On the Robustness of
Standalone Referring Expression Generation
Algorithms Using RDF Data

Pablo Duboue, Martín Domínguez and Paula Estrella
Our Work at a Glance

- The need of robust NLG algorithms.

- 3 algorithms of REG on anachronistic input.
  - On old data, produce a ref. expression, check whether holds.
  - We do not analyze whether the referring expression is good.
  - We have evaluated their goodness in [Pacheco et al., 2012].

- We found poor results with marginal differences among the algorithms.
  - Gardent’s algorithm might be ahead but using closed world assumptions.
  - Nice task and problem, worth extending.
Referring Expression Generation (REG)

• Classic NLG problem
  – Input: set of entities (with a distinguished element), set of triples pertaining to the entities.
  – Output: a Definite Descriptions (DD), set of *positive triples* and a set of *negative triples*.

• Question: how good are referring expressions over erroneous data?
  – Naturally occurring errors using anachronistic data.

• Example: distinguish *Paul McCartney* from *Ringo Starr, John Lennon, George Harrison*
  – NOT associated musical artist: Plastic Ono Band
Three REG Algorithms

- **DR [Dale and Reiter, 1995]**
  - Greedy approach, use a **default ordering**.

- **Gardent [Gardent, 2002]**
  - Constraint satisfaction programming
    - Also needs a default ordering.
  - Generates negations.

- **Graph [Krahmer et al., 2003]**
  - Graph isomorphism
    - Also needs a default ordering.
Experiments With Wikinews-derived REG Tasks

- Wikinews, news articles with *interwiki* links.

  Former [[New Mexico]] {{w|Governor of New Mexico|governor}} {{w|Gary Johnson}} ended his campaign for the {{w|Republican Party (United States)|Republican Party}}

- Focus on people and organizations

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Execution Errors</th>
<th>Dice</th>
<th>Omission Errors</th>
<th>Inclusion Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>People — Entity has “birth date”? ⇒ person (3,051 tasks)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incremental</td>
<td>232 (5%)</td>
<td>0.48</td>
<td>1,406 (50%)</td>
<td>145 (5%)</td>
</tr>
<tr>
<td>Gardent</td>
<td>0 (0%)</td>
<td>0.58</td>
<td>1,089 (36%)</td>
<td>554 (18%)</td>
</tr>
<tr>
<td>Graph</td>
<td>15 (0%)</td>
<td>0.38</td>
<td>1,870 (62%)</td>
<td>20 (0%)</td>
</tr>
<tr>
<td><strong>Organizations — Entity has “creation date”? ⇒ organization (2,370 tasks)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incremental</td>
<td>1,386 (45%)</td>
<td>0.69</td>
<td>305 (31%)</td>
<td>3 (0%)</td>
</tr>
<tr>
<td>Gardent</td>
<td>829 (27%)</td>
<td>0.70</td>
<td>338 (22%)</td>
<td>357 (23%)</td>
</tr>
<tr>
<td>Graph</td>
<td>934 (31%)</td>
<td>0.06</td>
<td>1,347 (94%)</td>
<td>2 (0%)</td>
</tr>
</tbody>
</table>
Implementation Details

- **Alusivo**: an Open Source implementation of REG algorithms
  - [https://github.com/DrDub/Alusivo](https://github.com/DrDub/Alusivo)
  - Java, Maven, RDF-based

- **Interface**
  - public ReferringExpression resolve(URI referent, List<URI> confusors, RepositoryConnection repo)

- **Libraries**
  - Sesame (RDF)
  - ChocoSolver (CSP)
  - jgrapht (Graph algorithms)
DBpedia [Bizer et al., 2009] is an ontology curated from Wikipedia infoboxes

- Infoboxes are the small tables containing structured information at the top of most Wikipedia pages.
- The mappings between the infoboxes labels to the ontology is done in a wiki itself: http://mappings.dbpedia.org/.
- The source code of the scrapping scripts is also available with all its development history.

Not to be confused with a new project targeting to provide structured information to Wikipedia, wikidata.
**Two Versions: Compared**

<table>
<thead>
<tr>
<th></th>
<th>Property 3.6</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type files analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of triples</td>
<td>6,173,940</td>
<td>28,031,852</td>
</tr>
<tr>
<td>Unique subjects (entities)</td>
<td>1,668,503</td>
<td>4,218,628</td>
</tr>
<tr>
<td>Unique objects (types)</td>
<td>250</td>
<td>547</td>
</tr>
<tr>
<td>Max objects per subject</td>
<td>6</td>
<td>16</td>
</tr>
</tbody>
</table>

### Mapping files analysis

<table>
<thead>
<tr>
<th></th>
<th>Property 3.6</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of verbs</td>
<td>1,100</td>
<td>1,370</td>
</tr>
<tr>
<td>Number of triples</td>
<td>13,795,664</td>
<td>33,449,633</td>
</tr>
</tbody>
</table>

- **However, many entities lost their types**
  - From 20,693 Politicians in 3.6, 4,542 are gone (20%-25%).
  - However, the total Politicians in 2014 is 40,343.
Error Prone Task

- **NAACL 2012 bug**
  - DBpedia distributed in two files, we used only one.

- **MICAI 2015 bug**

```java
FileInputStream inNew = new FileInputStream("data/people.tuples.new");
Model mNew = Rio.parse(inNew, "http://localhost/", RDFFormat.NTRIPLES);
Repository repNew = new SailRepository(new MemoryStore());
ValueFactory fNew = repNew.getValueFactory();
RepositoryConnection connNew = repNew.getConnection();
connNew.add(mNew);

FileInputStream inOld = new FileInputStream("data/people.tuples.old");
Model mOld = Rio.parse(inOld, "http://localhost/", RDFFormat.NTRIPLES);
Repository repOld = new SailRepository(new MemoryStore());
ValueFactory fOld = repOld.getValueFactory();
RepositoryConnection connOld = repNew.getConnection();
connOld.add(mOld);
```
Learning Orderings

- Iberamia, to appear:
  - *Using Robustness to Learn to Order Semantic Properties in Referring Expression Generation*, Duboue and Domínguez.

- Intuitions
  - A good ref. expression should refer to stable properties.

- Results
  - Robustness helps to learning orderings.
  - But popularity on a folksonomy is a stronger signal.
Conclusions

- DBpedia/Wikinews is a suitable source for doing research on robust REG algorithms.

Where to go from Here:

- Better understanding of the parameterization of the algorithms.
- Distinguish data changes from missing data.
- More REG algorithms.
- Better testing of our current implementations.
Backup Slides

References


