf} \). When checking the reverse implication to test equivalence, if the sample includes composers that only created musical compositions, we will find that the two relations are equivalent under \( pcaconf \), while if a composer is also a writer the reverse implication is false. A way to avoid such missing samples is to discover in \( K' \) a relation subsumed by \( K : \text{creatorOf} \) whose domain overlaps with the domain of \( K' : \text{composerOf} \). For instance, we can take the relation \( K' : \text{writerOf} \) and consider for sampling the composers that are also writers.

**Mining overlappings that are not subsumptions.** Consider in \( K' \) the relations \( \text{hasDirectorFor} \) for movies and their directors and \( \text{hasProducer} \) for movies and their producers, then in \( K \) the relation \( \text{directedBy} \) for movies and their directors. Since it often happens that the same person directs and produces the same movie, we might wrongly infer that \( K' : \text{hasProducer} \Rightarrow K : \text{directedBy} \). To filter out such cases even under \( pcaconf \), we may include in the sample movies whose producer and director are different.

To deal with both unbiased samples cases above, our method lays on candidate relations \( K' : r' \) and \( K' : r'' \), subsumed by \( K : r \) for simple samples. Unbiased samples will include facts for \( K' : r' \) and \( K' : r'' \) that share the same subjects but have different objects.

More precisely, unbiased samples would contain \( x \) such as \( r'(x,y), r''(x,y) \Rightarrow \neg r'(x,y) \). In the first case, the existence of \( r'(x,y) \) and \( r''(x,y) \) filters out the wrong equivalence. In the second one, the condition to filter out the wrong subsumption is to have \( r'(x,y) \) but not \( r''(x,y) \). We used here the same identifiers for equivalent entities in \( K \) and \( K' \).

### 3. EXPERIMENTAL EVALUATION

**Datasets.** We conduct our experiments on two KBs, with 92 relations from YAGO2 and 1313 relations from DBpedia.

**Baselines.** As baseline solution we consider the (pseudo-)random selection of Simple Sample Extraction described in Section 2. On the coalesce of the sets of samples retrieved from the two KBs, we have run the two ILP techniques \( pcaconf \) and \( cwconf \).

We evaluate the algorithms for a sample size of 10 samples (subject entities). Table 1 reports our preliminary results. For the two measures \( pcaconf \) and \( cwconf \), we have selected the thresholds \( \tau \) that led to the highest average F1 score for both ways implications, \( yago \subset dbp \) and \( dbp \subset yago \).

**Unbiased Sample Extraction.** The method that we propose extends the baseline solution of \( pcaconf \) by implementing the two strategies for filtering wrong candidates. To eliminate a “wrong” relation we need only one case which shows that there is a contradiction. The results of this method are indicated by the label UBS in Table 1. The results suggest that our method consistently prunes wrong candidates.

**Table 1: Alignment subsumptions – YAGO and DBpedia relations**

<table>
<thead>
<tr>
<th></th>
<th>ILP</th>
<th>yago ( \subset ) dbp</th>
<th>dbp ( \subset ) yago</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau &gt; 0.3 )</td>
<td>pcaconf</td>
<td>P 0.55</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.58</td>
<td>0.48</td>
</tr>
<tr>
<td>( \tau &gt; 0.1 )</td>
<td>cwconf</td>
<td>F1 0.56</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.59</td>
<td>0.53</td>
</tr>
<tr>
<td>UBS</td>
<td>pcaconf</td>
<td>P 0.95</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.97</td>
<td>0.82</td>
</tr>
</tbody>
</table>

### 4. REFERENCES


