Entity Resolution on Web Knowledge Graphs

Mayank Kejriwal
About me
We are moving from a Web of Linked ‘Documents’...
...to a Web of Linked ‘Data’

- ‘Linked Open Data’ started in 2007 with just 12 RDF datasets
- By mid-2010s, contained:
  - Millions of resources
  - 1000 datasets
  - 900,000 documents
  - 500 million inter-dataset links
  - Many domains!
- Applications include schema.org, Google Knowledge Graph, the Constitute Project...

Cyganiak and Jentzsch (2014)
Linkeddata.org
Linked Data

- A set of four best practices for publishing and connecting structured data on the Web

Bizer et al. (2009, 2014)
Resource Description Framework (RDF)

- An **RDF dataset** is a set of triples, visualized as a directed labeled graph.
- A **triple** is a 3-element tuple \((\text{subject}, \text{property}, \text{object})\) and represents an edge in the graph.
  - Subjects and properties are necessarily URIs.
  - Objects may be URIs or literals.

---

http://www.w3.org/RDF
Bizer et al. (2009)
Entity Resolution (ER)

- Connecting pairs of entities that refer to the **same underlying entity**
- Also known as ‘instance matching’, ‘entity matching’, ‘co-reference resolution’, ‘merge-purge’...

---

- Jaffri et al. (2008)
- Papadakis et al. (2010)
- Nikolov et al. (2011)
What’s the vision? A thesaurus for entities called an **Entity Name System (ENS)**

- Populating an ENS requires solutions to ER
- Many applications

Bouquet et al. (2008)
Research question

What *requirements* need to be fulfilled in order to populate a Linked Data Entity Name System?
Returning to our example...
Linked Open Data

- ‘Linked Open Data’ started in 2007 with just a handful of datasets
- At last survey (2014), contains:
  - Millions of resources
  - 1000 datasets
  - 900,000 documents
  - 500 million inter-dataset links
  - Many domains!

Cyganiak and Jentzsch (2014)
Linkeddata.org
Hypothesis

Populating a Linked Data Entity Name System requires simultaneously fulfilling the four DASH requirements of domain-independence, automation, scalability and heterogeneity.
Step 1: Type alignment
Step 2: Property alignment
Step 3: Similarity prediction?
Step 3: blocking and similarity

Apply *blocking key*
e.g. Tokens(LastName)

Dataset 1

Dataset 2

'Exhaustive' set: 4 X 6 = 24 pairs

Generate candidate set (7 pairs), apply *similarity function* on each pair

Christen (2012)
Final output

- freebase:non-profit
- freebase:Firm
- freebase:Microsoft
- freebase:Organisation
- freebase:Company
- dbpedia:Allen_Paul
- dbpedia:Inventor
- dbpedia:Person
- freebase:Co-founder_of
- freebase:bornOn
- "01/21/1953"
Supervised schematic (post type-alignment)

- Presented mainly to **static tabular** datasets; not viable for **dynamic linked** datasets

```
Training set of
duplicates/
non-duplicates

Learn
Property
Alignment

Aligned training set

Learn
blocking key

Blocking
key

Candidate
set

Execute
blocking

RDF dataset 1

RDF dataset 2

Learn Similarity
function

Trained
Classifier

Execute
similarity

:sameAs
links

Elmagarmid et al. (2007)
```
Semi-supervised schematic (post type-alignment)

- Hard to realize in practice both because of **class imbalance**, and because graphs are **hard to explore**

Seed training set of duplicates/non-duplicates

Learn Property Alignment

Aligned training set

Learn blocking key

Blocking key

Execute blocking

Candidate set

Learn Similarity function

Trained Classifier

Execute similarity

Most confident samples

▪ Hard to realize in practice both because of **class imbalance**, and because graphs are **hard to explore**

RDF dataset 1

RDF dataset 2

:.sameAs links

Kejriwal and Miranker (2015)
Unsupervised schematic?

Seed training set of duplicates/non-duplicates

Learn Property Alignment

Aligned training set

Learn blocking key

Blocking key

Candidate set

Execute blocking

RDF dataset 1

RDF dataset 2

Learn Similarity function

Trained Classifier

Candidate set

Execute similarity

Most confident samples

:sameAs links
Unsupervised schematic?

- Learn Property Alignment
- Learn blocking key
- Learn Similarity function
- Execute blocking
- Execute similarity
- Candidate set
- Blocking key
- Aligned training set
- Most confident samples
- Noisy seed training set of duplicates/ non-duplicates
- Training set generator?

RDF dataset 1
RDF dataset 2

Kejriwal and Miranker (2013-2015)
A complete, unsupervised schematic

- Implemented both **serially** and in **MapReduce** (using standard cloud services)
- Feasible for linking **large, cross-domain graphs** like Dbpedia and Freebase
- Does not ‘**assume away**’ any of the DASH requirements (e.g. property heterogeneity)
Is Entity Resolution Enough?
Let’s think about a complete KG ecosystem
Avenues for future research

• Fully end-to-end, scalable deployment of a complete KG pipeline (which includes Entity Resolution)
• Visualizing and interactively manipulating KGs
• Long-tail vs. short-tail ER
• Systems-level vs. reductionist understanding of the problem
  • For example, how does noise in information extraction affect performance of ER?
• Easy-to-use open-source software development
• Improving performance and speed!