From unstructured to linked data: entity extraction and disambiguation by collective similarity maximization
Outline

1. Introduction and motivation
   - Service-oriented knowledge extraction
   - Semantic entity disambiguation
2. Proposed method
3. Current development
4. Conclusion
5. Demo
The big picture:

- A lot of available knowledge is available as unstructured data, especially in text form
- We can do information extraction of:
  - topic, sentiment, named entities, semantic entities, relationships
- Enrycher: a web service for providing this additional knowledge
Problem domain:
- Identifying semantic entities in text
- Automatically describe text with an ontology to enable semantic integration
Main challenge

- Correctly disambiguating entities in text
  - Using different sources of information to improve disambiguation quality
  - Results are probabilistic
Related work

- Disambiguation of in-text expressions (NLP)
  - Machine learning vs. pattern matching and heuristics
- Ontology alignment (Semantic Web)
- Entity resolution (databases)
Semantic entity disambiguation

- Given a text document:
  - Extract named entity mentions
  - Consolidate entity mentions into in-text entities
  - Match in-text entities with entities from the ontology
As an entity resolution problem:
Our general approach

- An ontology-based approach:
  - Ranks candidate entities based on content similarity
  - Does not require learning

- The ontology should specify:
  - Aliases for entities \((\text{rdfs:label})\)
  - Descriptions of entities \((\text{rdfs:comment})\)
  - Types of entities \((\text{rdf:type})\)

- *Our example: Dbpedia+YAGO*
Architecture

News corpus → Named entity extraction → Local entity consolidation → Entity resolution → Enriched news corpus

Background knowledge (DBpedia)
Named entity extraction

- Identifies words or phrases that may represent a concept
- Identifies the type of named entity
- In-text mentions are ambiguous!
Named entity consolidation

- Data cleaning phase
  - Canonicalization
  - Resolve partial name and acronym matches
  - Simple attribute extraction (gender, title)
Entity resolution approaches

- **Pair-wise (baseline):**
  - For each in-text entity, choose the candidate entity which is the most similar

- Is each disambiguation choice independent?
  - Pair-wise vs. collective disambiguation
Our approach:

For each in-text entity, choose the candidate entity which is most similar to the in-text entity and all other candidate entities that are selected.
Similarity (relevance) criteria

- Similarity between candidate entity’s description \((rdfs:comment)\) and article text
- Similarity between attributes of in-text entity and candidate entity \((i.e. rdf:type, foaf:gender)\)
The algorithm

- Use greedy entity resolution for iteratively selecting entities
  1. Prior pair-wise evaluation of candidate entities;
  2. While top available candidate is good enough:
     1. Select top candidate;
     2. Update evaluations of available candidates;
- We evaluate candidates by:
  - similarity to local entity
  - similarity to other selected candidates
Intuition: entities that co-occur tend to be more similar

Select a subset of entities which are most related to each other

Formulated as maximizing graph density:

\[ \text{sim}_\text{collective}(C) = \frac{\sum_{e_i \in C} \sum_{e_j \in C, e_i \neq e_j} \text{sim}_\text{entity}(e_i, e_j)}{|C|^2} \]
Experiment

- Text corpus:
  - New York Times Annotated Corpus
  - Manually evaluated 693 entity resolution decisions in 50 articles
Results

![Graph showing Precision vs. Recall for pair-wise and collective with content models]
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>F₀.₂</th>
<th>F₁.₀</th>
<th>Recall at 80% prec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (pairwise)</td>
<td>0.772</td>
<td>0.749</td>
<td>0.51</td>
</tr>
<tr>
<td>Collective similarity maximization</td>
<td>0.789</td>
<td>0.750</td>
<td>0.66</td>
</tr>
</tbody>
</table>
Results (II.)

- Method shows improvement on high-precision operation
- Challenges: multi-theme documents
  - Maximizing collective similarity does not necessarily help
  - Solvable by segmentation to smaller single-theme sections
Results (III.)

- Challenge:
  - Errors, made on early decisions propagate throughout the document

- Performance
  - We avoid exhaustive search with collective entity resolution, having manageable polynomial computational complexity
Current development

- Content and attribute similarity is just one possible way of expressing relatedness between entities
  - Co-occurrences – a statistical learning approach
  - Explicit semantic relations as a relatedness measure
Future work

- Method improvement
  - Exploit different sources of relatedness for relational disambiguation – not only content similarity

- Application
  - Combine with triplet extractor: extract consistent knowledge from text in the form of assertions
Methods, used for entity resolution (ontology matching) are applicable to disambiguation

Manageable computational complexity

A collective approach leverages relatedness information
Semantic entity extraction is an important information extraction task.

Extracting information by hand is often not feasible.

Next step: extract not only entities, but also relations between entities.
- Visualize the document as a semantic graph.
- Demo.
Demo!

- **Enrycher**
  - Joint work with Lorand Dali, Delia Rusu, Blaž Fortuna, Marko Grobelnik and Dunja Mladenić
  - http://enrycher.ijs.si
Open Questions

- Are <100% precise annotations acceptable?
- User involvement?
- Your questions?