Type inference through the analysis of Wikipedia links

Andrea Giovanni Nuzzolese
nuzzoles@cs.unibo.it
Aldo Gangemi
aldo.gangemi@cnr.it
Valentina Presutti
valentina.presutti@cnr.it
Paolo Ciancarini
ciancarini@cs.unibo.it

16 April 2012 - Lyon, France - LDOW 2012
Outline

• Motivations
• Materials
• Applied methods
• Results
• Conclusions
Motivations

- Only a subset of the DBpedia resources is typed with the DBpedia ontology (DBPO)
- The typing procedure is top-down.
- Is the DBPO complete with respect to the DBpedia domain?
- How good and homogeneous is the granularity of DBPO types?

Resources used in wikilinks relations: 15,944,381

Resources having a DBPO type: 1,518,697
Wikilink triples with typed subject/object: 16,745,830

Wikilinks triples: 107,892,317

DBpedia ontology: 272 classes

DBpedia 3.6

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>wikilink triples</td>
<td>107,892,317</td>
</tr>
<tr>
<td>infobox mapping-based “data” triples</td>
<td>9,357,273</td>
</tr>
<tr>
<td>rdfs:label triples</td>
<td>7,972,225</td>
</tr>
<tr>
<td>rdf:type triples</td>
<td>6,173,940</td>
</tr>
<tr>
<td>infobox mapping-based “object” triples</td>
<td>4,251,239</td>
</tr>
</tbody>
</table>
What we did

• Wikilinks of a DBpedia resource convey knowledge that can be used for classifying it.

• Classification methods
  ♦ Inductive learning: k-Nearest Neighbor algorithm
  ♦ Abductive classification based on EKPs [1] and homotypes used as background knowledge

• The methods were performed on

Resources having a DBPO type: 1,518,697
Resources used in wikilinks relations: 15,944,381
Sample of untyped resources: 1,000

Inductive classification

• We designed two inductive classification experiments based on the $k$-NN algorithm
  ✦ on 272 features, i.e., all the classes in the DBPO
  ✦ on 27 features, i.e., the top-level classes in the DBPO hierarchy

• For each experiment we built a labeled feature space model as training set by using a randomly sampled 20% of typed resources
  ✦ the algorithms were tested on the remaining 80% of typed resources
Building the training set for K-Nearest Neighbor algorithm

<table>
<thead>
<tr>
<th>dbpedia:Steve_Jobs</th>
<th>dbpedia:Apple_Inc.</th>
<th>dbpedia:Cupertino,_California</th>
<th>dbpedia:Forbes</th>
<th>dbpedia:NeXT</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Mammal</th>
<th>Scientist</th>
<th>Company</th>
<th>Drug</th>
<th>City</th>
<th>Magazine</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dbpedia:Steve_Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
Building the training set for K-Nearest Neighbor algorithm

<table>
<thead>
<tr>
<th>dbpedia:Steve_Jobs</th>
<th>Mammal</th>
<th>Scientist</th>
<th>Company</th>
<th>Drug</th>
<th>City</th>
<th>Magazine</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>dbpo:Person</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Building the training set for K-Nearest Neighbor algorithm

<table>
<thead>
<tr>
<th>dbpedia:Steve_Jobs</th>
<th>Mammal</th>
<th>Scientist</th>
<th>Company</th>
<th>Drug</th>
<th>City</th>
<th>Magazine</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>dbpo:Person</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

stlab.istc.cnr.it
Building the training set for K-Nearest Neighbor algorithm

<table>
<thead>
<tr>
<th>Mammal</th>
<th>Scientist</th>
<th>Company</th>
<th>Drug</th>
<th>City</th>
<th>Magazine</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>dbpo:Person</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Building the training set for K-Nearest Neighbor algorithm

✦ Precision using all DBPO types as features: 31.65%
✦ Precision using the top-level of DBPO as features: 40.27%
Abductive classification with EKPs

**EKPs**

- A EKP of a certain entity type is a small vocabulary that captures the core types used for describing such entity type as it emerges from the Wikipedia crowds.

Visit aemoo.org for an exploratory tool based on EKPs.
How can we infer the type of “Galileo Galilei”?
How can we infer the type of “Galileo Galilei”?

We know its path types

http://www.aemoo.org
We have 231 EKPs

We compare the path types involving “Galileo Galilei” as subject with EKPs in order to identify the most similar, which is the "Scientist" EKP.

http://www.aemoo.org
The inferred type for the resource “Galileo Galilei” is the class “Scientist”
Distinctive weakness of some EKPs

- The distinctive weakness seems due to wide overlaps among some EKPs
- Systematic ambiguity of the 4 largest classes

- Precision and recall on all DBPO types both 44.4%
- Precision and recall on the top-level of DBPO hierarchy: 36.5% and 79.5%
Homotype-based abductive classification

- Homotypes are wikilinks that have the same type on both the subject and the object of the triple

- We have observed how the homotype is usually the most frequent (or in the top 3) wikilink type

- Given an untyped entity, we hypothesize that the most frequent type involved in its ingoing/outgoing wikilinks detects its homotype, hence it indicates its type
Homotype-based abductive classification

Incoming + outgoing wikilinks of dbpedia:Immanuel_Kant

- dbpo:Writer
- dbpo:Book
- dbpo:City
- dbpo:Country
- dbpo:Film
- dbpo:EthnicGroup
- dbpo:ComicsCreator
- dbpo:University
- dbpo:MusicalArtist
- dbpo:Scientist
- dbpo:Philosopher
- dbpo:President

Number of wikilinks
Homotype-based abductive classification

Incoming + outgoing wikilinks of dbpedia:Immanuel_Kant

- number of wikilinks

- dbpo:Writer
- dbpo:Book
- dbpo:City
- dbpo:Country
- dbpo:Film
- dbpo:EthnicGroup
- dbpo:ComicsCreator
- dbpo:University
- dbpo:MusicalArtist
- dbpo:Scientist
- dbpo:Philosopher
- dbpo:President
Results on classifying already typed resources
Results on untyped resources

- Results on a sample of 1,000 untyped resources are much less satisfactory

With EKPs

With Homotypes
Why? [1]

• Typed entities: 2:3 typed wikilinks ratio

• Untyped entities: 1:3 typed wikilinks ratio

• Link structure for untyped entities is not rich enough
Why? [2]

- DBPO does not provide a complete set of classes for correctly typing DBpedia resources

  - dbpedia:List_of_FIFA_World_Cup_finals → Collection
  - dbpedia:Computer_Science → ScientificDiscipline
  - dbpedia:Counterattack → Plan
  - dbpedia:Eros(concept) → Concept
  - dbpedia:Gentlemen’s_agreement → Agreement
Conclusions

- We have investigated different approaches for typing DBpedia resources based on the data set of wikilinks.

- Results are acceptable in the test set, but extensive untypedness in output links, and poor DBPO coverage severely compromise automatic typing for untyped resources.

- We have analyzed possible causes deriving from some bias in DBpedia.
Future work

- **Yago could be helpful but**
  - there is a lack of mapping between YAGO and DBPO
  - it has larger coverage and only an overlap with DBPO
  - the granularity of its categories is finer, and not easily reusable, because the top level is very large
Thank you

Andrea Nuzzolese
- STLab, ISTC-CNR
& Dipartimento di Scienze dell’Informazione
University of Bologna
Italy