DEER
Automating RDF Dataset Transformation and Enrichment

Mohamed Ahmed Sherif, Axel-Cyrille Ngonga Ngomo and Jens Lehmann

June 3, 2015
Outline

1. Motivation
2. Approach
3. Evaluation
4. Conclusion and Future Work
1 Motivation

2 Approach

3 Evaluation

4 Conclusion and Future Work
Dataset *DrugBank*

Goal Gather information about companies related to drugs for a market study
**Why RDF Transformation & Enrichment?**

**Dataset** *DrugBank*

**Goal** Gather information about companies related to drugs for a market study

Ibuprofen was extracted by the research arm of **Boots company** during the 1960s ...
Need for enriched datasets

- Tourism
- Question Answering
- Enhanced Reality
- ...

RDF transformation and enrichment

- Triples to be added to the original KB and/or
- Triples to be deleted from the original KB
Manual Knowledge Base Enrichment

Demands for the specification of data enrichment pipelines

- Describe how data is to be integrated (usually manually)

Manual customized enrichment pipelines

- Leads to the expected results
- Time consuming
- Cannot be ported easily to other datasets
Manual Knowledge Base Enrichment

- Demands for the specification of data *enrichment pipelines*
- Describe how data is to be integrated (usually manually)

**Manual customized enrichment pipelines**

- Leads to the expected results
- Time consuming
- Cannot be ported easily to other datasets
Automatic Knowledge Base Enrichment

- **Enrichment pipeline** $M : \mathcal{K} \rightarrow \mathcal{K}$ that maps KB $K$ to an enriched KB $K'$ with $K' = M(K)$.

- $M$ is an ordered list of *atomic enrichment functions* $m \in \mathcal{M}$

$$M = \begin{cases} \phi & \text{if } K = K', \\ (m_1, \ldots, m_n), \text{ where } m_i \in \mathcal{M}, 1 \leq i \leq n & \text{otherwise.} \end{cases}$$

**Research questions**

1. How to create self-configuring atomic enrichment functions $m \in \mathcal{M}$?
2. How to automatically generate an enrichment pipeline $M$?
Automatic Knowledge Base Enrichment

- *Enrichment pipeline* $M : \mathcal{K} \rightarrow \mathcal{K}$ that maps KB $K$ to an enriched KB $K'$ with $K' = M(K)$.
- $M$ is an ordered list of *atomic enrichment functions* $m \in \mathcal{M}$

\[
M = \begin{cases} 
\phi & \text{if } K = K', \\
(m_1, \ldots, m_n), & \text{where } m_i \in \mathcal{M}, 1 \leq i \leq n \text{ otherwise.}
\end{cases}
\]

**Research questions**

1. How to create self-configuring atomic enrichment functions $m \in \mathcal{M}$?
2. How to automatically generate an enrichment pipeline $M$?
Outline

1 Motivation

2 Approach

3 Evaluation

4 Conclusion and Future Work
I. Dereferencing atomic enrichment function

- Datasets are linked (e.g., using owl:sameAs)
- Deferences pre-specified set of predicates
- Adds found predicates to source the dataset
Atomic Enrichment Functions

I. Dereferencing atomic enrichment function

- Datasets are linked (e.g., using owl:sameAs)
- Dererences pre-specified set of predicates
- Adds found predicates to source the dataset

Ibuprofen was extracted by the research arm of Boots company during the 1960s...
Datasets are linked (e.g., using `owl:sameAs`)
- Defeferences pre-specified set of predicates
- Adds found predicates to source the dataset

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...
Datasets are linked (e.g., using owl:sameAs)

- Deferences pre-specified set of predicates
- Adds found predicates to source the dataset

- Ibuprofen was extracted by the research arm of Boots company during the 1960s...
Find the set of predicates $D_p$ from the enriched CBDs that are missing from source CBDs.

### Non-enriched CBD of Ibuprofen

- db:Ibuprofen ➔ owl:sameAs ➔ :Ibuprofen ➔ a ➔ :Drug

### Enriched CBD of Ibuprofen

- db:Ibuprofen ➔ owl:sameAs ➔ :Ibuprofen ➔ a ➔ :Drug
- :BootsCompany ➔ :relatedCompany ➔ rdfs:comment

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...

$$D_p = \{ :relatedCompany, rdfs:comment \}$$
Finds the set of predicates $D_p$ from the enriched CBDs that are missing from source CBDs

Non-enriched CBD of Ibuprofen

Enriched CBD of Ibuprofen

$D_p = \{\text{:relatedCompany}, \text{rdfs:comment}\}$
Finds the set of predicates $D_p$ from the enriched CBDs that are missing from source CBDs.

$D_p = \{\text{:relatedCompany}, \text{rdfs:comment}\}$
Finds the set of predicates $D_p$ from the enriched CBDs that are missing from source CBDs

$D_p = \{ \text{:relatedCompany}, \text{rdfs:comment} \}$
Dereferences $D_p = \{:\text{relatedCompany}, \text{rdfs:comment}\}$

CBD of Ibuprofen

- Finds only rdfs:comment, adds it to the source dataset

Dereferencing enriched CBD of Ibuprofen

Ibuprofen was extracted by the research arm of Boots company during the 1960s...
Dereferences $D_p = \{:\text{relatedCompany}, \text{rdfs:comment}\}$

CBD of Ibuprofen

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...

- Finds only rdfs:comment, adds it to the source dataset
Dereferences $D_p = \{ \text{:relatedCompany}, \text{rdfs:comment} \}$

CBD of Ibuprofen

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...

Finds only rdfs:comment, adds it to the source dataset
Dereferences $D_p = \{:\text{relatedCompany}, \text{rdfs:comment}\}$

CBD of Ibuprofen

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...

Finds only rdfs:comment, adds it to the source dataset

Dereferencing enriched CBD of Ibuprofen

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...
II. NLP atomic enrichment function

- Datatype objects contain unstructured information
- Uses *Named Entity Recognition* to extract implicit data
- Adds extracted entities to the source datasets

Ibuprofen was extracted by the research arm of Boots company during the 1960s...
Datatype objects contain unstructured information

- Uses Named Entity Recognition to extract implicit data
- Adds extracted entities to the source datasets

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...
**II. NLP atomic enrichment function**

- Datatype objects contain unstructured information
- Uses *Named Entity Recognition* to extract implicit data
- Adds extracted entities to the source datasets

**Diagram:**

```
db:Ibuprofen --> owl:sameAs --> :Ibuprofen --> a --> :Drug
```

Ibuprofen was extracted by the research arm of *Boots company* during the 1960s...
- Datatype objects contain unstructured information
- Uses *Named Entity Recognition* to extract implicit data
- Adds extracted entities to the source datasets

Ibuprofen was extracted by the research arm of *Boots company* during the 1960s ...
Extracts all possible named entity types
- Adds extracted entities to the source dataset

NLP enriched CBD of Ibuprofen

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...
NLP enriched CBD of Ibuprofen

- Extracts all possible named entity types
- Adds extracted entities to the source dataset

Ibuprofen was extracted by the research arm of Boots company during the 1960s...
- Extracts all possible named entity types
- Adds extracted entities to the source dataset

NLP enriched CBD of Ibuprofen

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...
Enriched datasets may contain diverse ontologies.

Predicate conformation maps a set of pre-specified predicates to a target ontology.

Ibuprofen was extracted by the research arm of Boots company during the 1960s ...
Enriched datasets may contain diverse ontologies

- Predicate conformation maps a set of a pre-specified predicates to a target ontology

---

Ibuprofen was extracted by the research arm of Boots company during the 1960s...
Enriched datasets may contain diverse ontologies

Predicate conformation maps a set of pre-specified predicates to a target ontology.

Ibuprofen was extracted by the research arm of Boots company during the 1960s...
Self-Configuration
III. Predicate conformation Enrichment Function

- Finds list of predicates $P_s$ and $P_t$ from the source resp. target datasets with the same subject and objects
- Changes each $P_s$ with its respective $P_t$

NLP enriched CBD of Ibuprofen

Enriched CBD of Ibuprofen (positive example target)
Self-Configuration

III. Predicate conformation Enrichment Function

- Finds list of predicates $P_s$ and $P_t$ from the source resp. target datasets with the same subject and objects
- Changes each $P_s$ with its respective $P_t$

NLP enriched CBD of Ibuprofen

```
```

Enriched CBD of Ibuprofen (positive example target)

```
```

Ibuprofen was extracted by the research arm of Boots company during the 1960s...
Self-Configuration

III. Predicate conformation Enrichment Function

- Finds list of predicates $P_s$ and $P_t$ from the source resp. target datasets with the same subject and objects
- Changes each $P_s$ with its respective $P_t$

NLP enriched CBD of Ibuprofen

Enriched CBD of Ibuprofen (positive example target)
## KB Enrichment Refinement Operator

### Input
- Set of atomic enrichment functions $\mathcal{M}$
- Set of positive examples $\mathcal{E}$

### Refinement Operator
\[
\rho(\mathcal{M}) = \bigcup_{m \in \mathcal{M}} \mathcal{M} ++ m \quad (\text{++ is the list append operator})
\]

### Output
- Enrichment pipeline $\mathcal{M}$
<table>
<thead>
<tr>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Set of atomic enrichment functions $\mathcal{M}$</td>
</tr>
<tr>
<td>- Set of positive examples $\mathcal{E}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Refinement Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho(\mathcal{M}) = \bigcup_{\forall m \in \mathcal{M}} \mathcal{M} + + m$</td>
</tr>
<tr>
<td>( $+ +$ is the list append operator )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Enrichment pipeline $\mathcal{M}$</td>
</tr>
</tbody>
</table>
KB Enrichment Refinement Operator

Input
- Set of atomic enrichment functions $\mathcal{M}$
- Set of positive examples $\mathcal{E}$

Refinement Operator
$$\rho(\mathcal{M}) = \bigcup_{m \in \mathcal{M}} \mathcal{M} \mathbin{++} m \quad (\mathbin{++} \text{ is the list append operator})$$

Output
- Enrichment pipeline $\mathcal{M}$
Ibuprofen was extracted by the research arm of Boots company during the 1960s.

**Positive Example**

**Non-enriched CBD of Ibuprofen**

**Enriched CBD of Ibuprofen**
Learning Algorithm

1. Start by empty enrichment pipeline \( M = \bot \)
2. Self-configure all \( m_i \in M \), add as child to \( \bot \)
3. Select most promising node
4. Expand most promising node
Learning Algorithm

1. Start by empty enrichment pipeline $M = \bot$
2. Self-configure all $m_i \in \mathcal{M}$, add as child to $\bot$
3. Select most promising node
4. Expand most promising node
Learning Algorithm

1. Start by empty enrichment pipeline $M = \perp$
2. Self-configure all $m_i \in M$, add as child to $\perp$
3. Select most promising node
4. Expand most promising node
Learning Algorithm

1. Start by empty enrichment pipeline $M = \bot$
2. Self-configure all $m_i \in M$, add as child to $\bot$
3. Select most promising node
4. Expand most promising node

\[
\begin{align*}
(m_1, m_2) & \quad (m_1, m_3) \\
&m_1 \quad (m_2) \quad (m_3) \\
\bot
\end{align*}
\]
Learning Algorithm

1. Start by empty enrichment pipeline $M = \bot$
2. Self-configure all $m_i \in \mathcal{M}$, add as child to $\bot$
3. Select most promising node
4. Expand most promising node
Learning Algorithm

1. Start by empty enrichment pipeline $M = \perp$
2. Self-configure all $m_i \in M$, add as child to $\perp$
3. Select most promising node
4. Expand most promising node
Learning Algorithm

1. Start by empty enrichment pipeline $M = \bot$
2. Self-configure all $m_i \in M$, add as child to $\bot$
3. Select most promising node
4. Expand most promising node
Learning Algorithm

1. Start by empty enrichment pipeline $M = \bot$
2. Self-configure all $m_i \in M$, add as child to $\bot$
3. Select most promising node
4. Expand most promising node
Most Promising Node Selection

Node complexity $c(n)$
- Linear combination of the node's children count and level

Node fitness $f(n)$
- Difference between node's enrichment pipeline F-measure and weighted complexity, $f(n) = F(n) - \omega \cdot c(n)$
- $\omega$ controls the tradeoff between
  - Greedy search ($\omega = 0$)
  - Search strategies closer to breadth-first search ($\omega > 0$).

Most promising node
- The leaf node with the maximum fitness through the whole refinement tree
**Node complexity** $c(n)$
- Linear combination of the node's children count and level

**Node fitness** $f(n)$
- Difference between node’s enrichment pipeline F-measure and weighted complexity, $f(n) = F(n) - \omega \cdot c(n)$
- $\omega$ controls the tradeoff between
  - Greedy search ($\omega = 0$)
  - Search strategies closer to breadth-first search ($\omega > 0$).

**Most promising node**
- The leaf node with the maximum fitness through the whole refinement tree
### Most Promising Node Selection

#### Node complexity $c(n)$
- Linear combination of the node's children count and level

#### Node fitness $f(n)$
- Difference between node's enrichment pipeline F-measure and weighted complexity, $f(n) = F(n) - \omega \cdot c(n)$
- $\omega$ controls the tradeoff between
  - Greedy search ($\omega = 0$)
  - Search strategies closer to breadth-first search ($\omega > 0$).

#### Most promising node
- The leaf node with the maximum fitness through the whole refinement tree
Outline

1. Motivation
2. Approach
3. Evaluation
4. Conclusion and Future Work
**Experimental Setup**

**Datasets**
- 1 manual experimental enrichment pipelines for *Jamendo*
- 2 manual experimental enrichment pipelines for *DrugBank*
- 5 manual experimental enrichment pipelines for *DBpedia*
  (AdministrativeRegion)

**Learning Algorithm**
- 6 atomic enrichment functions
- Termination criterion:
  - Maximum number of iterations of 10
  - Optimal enrichment pipeline found (F-score = 1)
### Experimental Setup

#### Datasets
- 1 manual experimental enrichment pipelines for *Jamendo*
- 2 manual experimental enrichment pipelines for *DrugBank*
- 5 manual experimental enrichment pipelines for *DBpedia* (AdministrativeRegion)

#### Learning Algorithm
- 6 atomic enrichment functions
- Termination criterion:
  - Maximum number of iterations of 10
  - Optimal enrichment pipeline found (F-score = 1)
Node fitness
\[ f(n) = F(n) - \omega . c(n) \]
- \( \omega \) controls the tradeoff between
  - Greedy search (\( \omega = 0 \))
  - Search strategies closer to breadth first search (\( \omega > 0 \)).
- Result: \( \omega = 0.75 \) leads to the best results

\[
\begin{array}{cccc}
\omega & P & R & F \\
0 & 1.0 & 0.99 & 0.99 \\
0.25 & 1.0 & 0.99 & 0.99 \\
0.50 & 1.0 & 0.99 & 0.99 \\
0.75 & 1.0 & 1.0 & 1.0 \\
1.0 & 1.0 & 0.99 & 0.99 \\
\end{array}
\]
### Effect of Positive Examples

<table>
<thead>
<tr>
<th>Manual $M$</th>
<th>Examples count</th>
<th>Size of $M$</th>
<th>Time $M (KB)$</th>
<th>Size of learned $M'$</th>
<th>Time $M' (KB)$</th>
<th>Learn Time</th>
<th>Iterations count</th>
<th>$F$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M^1_{DBpedia}$</td>
<td>1</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
<td>1.6</td>
<td>1.3</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
<td>1.8</td>
<td>1.3</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>$M^2_{DBpedia}$</td>
<td>1</td>
<td>2</td>
<td>23.3</td>
<td>1</td>
<td>0.1</td>
<td>0.2</td>
<td>1</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>15</td>
<td>2</td>
<td>17</td>
<td>0.3</td>
<td>9</td>
<td>0.99</td>
</tr>
<tr>
<td>$M^3_{DBpedia}$</td>
<td>1</td>
<td>3</td>
<td>14.7</td>
<td>3</td>
<td>15.2</td>
<td>6.1</td>
<td>9</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>15</td>
<td>2</td>
<td>15.1</td>
<td>0.1</td>
<td>9</td>
<td>0.99</td>
</tr>
<tr>
<td>$M^4_{DBpedia}$</td>
<td>1</td>
<td>4</td>
<td>0.4</td>
<td>2</td>
<td>0.1</td>
<td>0.7</td>
<td>2</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>0.6</td>
<td>2</td>
<td>0.3</td>
<td>0.9</td>
<td>2</td>
<td>0.99</td>
</tr>
<tr>
<td>$M^5_{DBpedia}$</td>
<td>1</td>
<td>5</td>
<td>22</td>
<td>2</td>
<td>0.1</td>
<td>0.7</td>
<td>2</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5</td>
<td>25.5</td>
<td>2</td>
<td>0.2</td>
<td>0.9</td>
<td>2</td>
<td>1.0</td>
</tr>
<tr>
<td>$M^1_{DrugBank}$</td>
<td>1</td>
<td>2</td>
<td>3.5</td>
<td>1</td>
<td>4.1</td>
<td>0.1</td>
<td>10</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>3.6</td>
<td>1</td>
<td>3.4</td>
<td>0.1</td>
<td>10</td>
<td>0.99</td>
</tr>
<tr>
<td>$M^2_{DrugBank}$</td>
<td>1</td>
<td>3</td>
<td>25.2</td>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>10</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>22.8</td>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>61</td>
<td>0.99</td>
</tr>
<tr>
<td>$M^1_{Jamendo}$</td>
<td>1</td>
<td>1</td>
<td>10.9</td>
<td>2</td>
<td>10.6</td>
<td>0.1</td>
<td>2</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>10.4</td>
<td>2</td>
<td>10.4</td>
<td>0.1</td>
<td>1</td>
<td>0.99</td>
</tr>
</tbody>
</table>
Outline

1. Motivation
2. Approach
3. Evaluation
4. Conclusion and Future Work
Conclusion

- Presented self-configuring atomic enrichment functions
- Presented an approach for learning enrichment pipelines based on a refinement operator
- Showed that our approach can easily reconstruct manually created enrichment pipelines

Future Work

- Parallelize the algorithm on several CPUs as well as load balancing
- Support directed acyclic graphs as enrichment specifications by allowing to split and merge datasets
- Pro-active enrichment strategies and active learning
Conclusion

- Presented self-configuring atomic enrichment functions
- Presented an approach for learning enrichment pipelines based on a refinement operator
- Showed that our approach can easily reconstruct manually created enrichment pipelines

Future Work

- Parallelize the algorithm on several CPUs as well as load balancing
- Support directed acyclic graphs as enrichment specifications by allowing to split and merge datasets
- Pro-active enrichment strategies and active learning
Thank You!

Questions?

Mohamed Sherif
Augustusplatz 10
D-04109 Leipzig
sherif@informatik.uni-leipzig.de
http://aksw.org/MohamedSherif
http://aksw.org/Projects/DEER

#akswgroup