Learning Arbitrary RDF Dataset Enrichment Graphs Using Pre- & Postcondition Broadcasting

Kevin Dreßler

Institut für Informatik
Universität Leipzig

15.02.2019

Outline

Section 1

Motivation

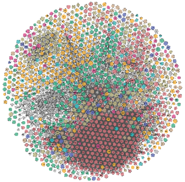
Linked Open Data

- Linked Data is becoming more and more important

[1]Stats taken from LOD Laundromat. Links estimated.

Linked Open Data

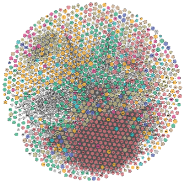
- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$



[1] Stats taken from LOD Laundromat. Links estimated.

Linked Open Data

- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$
- Can we use this vast amount of data as is?



[1] Stats taken from LOD Laundromat. Links estimated.

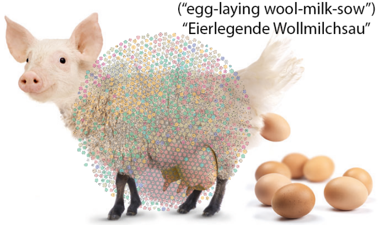
Linked Open Data

- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$
- Can we use this vast amount of data as is?

[1] Stats taken from LOD Laundromat. Links estimated.

Linked Open Data

- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$
- Can we use this vast amount of data as is?
- Not so easy, due to
  - limited resources



[1] Stats taken from LOD Laundromat. Links estimated.

Linked Open Data

- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$
- Can we use this vast amount of data as is?
- Not so easy, due to
  - limited resources
  - unreliable SPARQL endpoints

[1] Stats taken from LOD Laundromat. Links estimated.

Linked Open Data

- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$
- Can we use this vast amount of data as is?
- Not so easy, due to
  - limited resources
  - unreliable SPARQL endpoints
  - missing links

[1] Stats taken from LOD Laundromat. Links estimated.

Linked Open Data

- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$
- Can we use this vast amount of data as is?
- Not so easy, due to
  - limited resources
  - unreliable SPARQL endpoints
  - missing links
  - application-specific ontologies

[1] Stats taken from LOD Laundromat. Links estimated.

Linked Open Data

- Linked Data is becoming more and more important
- Linked Open Data (LOD) Cloud [1]
  - 650k datasets
  - 38 billion triples
  - $n \cdot 10^6$ links $(10 < n < 1000)$
- Can we use this vast amount of data as is?
- Not so easy, due to
  - limited resources
  - unreliable SPARQL endpoints
  - missing links
  - application-specific ontologies
- RDF Dataset Enrichment to the rescue!

[1] Stats taken from LOD Laundromat. Links estimated.

DEER - The RDF Dataset Enrichment Framework

- **RDF Dataset Enrichment:** Adding / Deleting triples from an RDF Dataset

DEER

- **RDF Dataset Enrichment:** Adding / Deleting triples from an RDF Dataset

Definition Enrichment Operator

Let $\mathcal{D}$ be the set of all RDF Datasets. We call a function $o : \mathcal{D}^n \to \mathcal{D}^m$ an *Enrichment Operator*. We call n the in degree and m the out degree of o.

Definition **Enrichment Graph**

An Enrichment Graph $G = (V, E, L)$ is a Directed Acyclic Labeled Multigraph.
The root vertices $V_r = v \in V | \nexists u \in V, (u, v) \in E$ of an Enrichment Graph represent input RDF Datasets emitters.
The leaf vertices $V_l = v \in V | \nexists u \in V, (v, u) \in E$ of an Enrichment Graph represent output RDF Datasets acceptors.
The intermediate vertices $V_i = V \setminus V_r \setminus V_l$ represent Enrichment Operators.
The function $O : V \to \mathcal{O}$ maps vertices to the entities they represent.
An edge $(u, v) \in E \subseteq V \times V$ represents flow of data.
The label function $L : E \to 2^{(\mathbb{N}\times\mathbb{N})}$ induces a mapping from the components of images and arguments of represented entities, i.e. for $e = (u, v)$, an entry of the label multiset $l \in L(e) = (i, j)$ establishes a flow of data from the ith component in the image of $O(u)$ to the jth argument of $O(v)$.

Definition Enrichment Graph (Continued)

An Enrichment Graph $G = (V, E, L)$ with $\forall e \in E : |L(e)| = 1$ is called an Enrichment Pipeline or a type 0 Enrichment Graph.
An Enrichment Graph $G = (V, E, L)$ with $\exists e \in E : |L(e)| > 1 \wedge |V_r| = |V_l| = 1$ is called a type 1 Enrichment Graph.
An Enrichment Graph $G = (V, E, L)$ with $\forall e \in E : |L(e)| > 1 \wedge |V_r| > 1 \wedge |V_l| = 1$ is called a type 2 Enrichment Graph.
An Enrichment Graph $G = (V, E, L)$ with $\forall e \in E : |L(e)| > 1 \wedge |V_r| > 1 \wedge |V_l| > 1$ is called a type 3 Enrichment Graph.

Configuring DEER

- Plugin architecture
- Most Enrichment Operators need parametrization
- Defining an Enrichment Graph requires expert knowledge
- $\rightarrow$ Use Machine Learning

Existing ML Algorithm in DEER

- Learns type 1 Enrichment Graphs based on Refinement Operators
- Training data: RDF Datasets Source (S) and Target (T)
- Fitness Function: F_1 score over triples
- Iteratively construct Enrichment Graph G from Set of all Enrichment Operators $\mathcal{O}$
- Set $G := nil$, $B := S$
- Until Termination Criterion met:
  - $\forall o \in \mathcal{O} : o.selfConfig(B, T)$
  - $G := G \circ \arg\max_{F_1(T,B')}\{o' \in \mathcal{O} | B' = (G \circ o')(B)\}$
  - $B := G(B)$

Analysis of Existing ML Algorithm in DEER

- Good theoretical properties (finite, proper, complete, not redundant)
- But several incorrect implicit assumptions
  - F_1 score over triples is indicative for fitness $\lightning$
    - Counterexample: Filtering first
  - Enrichment Operators are independent $\lightning$
    - Counterexample: Linking $\rightarrow$ Dereferencing
  - Training data always contains sufficient information for self-configuration $\lightning$
    - possible, but not realistic
    - at least as hard to satisfy as manual configuration
- $\rightarrow$ Objective of this thesis: develop new approach

Derived Goals

The new Approach should:

1. derive proper fitness function
2. take interdependency of Enrichment Operators into account
3. cope with incomplete information w.r.t. self-configuration
4. be applicable for type 0, 1 & 2 Enrichment Graphs

Section 2

Approach

(1) Fixing the Fitness Function

- Weight recall higher than precision (F_2 score)
- Linear combination of F_2 score over triples, URI resources & literals
- Apply fuzzy matching to URI resources & literals

(2) Leveraging Enrichment Operator Interdependency

- avoid local minima $\rightarrow$ employ genetic programming
- infer possible dependencies using top-down self-configuration
- recursive broadcast algorithm: $broadcast(\mathcal{G}, D_\perp, D_\top)$
  - each o assesses if $D_\top$ is a possible postcondition
  - each o reports a number between 0 and 1 based on its assessment
  - if some termination criteria holds, return $\mathcal{G}$
  - the Enrichment Operators reporting the highest k values are chosen
  - for each chosen Enrichment Operator o_c call $broadcast(\mathcal{G}' = \{G \circ o_c | G \in \mathcal{G}\}, D_\perp, o_c^{-1}(D_\top))$
- started as: $\forall o \in \mathcal{O} : broadcast(\{\}, S, T)$

(3) Reconstructing Hidden Information for Self-Configuration

- top-down self-configuration reconstructs hidden information!

(4) From Type 0 to Type 2

- Type 0: Base implementation
- Type 1: Extends Type 0
  - Allow branching and merging in inital population generation
  - Force merging at leafs
  - Detect independent Enrichment Operators & parallelize
- Type 2: Theory not finished yet

Section 3

Evaluation Setup

Evaluation Setup

- find use cases for each Enrichment Graph type
- write manual configurations to generate training data
- run each experiment ten times
- record precision, recall, run time, logs
- $\rightarrow$ compare with baseline
- compare complexity of manual and learned Enrichment Graphs

Section 4

Roadmap

Roadmap

- Evaluation of Type 0: 22.02.
- Evaluation of Type 1: 01.03.
- Evaluation of Type 2: 15.03.
- Hand in thesis: 29.03.
- Final presentation: 12.04.
- Official due date: 07.05.

Thank You for Your Attention