Efficient Multidimensional Blocking for Link Discovery without losing Recall

- Robert Isele, Freie Universität Berlin
- Anja Jentzsch, Freie Universität Berlin
- Christian Bizer, Freie Universität Berlin
Outline

- Problem Statement
- MultiBlock
- Silk Link Discovery Framework
- Evaluation
Problem

- The Web of Data is a single global data space because data sources are connected by links
- 28 billion triples published as Linked Open Data and growing
- **But:**
  - Less than 400 million links
  - Most publishers only link to one other dataset

http://lod-cloud.net/state/
Interlinking Data Sources

- Tools enable data publishers to set links
- Most tools generate links based on user-defined link specifications
- A link specification typically aggregates several different similarity measures.
- Naive solution: Evaluate the link specification for the complete cartesian product
  - Not feasible for large datasets
- Idea: Dismiss definitive non-matches prior to comparison
**Requirements**

- Linked Data is heterogenous and uses a variety of datatypes
  
  **Should be flexible**

- We don’t want to lose recall
  
  **No false dismissals**

- Some similarity measures are no metrics (e.g. Jaro-Winkler)
  
  **Support non-metric similarity measures**

- Linked Data application architectures usually want to integrate a incoming stream of entities (e.g. Silk Server)
  
  **Should not require any pre- or postprocessing**
## Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Lossless</th>
<th>Non-Metrics</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Blocking</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Sorted-Neighborhood</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sorted Blocks</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>FastMap</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MetricMap</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SparseMap</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>StringMap</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Modified SparseMap</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MultiBlock</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
MultiBlock

- Map all entities to a multidimensional space preserving the distances of the entities
Approach

1. General framework
   - Defines the workflow of the indexing
   - Does not define a specific similarity measure or aggregation
   - Instead: Defines the properties a similarity measure/aggregation must adhere to

2. Specific similarity measures and aggregations
   - Can be plugged into the general framework
Indexing Workflow

Index Generation
- Build an index for each similarity measure in the link specification
- Idea: Preserve the distances of the entities

Index Aggregation
- Aggregate all indexes into one compound multidimensional index

Comparison Pair Generation
- Generate a comparison pair for each two entities which share an index
Index Generation

- Generate an (multidimensional) index for each similarity measure

- Basic function of a similarity measure:

$$sim_s : A \times B \rightarrow [0, 1]$$

- Indexing function:

$$index_s : (A \cup B) \times [0, 1] \rightarrow \mathcal{P}(\mathbb{N}^n)$$

- All similarity measures must adhere to:

$$sim_s(e_1, e_2) \leq \theta \iff index_s(e_1) \cap index_s(e_2) \neq \emptyset$$

(For all entities $e_1, e_2$ and a threshold $\theta$ )
Index aggregation

- Aggregate all indices into a single compound index

\[ \text{aggIndex}_a : \mathcal{P}(\mathbb{N}^n) \times \mathcal{P}(\mathbb{N}^n) \rightarrow \mathcal{P}(\mathbb{N}^n) \]

- Example:
Comparison pair generation

- Generate pairs based on overlapping blocks
- Two entities which share the same block will be compared:

\[ \{(a, b); \, i_a = i_b, \, i_a \in index(a), \, i_b \in index(b), \, a \in A, \, b \in B\} \]
The Silk Link Discovery Framework

- Open source link discovery framework, running on all major platforms
- Flexible, declarative language for specifying link conditions
- Scalability and high performance through efficient data handling
  - Reduction of network load by caching and reusing of SPARQL result sets
  - Multi-threaded computation of the data item comparisons
  - Blocking of data items using MultiBlock
**Link Conditions**

- Specify which conditions two entities must fulfill in order to be interlinked.

- A link condition is expressed as a combination of:

  - **RDF paths**
    - Similar to SPARQL 1.1 Property Paths
    - Examples:
      - ?movie/dbpedia:director/rdfs:label
      - ?person/label[@lang='en']

  - **Transformations**
    - Transforms the result set of an RDF paths
    - Variety of built-in transformations
    - Examples:
      -LowerCase
      -RegexReplace
      -Stem

  - **Similarity Metrics**
    - Similarity of two inputs based on a user-defined metric.
    - Examples:
      - Various string similarity metrics
      - Geographic similarity
      - Date similarity

  - **Aggregations**
    - Aggregates multiple similarity metrics
    - Examples:
      - Min, Max, Average
      - Quadratic Mean
      - Geometric Mean
<LinkCondition>
  <Aggregate type="average">
    <Compare metric="levenshtein">
      <Input path="?a/rdfs:label[@lang='en']"/>
      <Input path="?b/rdfs:label[@lang='en']"/>
      <Param name="maxDistance" value="10"/>
    </Compare>
    <Compare metric="wgs84" required="true">
      <Input path="?a/wgs84:geometry"/>
      <Input path="?b/wgs84:geometry"/>
      <Param name="unit" value="km"/>
      <Param name="threshold" value="50"/>
    </Compare>
  </Aggregate>
</LinkCondition>
Silk Versions

- **Silk Single Machine**
  - Generate links on a single machine
  - Local or remote data sets

- **Silk MapReduce (from Silk 2.2)**
  - Generate RDF links using a cluster of multiple machines
  - Based on Hadoop (Can be run on Amazon Elastic MapReduce)

- **Silk Server (from Silk 2.1)**
  - Provides an HTTP API for matching instances from an incoming stream of RDF data while keeping track of known entities
  - Can be used as an identity resolution component within applications that consume Linked Data from the Web
  - Can be used for instance together with a Linked Data crawler to populate a local duplicate-free cache with data from the Web
Silk Linking Engine

- **Loading**
  - Loads the data from the configured data sources
  - Supported sources: SPARQL endpoints. Planned: RDF dumps

- **MultiBlock**
  -Indexes the instances. Only instances with the same index will be matched.
  -This avoids matching the complete Cartesian product.

- **Matching**
  -Computes a similarity value for each pair of instances from the same cluster.
  -The similarity value is based on a user-defined link condition.

- **Filtering**
  -Removes all links with a lower confidence than the user-defined threshold
  -Only a limited number of links from the same subject are yielded

- **Output**
  - Writes the generated links to a user-defined destination
  - Supported formats: N-Triples, OAEI Alignments, Planned: EDOAL Alignments
Performance Evaluation

- Interlinking 204,109 settlements from DBpedia and 530,606 settlements from LinkedGeoData

- Compared MultiBlock with traditional blocking by labels with an overlapping factor of 0.2

* 3GHz Intel(R) Core i7 CPU with 4 core and 8GB of RAM.

<table>
<thead>
<tr>
<th>Method</th>
<th>Comparisons</th>
<th>Runtime (*)</th>
<th>Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Evaluation</td>
<td>108,301,460,054</td>
<td>305,188s</td>
<td>70,037</td>
</tr>
<tr>
<td>Blocking, 100 blocks</td>
<td>3,349,755,846</td>
<td>22,453s</td>
<td>69,403</td>
</tr>
<tr>
<td>Blocking, 1000 blocks</td>
<td>1,049,015,356</td>
<td>7,909s</td>
<td>60,025</td>
</tr>
<tr>
<td>MultiBlock</td>
<td>37,667,462</td>
<td>420s</td>
<td>70,037</td>
</tr>
</tbody>
</table>

- MultiBlock reduces the number of comparisons by a factor of 2,875 and is over 700 times faster than the full evaluation
Conclusion

- MultiBlock uses a multidimensional index to increase its efficiency significantly
- It guarantees that no false dismissals can occur
- It does not require the similarity space to form a metric space
- MultiBlock has been implemented as part of the Silk Link Discovery Framework
- Speedup factor of several 100 for large datasets compared to the full evaluation without losing recall
Get Silk from: http://www4.wiwiss.fu-berlin.de/bizer/silk

This work was supported in part by Vulcan Inc. as part of its Project Halo (www.projecthalo.com) and by the EU FP7 project LOD2 - Creating Knowledge out of Interlinked Data (http://lod2.eu/, Ref. No. 257943).