A Statistical and Schema Independent Approach to Identify Equivalent Properties on Linked Data

Kalpa Gunaratna  
*Wright State University - Main Campus*, gunaratna.2@wright.edu

Krishnaprasad Thirunarayan  
*Wright State University - Main Campus*, t.k.prasad@wright.edu

Prateek Jain  
*Wright State University - Main Campus*

Amit P. Sheth  
*Wright State University - Main Campus*, amit.sheth@wright.edu

Sanjaya Wijeratne  
*Wright State University - Main Campus*, wijeratne.2@wright.edu

Follow this and additional works at: [https://corescholar.libraries.wright.edu/knoesis](https://corescholar.libraries.wright.edu/knoesis)

Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation

[https://corescholar.libraries.wright.edu/knoesis/565](https://corescholar.libraries.wright.edu/knoesis/565)
A Statistical and Schema Independent Approach to Identify Equivalent Properties on Linked Data

Kalpa Gunaratna†, Krishnaprasad Thirunarayan†, Prateek Jain‡, Amit Sheth†, and Sanjaya Wijeratne†

{kalpa,tkprasad,amit,sanjaya}@knoesis.org, jainpr@us.ibm.com

†Kno.e.sis Center
Wright State University
Dayton OH, USA

‡IBM T J Watson Research Center
Yorktown Heights
New York NY, USA

iSemantics 2013, Graz, Austria
Motivation

Why we need property alignment and it is so important?
Many datasets. We can query!

Therefore, data integration for better presentation is required.
Roadmap

- Background
- Statistical Equivalence of properties
- Evaluation
- Discussion, interesting facts, and future directions
- Conclusion
Existing techniques for property alignment fall into three categories.

I. Syntactic/dictionary based
   - Uses string manipulation techniques, external dictionaries and lexical databases like WordNet.

II. Schema dependent
   - Uses schema information such as, domain and range, definitions.

III. Schema independent
   - Uses instance level information for the alignment.

Our approach falls under schema independent.
Properties capture meaning of triples and hence they are complex in nature.

Syntactic or dictionary based approaches analyze property names for equivalence. But in LOD, name heterogeneities exist. Therefore, syntactic or dictionary based approaches have limited coverage in property alignment.

Schema dependent approaches including processing domain and range, class level tags do not capture semantics of properties well.
Roadmap

- Background
- Statistical Equivalence of properties
- Evaluation
- Discussion, interesting facts, and future directions
- Conclusion
Statistical Equivalence is based on analyzing owl:equivalentProperty.

owl:equivalentProperty - properties that have same property extensions.

Example 1:
Property P is defined by the triples, \{ a P b, c P d, e P f \}
Property Q is defined by the triples, \{ a Q b, c Q d, e Q f \}
P and Q are owl:equivalentProperty, because they have the same extension, \{ \{a,b\}, \{c,d\}, \{e,f\} \}

Example 2:
Property P is defined by the triples, \{ a P b, c P d, e P f \}
Property Q is defined by the triples, \{ a Q b, c Q d, e Q h \}
Then, P and Q are not owl:equivalentProperty, because their extensions are not the same. But they provide statistical evidence in support of equivalence.
Intuition

- Higher rate of subject-object matches in extensions leads to equivalent properties. In practice, it is hard to have exact same extensions for matching properties. Because,
  - Datasets are incomplete.
  - Same instance may be modelled differently in different datasets.
- Therefore, we analyze the property extensions to identify equivalent properties between datasets.
- We define the following notions. Let the statement below be true for all the definitions.
  \( S_1P_1O_1 \) and \( S_2P_2O_2 \) be two triples in Dataset \( D_1 \) and \( D_2 \) respectively.
Definition 1: Candidate Match

The two properties $P_1$ and $P_2$ are a candidate match iff $S_1 \overset{ECR*}{\leftrightarrow} S_2$ and $O_1 \overset{ECR*}{\leftrightarrow} O_2$.

We say two instances are connected by an $ECR^*$ link if there is a link path between the instances using $ECR$ links (* is the Kleene star notation). $ECR$ links are Entity Co-reference Relationships such as those formalized using $owl:sameAs$ and $skos:exactMatch$.

Example

The two datasets DBpedia(d) and Freebase(f)

<table>
<thead>
<tr>
<th>d:Arthur Purdy Stout</th>
<th>d:place of birth</th>
<th>d:New York City</th>
</tr>
</thead>
<tbody>
<tr>
<td>f:Arthur Purdy Stout</td>
<td>f:place of death</td>
<td>f:New York City</td>
</tr>
</tbody>
</table>

- The above is a candidate match, but not equivalent, because intensions are different (coincidental match).
- We need further analysis to decide on equivalence.
Match Count $\mu(P_1, P_2)$ – Number of triple pairs for $P_1$ and $P_2$ that participate in candidate matches.

$$\mu(P_1, P_2) = | \left\{ S_1P_1O_1 \in D_1 \mid \exists S_2P_2O_2 \in D_2 \land S_1 \xrightarrow{ECR^*} S_2 \land O_1 \xrightarrow{ECR^*} O_2 \right\} |$$

Co-appearance Count $\lambda(P_1, P_2)$ – Number of triple pairs for $P_1$ and $P_2$ that have matching subjects.

$$\lambda(P_1, P_2) = | \left\{ S_1P_1O_1 \in D_1 \mid \exists S_2P_2O_2 \in D_2 \land S_1 \xrightarrow{ECR^*} S_2 \right\} |$$

Definition 2: Statistically Equivalent Properties

The pair of properties $P_1$ and $P_2$ are statistically equivalent to degree $(\alpha, k)$ iff,

$$F = \frac{\mu(P_1, P_2)}{\lambda(P_1, P_2)} \geq \alpha,$$

Where, $\mu(P_1, P_2) \geq k$, and $0 < \alpha \leq 1$, $k > 1$
Candidate Matching Algorithm Process

Step 1

Dataset 1

I₁

owl:sameAs

I₂

Dataset 2

d₂:theodore_harold_maiman

P₂=d₂:education.

Step 2

academic.advisees

I₁

P₁=d₁:doctoralStudent

I₂

triple 1

triple 2

triple 3

Step 3

matching resources

I₁=d₁:Willis_Lamb

triple 4

triple 5

I₂=d₂:willis_lamb

property P₁ and property P₂ are a candidate match
Algorithm 1 GenerateCandidateMatches($X$, *namespace*)

Input: $X$, *namespace*
Output: $\lambda$, $\mu$ for each property pair
1: for $i = 1$ to $\text{Size}(X)$ do
2: \hspace{1em} subject $S1 \leftarrow X[i]$
3: \hspace{1em} subject $S2 \leftarrow \text{GetCorsEntity}(\text{subject}S1, \text{namespace})$
4: \hspace{1em} map $l1 \leftarrow \text{ExtractPO}(S1)$
5: \hspace{1em} map $l2 \leftarrow \text{ExtractPO}(S2)$
6: \hspace{1em} for each property $p \in l1$ do
7: \hspace{2em} object $o1 \leftarrow l1(p)$
8: \hspace{2em} $\text{Set} \ o1_{ecr} \leftarrow \text{GetECRLinks}(o1)$
9: \hspace{2em} for each property $q \in l2$ do
10: \hspace{3em} if $p$ exactmatch $q$ then
11: \hspace{4em} $p$ matches $q$
12: \hspace{3em} else
13: \hspace{4em} update $\lambda(p, q)$
14: \hspace{4em} object $o2 \leftarrow l2(q)$
15: \hspace{4em} $\text{Set} \ o2_{ecr} \leftarrow \text{GetECRLinks}(o2)$
16: \hspace{4em} $\text{isMatch} \leftarrow \text{Match}(o1_{ecr}, o2_{ecr})$
17: \hspace{4em} if $\text{isMatch}$ then
18: \hspace{5em} update $\mu(p, q)$
19: \hspace{4em} end if
20: \hspace{3em} end if
21: \hspace{2em} end for
22: \hspace{1em} end for
23: end for

Complexity:
If the average number of properties for an entity is $x$ and for each property, average number of objects is $j$. For $n$ subjects, it requires $n^2j^2x^2 + 2n$ comparisons. Since $n > j$, $n > x$, and $x$ and $j$ are independent of $n$, $O(n)$. 
Example:

Same instances are shown using dotted arrows between the two datasets.

Generated Candidate Matching Property Lists – Matches selected are in Boldface

- [D1:doctoalStudent]  [D2:academic.advisees, 2:2], [D2:influenced, 1:2]  matching list for doctoralStudent
- [D1:birth_place] [D2:place_of_birth, 2:3], [D2:place_of_death, 1:1]  matching list for birth_place

<table>
<thead>
<tr>
<th>Property Pair (matched)</th>
<th>MatchCount</th>
<th>Co-appearanceCount</th>
</tr>
</thead>
<tbody>
<tr>
<td>[D1:doctoalStudent, D2:academic.advisees]</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>[D1:birth_place, D2:place_of_birth]</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Selection using $\alpha = 0.5$ and $k = 2$
Parallel computation (Map-Reduce implementation)

- Generating candidate matches can be done for each instance independently. Hence, we implemented the algorithm in Hadoop 1.0.3 framework.
- Generating candidate matches for instances is distributed among mappers and each mapper outputs $\mu$ and $\lambda$ to the reducer for property pairs.
  - Map Phase
    - Let the number of subject instances in dataset $D_1$ be $X$ and namespace of dataset $D_2$ be $ns$. For each subject $i \in X$, start a mapper job for $\text{GenerateCandidateMatches}(i, ns)$.
    - Each mapper outputs (key,value) pairs as $(p:q, \mu(p,q):\lambda(p,q))$. $p \in D_1$ and $q \in D_2$.
- The reducer collects all $\mu$ and $\lambda$ values and aggregate them for final analysis.
  - Reduce phase
    - Collects output from mappers and aggregates $\mu(p,q)$ and $\lambda(p,q)$ for each key $p:q$.
- The map reduce version on a 14 node cluster was able to achieve a speed up of 833% compared to the desktop version.
Roadmap

- Background
- Statistical Equivalence of properties
- Evaluation
- Discussion, interesting facts, and future directions
- Conclusion
Objectives of the evaluation

- Show the effectiveness of the approach in linked datasets
- Compare with existing aligning techniques

We selected 5000 instance samples from DBpedia, Freebase, LinkedMDB, DBLP L3S, and DBLP RKB Explorer datasets.

These datasets have,

- Complete data for instances in different viewpoints
- Many inter-links
- Complex properties
Experiment details

- $\alpha = 0.5$ for all experiments (works for LOD) except DBpedia and Freebase movie alignment where it was 0.7.
- $k$ was set as 14, 6, 2, 2, and 2 respectively for Person, Film and Software between DBpedia and Freebase, Film between LinkedMDB and DBpedia, and article between DBLP datasets.
- $k$ can be estimated using the data as follows,
  - Set $\alpha = 0.5$ and $k = 2$ (lowest positive values).
  - Get exact matching property (property names) pairs not identified by the algorithm and their $\mu$.
  - Get the average of those $\mu$ values.

- 0.92 for string similarity algorithms.
- 0.8 for WordNet similarity.
## Alignment results

<table>
<thead>
<tr>
<th>Measure type</th>
<th>DBpedia – Freebase (Person)</th>
<th>DBpedia – Freebase (Film)</th>
<th>DBpedia – Freebase (Software)</th>
<th>DBpedia – LinkedMDB (Film)</th>
<th>DBLP_RKB – DBLP_L3S (Article)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extension Based Algorithm</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>0.8758</td>
<td>0.9737</td>
<td>0.6478</td>
<td>0.7560</td>
<td>1.0000</td>
<td>0.8427</td>
</tr>
<tr>
<td>Recall</td>
<td>0.8089*</td>
<td>0.5138</td>
<td>0.4339</td>
<td>0.8157</td>
<td>1.0000</td>
<td>0.7145</td>
</tr>
<tr>
<td>F measure</td>
<td>0.8410*</td>
<td>0.6727</td>
<td>0.5197</td>
<td>0.7848</td>
<td>1.0000</td>
<td>0.7656</td>
</tr>
<tr>
<td><strong>WordNet Similarity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>0.5200</td>
<td>0.8620</td>
<td>0.7619</td>
<td>0.8823</td>
<td>1.0000</td>
<td>0.8052</td>
</tr>
<tr>
<td>Recall</td>
<td>0.4140*</td>
<td>0.3472</td>
<td>0.3018</td>
<td>0.3947</td>
<td>0.3333</td>
<td>0.3582</td>
</tr>
<tr>
<td>F measure</td>
<td>0.4609*</td>
<td>0.4950</td>
<td>0.4324</td>
<td>0.5454</td>
<td>0.5000</td>
<td>0.4867</td>
</tr>
<tr>
<td><strong>Dice Similarity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>0.8064</td>
<td>0.9666</td>
<td>0.7659</td>
<td><strong>1.0000</strong></td>
<td>0.0000</td>
<td>0.7078</td>
</tr>
<tr>
<td>Recall</td>
<td>0.4777*</td>
<td>0.4027</td>
<td>0.3396</td>
<td>0.3421</td>
<td>0.0000</td>
<td>0.3124</td>
</tr>
<tr>
<td>F measure</td>
<td>0.6000*</td>
<td>0.5686</td>
<td>0.4705</td>
<td>0.5098</td>
<td>0.0000</td>
<td>0.4298</td>
</tr>
<tr>
<td><strong>Jaro Similarity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td>0.6774</td>
<td>0.8809</td>
<td><strong>0.7755</strong></td>
<td>0.9411</td>
<td>0.0000</td>
<td>0.6550</td>
</tr>
<tr>
<td>Recall</td>
<td>0.5350*</td>
<td><strong>0.5138</strong></td>
<td>0.3584</td>
<td>0.4210</td>
<td>0.0000</td>
<td>0.3656</td>
</tr>
<tr>
<td>F measure</td>
<td>0.5978*</td>
<td>0.6491</td>
<td>0.4903</td>
<td>0.5818</td>
<td>0.0000</td>
<td>0.4638</td>
</tr>
</tbody>
</table>

* Marks estimated values for experiment 1 because of very large comparisons to check manually. Boldface marks highest result for each experiment.
### Example identifications

<table>
<thead>
<tr>
<th>Property pair types</th>
<th>Dataset 1 (DBpedia)</th>
<th>Dataset 2 (Freebase)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple string similarity matches</td>
<td>db:nationality</td>
<td>fb:nationality</td>
</tr>
<tr>
<td></td>
<td>db:religion</td>
<td>fb:religion</td>
</tr>
<tr>
<td>Synonymous matches</td>
<td>db:occupation</td>
<td>fb:profession</td>
</tr>
<tr>
<td></td>
<td>db:battles</td>
<td>fb:participated_in_conflicts</td>
</tr>
<tr>
<td>Complex matches</td>
<td>db:screenplay</td>
<td>fb:written_by</td>
</tr>
<tr>
<td></td>
<td>db:doctoralStudent</td>
<td>fb:advisees</td>
</tr>
</tbody>
</table>

WordNet similarity failed to identify any of these
Roadmap

- Background
- Statistical Equivalence of properties
- Evaluation
- Discussion, interesting facts, and future directions
- Conclusion
Our experiment covered multi-domain to multi-domain, multi-domain to specific domain and specific-domain to specific-domain dataset property alignment.

In every experiment, the extension based algorithm outperformed others (F measure). F measure gain is in the range of 57% to 78%.

Some properties that are identified are intentionally different, e.g., db:distributor vs fb:production_companies.

– This is because many companies produce and also distribute their films.

Some identified pairs are incorrect due to errors in data modeling.

– For example, db:issue and fb:children.

owl:sameAs linking issues in LOD (not linking exact same thing), e.g., linking London and Greater London.

– We believe few misused links wont affect the algorithm as it decides on a match after analyzing many matches for a pair.
Less number of interlinks.
   - Evolve over time.
   - Look for possible other types of ECR links (i.e., rdf:seeAlso).

Properties do not have uniform distribution in a dataset.
   - Hence, some properties do not have enough matches or appearances.
   - This is due to rare classes and domains they belong to.
   - We can run the algorithm on instances that these less frequent properties appear iteratively.

Current limitations,
   - Requires ECR links
   - Requires overlapping datasets
   - Object-type properties
   - Inability to identify property – sub property relationships
Roadmap

- Background
- Statistical Equivalence of properties
- Evaluation
- Discussion, interesting facts, and future directions
- Conclusion
We approximate owl:equivalentProperty using Statistical Equivalence of properties by analyzing property extensions, which is schema independent.

This novel extension based approach works well with interlinked datasets.

The extension based approach outperforms syntax or dictionary based approaches. F measure gain in the range of 57% - 78%.

It requires many comparisons, but can be easily parallelized evidenced by our Map-Reduce implementation.
Questions?

Thank You

http://knoesis.wright.edu/researchers/kalpa
kalpa@knoesis.org

Kno.e.sis – Ohio Center of Excellence in Knowledge-enabled Computing
Wright State University, Dayton, Ohio, USA