TermPicker: Enabling the Reuse of Vocabulary Terms by Exploiting Data from the Linked Open Data Cloud

Johann Schaible, Thomas Gottron, and Ansgar Scherp.
at ESWC 2016
Problem statement

- When modeling LOD, it is accustomed to reuse vocabulary terms (⇒ classes and properties)
- However, it is a challenging task

Need for Vocabulary Term Recommendations

- dc:creator
- foaf:maker
- dcterms:creator
- swrc:author

http://ex.com/001
http://ex.com/002
http://ex.com/003

swrc:Publication

⇒

swrc:Person

http://ex.com/p/001
http://ex.com/p/002
http://ex.com/p/003
Term recommendations based on...?

- `rdfs:domain`, `rdfs:range`, and other information encoded in vocabularies?
- Popularity of a vocabulary term?
- Classes and properties from domain specific vocabularies?
- etc.

Which vocabulary terms did other data providers on the LOD cloud use *in a similar scenario*?
How to capture a *Scenario*?

- A scenario is defined by vocabulary terms used for a part of a model → patterns on schema level

**Schema-Level Patterns (SLPs)**
A tuple describing the connection between two sets of classes via a set of properties

Example:

$$slp = (\{\text{swrc:Publication}\}, \{\text{dc:creator}\}, \{\text{foaf:Person}\})$$

Resources of type *swrc:Publication* are connected to resources of type *foaf:Person* via the property *dc:creator*

In General:

$$slp = (sts, ps, ots)$$
Vocabulary Term Recommendations Based on LOD

Recommender of vocabulary terms: \( \{x_1, \ldots, x_n\} \)

Feature Computation

\( \{F(slp_q, x_1), \ldots, F(slp_q, x_n)\} \)

Ranking Model

\( g(\{F(slp_q, x_1), \ldots, F(slp_q, x_n)\}) \)

query input

\( slp_q = (\{\text{mo: SoloMusicArtist}\}, \emptyset, \emptyset) \)

query output

Classes for subject: <..., mo: MusicArtist, foaf: Person,...>
properties: <..., foaf: made,..., mo: member_of,...>
Classes for object: <..., mo: Record, mo: MusicGroup,...>
Overview

Recommender of vocabulary terms: \( \{x_1, \ldots, x_n\} \)

Feature Computation
\( \{F(slp_q, x_1), \ldots, F(slp_q, x_n)\} \)

Ranking Model
\( \varrho(\{F(slp_q, x_1), \ldots, F(slp_q, x_n)\}) \)

Classes for subject: \(<..., \text{mo:MusicArtist}, \text{foaf:Person},...>\>

Properties: \(<..., \text{foaf:made},..., \text{mo:member_of},...>\>

Classes for object: \(<..., \text{mo:Record}, \text{mo:MusicGroup},...>\>
Feature Computation: The SLP-Feature

\[ SLP_{LOD} = \text{SPLs computed from datasets on the LOD cloud} \]

\[ SLP_{LOD} = \{(\{\text{mo:SoloMusicArtist, mo:MusicArtist}\}, \{\text{mo:member_of}\}, \{\text{mo:MusicBand}\}) \]
\[ (\{\text{mo:SoloMusicArtist, dbo:Actor}\}, \{\text{foaf:made, mo:recorded}\}, \{\text{mo:Record}\}) \]
\[ (\{\text{foaf:Person}\}, \{\text{foaf:knows}\}, \{\text{foaf:Person}\}) \} \]

\[ slp_q = (\{\text{mo:SoloMusicArtist}\}, \{\}, \{\}) \]

If \( slp_q \subseteq slp_i \) (\( slp_i \in SLP_{LOD} \))

Then Sets of recommendations: \( slp_i - slp_q \)

Classes for subject: \(<\text{ mo:MusicArtist, dbo:Actor }>\)

Properties: \(<\text{ mo:member_of, foaf:made, mo:recorded }>\)

Classes for object: \(<\text{ mo:MusicBand, mo:Record }>\)
# Feature Computation: State of the Art Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Definition of the Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>Number of datasets on the LOD cloud using the recommendation candidate $x$</td>
</tr>
<tr>
<td>$f_2$</td>
<td>Number of datasets on the LOD cloud using the vocabulary $V_x$ of recommendation candidate $x$</td>
</tr>
<tr>
<td>$f_3$</td>
<td>Total number of occurrences of recommendation candidate $x$ on the LOD cloud</td>
</tr>
<tr>
<td>$f_4$</td>
<td>Whether the recommendation candidate $x$ is from a vocabulary that is already used in query-SLP $slp_q$</td>
</tr>
</tbody>
</table>

$f_1 - f_3$: Reusing **popular** vocabularies/vocabulary terms

$f_4$: Reusing vocabulary terms from the **same vocabulary**

1) Schaible, Gottron, and Scherp: Survey on Common Strategies of Vocabulary Reuse in Linked Open Data Modeling (ESWC 2104)
Overview

Recommender of vocabulary terms: \( \{x_1, \ldots, x_n\} \)

Classes for subject: <..., mo:MusicArtist, foaf:Person,...>

Properties: <..., foaf:made,..., mo:member_of,...>

Classes for object: <..., mo:Record, mo:MusicGroup,...>

Feature Computation
\[
\{F(sl_{pq}, x_1), \ldots, F(sl_{pq}, x_n)\}
\]

Ranking Model
\[
\rho(\{F(sl_{pq}, x_1), \ldots, F(sl_{pq}, x_n)\})
\]

query input

query-SLP: \( sl_{pq} = (\{mo:SolomusicArtist\}, \emptyset, \emptyset) \)

query output

Classes for subject: <..., mo:MusicArtist, foaf:Person,...>

Properties: <..., foaf:made,..., mo:member_of,...>

Classes for object: <..., mo:Record, mo:MusicGroup,...>
Calculating a Ranking Model

### How to weight the feature values?

<table>
<thead>
<tr>
<th>$F$</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
<th>SLP-feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(slp_q, x_1)$</td>
<td>7</td>
<td>9</td>
<td>20</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$(slp_q, x_2)$</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>$(slp_q, x_3)$</td>
<td>10</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$(slp_q, x_4)$</td>
<td>4</td>
<td>20</td>
<td>29</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

- **Learning to Rank (L2R):**
  - Family of supervised machine learning algorithms based on data with relevance annotations
  - State of the art in IR to compute a generalized ranking model over a given set of features
  - Ranking model is derived by observing correlations between feature values and candidate relevance
Overview

Recommender of vocabulary terms: \( \{x_1, \ldots, x_n\} \)

Feature Computation
\( \{F(slp_q, x_1), \ldots, F(slp_q, x_n)\} \)

\( slp_q = (\{\text{mo:SoloMusicArtist}\}, \emptyset, \emptyset) \)

Classes for subject: <..., mo:MusicArtist, foaf:Person,...>

properties: <..., foaf:made,..., mo:member_of,...>

Clases for object: <..., mo:Record, mo:MusicGroup,...>

\( q(\{F(slp_q, x_1), \ldots, F(slp_q, x_n)\}) \)
Evaluation

What is the benefit of the SLP-feature?

- Baseline *POP*: Reuse popular vocabulary terms
  - Based on features: $f_1 - f_3$

- Baseline *SAME*: Reuse terms from same vocabulary
  - Based on features: $f_1 - f_4$

- SLP-feature-based: Utilizing the SLP-feature
  - Based on features: $f_1 - f_4$ ( + SLP-feature )
Evaluation Procedure

- Offline evaluation with hidden information

Example:

\[ slp_q = \{(\text{mo:SoloMusicArtist}), \{(\text{foaf:made})\}, \{(\text{mo:Record})\} \]  
Randomly hidden term: “foaf:made”  
Result list: \(< \text{foaf:name}, \text{mo:remixed}, \text{foaf:made}, \ldots >\]

- Measuring quality of recommendations
  - Mean Average Precision (MAP)
  - Mean Reciprocal Rank at the first 5 position (MRR@5)

- Use of the RankLib\(^2\) library

2) [https://sourceforge.net/p/lemur/wiki/RankLib/](https://sourceforge.net/p/lemur/wiki/RankLib/)
Evaluation Data for Recommendations

- Two evaluations based on BTC 2014\(^3\) and DyLDO\(^4\)

<table>
<thead>
<tr>
<th></th>
<th>BTC 2014</th>
<th>DyLDO</th>
</tr>
</thead>
<tbody>
<tr>
<td># of triples</td>
<td>first 34 mio. (reduce overhead)</td>
<td>10.8 mio</td>
</tr>
<tr>
<td># of PLDs</td>
<td>3,500</td>
<td>382</td>
</tr>
<tr>
<td># of distinct terms</td>
<td>5.5 mio.</td>
<td>2.3 mio.</td>
</tr>
<tr>
<td># of vocabularies</td>
<td>1,500</td>
<td>600</td>
</tr>
<tr>
<td># of computed SLPs</td>
<td>227,000</td>
<td>118,000</td>
</tr>
</tbody>
</table>

- 10-fold leave-one-out validation based on PLDs

---

3) [http://km.aifb.kit.edu/projects/btc-2014/](http://km.aifb.kit.edu/projects/btc-2014/)
4) [http://swse.deri.org/dyldo/](http://swse.deri.org/dyldo/)
Results – Box Plots MAP

Features used

L2R Algorithm
Discussion

- Using “from same vocabulary”-feature not significant
  - only in few cases terms from same vocabulary are used
- Using SLPs significant improvement (ca. 35% in MAP)
  - already now: looking at how others model their data
- Better performance on BTC 2014
  - More data in BTC 2014 to train the ranking model
  - 37% more relevant candidates
Conclusion

- Using SLPs, relevant recommendations are ranked significantly higher in the result list
  - Can aid the engineer even more in modeling data in a way how other data providers do
- Using L2R, the more relevant candidates correlate with a feature, the better the results

However,
Offline evaluations do not observe actual user behavior → online evaluation needed
Thank You!

Tool URL: http://termpicker.lodrec.org
Evaluation data and raw results: https://github.com/WanjaSchaible/l2r_eval_material
Feature Computation: Infer SLPs from LOD

- Iterate over RDF Triple data
  - If RDF triple contains type information, store class
  - Else, store property between resources

- Iterate over Property Map and build SLPs

<table>
<thead>
<tr>
<th>Class Map</th>
<th>Resource</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r_1$</td>
<td>${c_1}$</td>
</tr>
<tr>
<td></td>
<td>$r_2$</td>
<td>${c_2, c_4}$</td>
</tr>
<tr>
<td></td>
<td>$r_3$</td>
<td>${c_3}$</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property Map</th>
<th>Resource Pair</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(r_1, r_2)$</td>
<td>${p_1, p_2}$</td>
</tr>
<tr>
<td></td>
<td>$(r_1, r_3)$</td>
<td>${p_3}$</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$$slp(r_1, r_2) = (\{c_1\}, \{p_1, p_2\}, \{c_2, c_4\})$$
$$slp(r_1, r_3) = (\{c_1\}, \{p_3\}, \{c_3\})$$

SLPs computed from LOD cloud:
$$\text{SLP}_{LOD} = \{slp_1, slp_2, ..., slp_n\}$$
Using Learning To Rank

- Training Data is a set of query-SLPs with relevance information on recommendation candidates

\[ s l p_q = (m o : S o l o M u s i c A r t i s t , \emptyset , \emptyset ) \quad \text{Relevant} \quad \{f o a f : m a d e , m o : m e m b e r _ o f\} \]

- L2r algorithms iterate over training data, such that relevant candidates appear at the top of the list
- Assumption: Derived ranking model is as good for new and previously unknown queries
## Results – MAP values

<table>
<thead>
<tr>
<th>Data set</th>
<th>Features</th>
<th>sts</th>
<th>ps</th>
<th>ots</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTC</td>
<td>POP</td>
<td>.32 (.20)</td>
<td>.40 (.21)</td>
<td>.26 (.12)</td>
<td>.28 (.12)</td>
</tr>
<tr>
<td></td>
<td>SAME</td>
<td>.52 (.16)</td>
<td>.56 (.15)</td>
<td>.37 (.14)</td>
<td>.39 (.14)</td>
</tr>
<tr>
<td></td>
<td>SLP</td>
<td>.72 (.11)</td>
<td>.80 (.10)</td>
<td>.75 (.10)</td>
<td>.77 (.10)</td>
</tr>
<tr>
<td>DyLDO</td>
<td>POP</td>
<td>.44 (.29)</td>
<td>.55 (.31)</td>
<td>.35 (.28)</td>
<td>.36 (.28)</td>
</tr>
<tr>
<td></td>
<td>SAME</td>
<td>.59 (.27)</td>
<td>.65 (.24)</td>
<td>.46 (.24)</td>
<td>.46 (.24)</td>
</tr>
<tr>
<td></td>
<td>SLP</td>
<td>.65 (.26)</td>
<td>.70 (.24)</td>
<td>.63 (.25)</td>
<td>.63 (.24)</td>
</tr>
</tbody>
</table>
Evaluation Data

- 10-fold leave one out-validation
- Selected based on
  - (C1): high number of distinct vocabulary terms
  - (C2): high ration b/w reused and all vocabulary terms

<table>
<thead>
<tr>
<th>DyLDO</th>
<th>BTC 2014</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PLD</td>
</tr>
<tr>
<td>kasei.us</td>
<td>100</td>
</tr>
<tr>
<td>thefigtrees.net</td>
<td>89</td>
</tr>
<tr>
<td>bblfish.net</td>
<td>82</td>
</tr>
<tr>
<td>wiker.org</td>
<td>96</td>
</tr>
<tr>
<td>bl.uk</td>
<td>102</td>
</tr>
<tr>
<td>kanzaki.com</td>
<td>176</td>
</tr>
<tr>
<td>taxonconcept.org</td>
<td>139</td>
</tr>
<tr>
<td>fundacionctic.org</td>
<td>110</td>
</tr>
<tr>
<td>data.gov.uk</td>
<td>258</td>
</tr>
<tr>
<td>bbc.co.uk</td>
<td>146</td>
</tr>
</tbody>
</table>