**Anja Jentzsch**
mail@anjajentzsch.de

**Christian Bizer**
chris@bizer.de

**Robert Isele**
mail@robertisele.com

Freie Universität Berlin

# Silk - Generating RDF Links while Publishing or Consuming Linked Data

The Silk Framework is a tool for discovering relationships between data items within different Linked Data sources. Silk Single Machine and Silk MapReduce can be used by data publishers to generate links between data sets. Silk Server can be used by Linked Data consumers as an identity resolution component in order to augment Web data with additional RDF links.

## Features

- Flexible, declarative language for specifying link conditions
- Support of RDF link generation (owl:sameAs links as well as other types)
- Employment in distributed environments (by accessing local and remote SPARQL endpoints)
- Usable in situations where terms from different vocabularies are mixed and no consistent RDFS or OWL schemata exist
- 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
- Optional blocking of data items

## Silk Versions

**Silk Single Machine** is used to generate RDF links on a single machine. The datasets that should be interlinked can either reside on the same machine or on remote machines which are accessed via the SPARQL protocol. Silk Single Machine provides multithreading and caching.

**Silk MapReduce** is used to generate RDF links between data sets using a cluster of multiple machines. Silk MapReduce is based on Hadoop and can for instance be run on Amazon Elastic MapReduce. Silk MapReduce enables Silk to scale out to very big datasets.

**Silk Server** provides an HTTP API for matching instances from an incoming stream of RDF data while keeping track of known entities. It can be used together with a Linked Data crawler to populate a local duplicate-free cache with data from the Web.

## Workflow

**1. Blocking**

Partitions the instances into clusters. Only instances from the same cluster will be matched. This avoids matching the complete cartesian product.

**2. 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.

**3. Filtering**

Filters the incoming links in two stages:

1. All links with a lower confidence than the user-defined threshold are removed.
2. Only a limited number of links from the same subject are yielded

## Link Conditions

**Comparison Operators**

- Jaro
- Jaro-Winkler
- Levenshtein
- Q-Grams
- Numeric distance
- Date similarity
- Geographical distance

**Aggregation Operators**

- Average
- Maximum
- Minimum
- Quadratic mean (Euclidian dist.)
- Geometric mean

**Example Linking Cities in DBpedia and LinkedGeoData**

```
<LinkCondition>
 <Aggregate type="average">
  <Compare metric="jaro">
   <Input path="?a/rdfs:label[@lang='en']"/>
   <Input path="?b/rdfs:label[@lang='en']"/>
  </Compare>
  <Aggregate type="max" required="true">
   <Compare metric="wgs84">
    <Input path="?a/georss:point"/>
    <Input path="?b/georss:point"/>
    <Param name="unit" value="km"/>
    <Param name="threshold" value="50"/>
   </Compare>
  </Aggregate>
 </Aggregate>
</LinkCondition>
```

## Usage examples

### Processing big datasets

Finding links between cities in a dataset consisting of 10,500 settlements from DBpedia and 59,000 cities and towns from LinkedGeoData.

| Silk Version | Link Generation Time | Number of Links |
|---|---|---|
| *Without Blocking* (resulting in over 6 billion instance comparisons) | | |
| Silk Single Machine[1] | 54 hours | 9,283 |
| Silk MapReduce[2] | 6.7 hours | 9,283 |
| *With Blocking* (cities blocked by name using 50 blocks) | | |
| Silk Single Machine[1] | 155.5 minutes | 9,224 |
| Silk MapReduce[2] | 14.4 minutes | 9,224 |

[1] running on a Intel Core2Duo E8500 with 8GB of RAM
[2] running on Amazon Elastic MapReduce cluster consisting of 10 Amazon EC2 instances (High-CPU Medium Instance Profile)

### Generating links from a data stream

Identifying duplicate person descriptions from a stream of FOAF profiles using Silk Server. Linking persons from the Semantic Web Dog Food corpus with a stream of FOAF profiles generated by LDSpider.

| | |
|---|---|
| Persons in Semantic Web Dog Food | 3739 |
| Existing links to FOAF profiles | 56 |
| Links generated by Silk Server | 228 |

### Acknowledgments

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 (Grant No. 257943, http://lod2.eu/).

VULCAN A Paul G. Allen Company

LOD2 Creating Knowledge out of Interlinked Data

Further information and download: **http://www4.wiwiss.fu-berlin.de/bizer/silk**