LODifier: Generating Linked Data from Unstructured Text

Isabelle Augenstein[1] Sebastian Padó[1] Sebastian Rudolph[2]

[1]Department of Computational Linguistics, Universität Heidelberg, Germany
[2]Institute AIFB, Karlsruhe Institute of Technology, Germany

{augenste,pado}@cl.uni-heidelberg.de, rudolph@kit.edu

Extended Semantic Web Conference, 2012

Outline

Introduction

What?

- Represent information **from NL text** as Linked Data
- Creation of graph-based representation for **unstructured** plain text

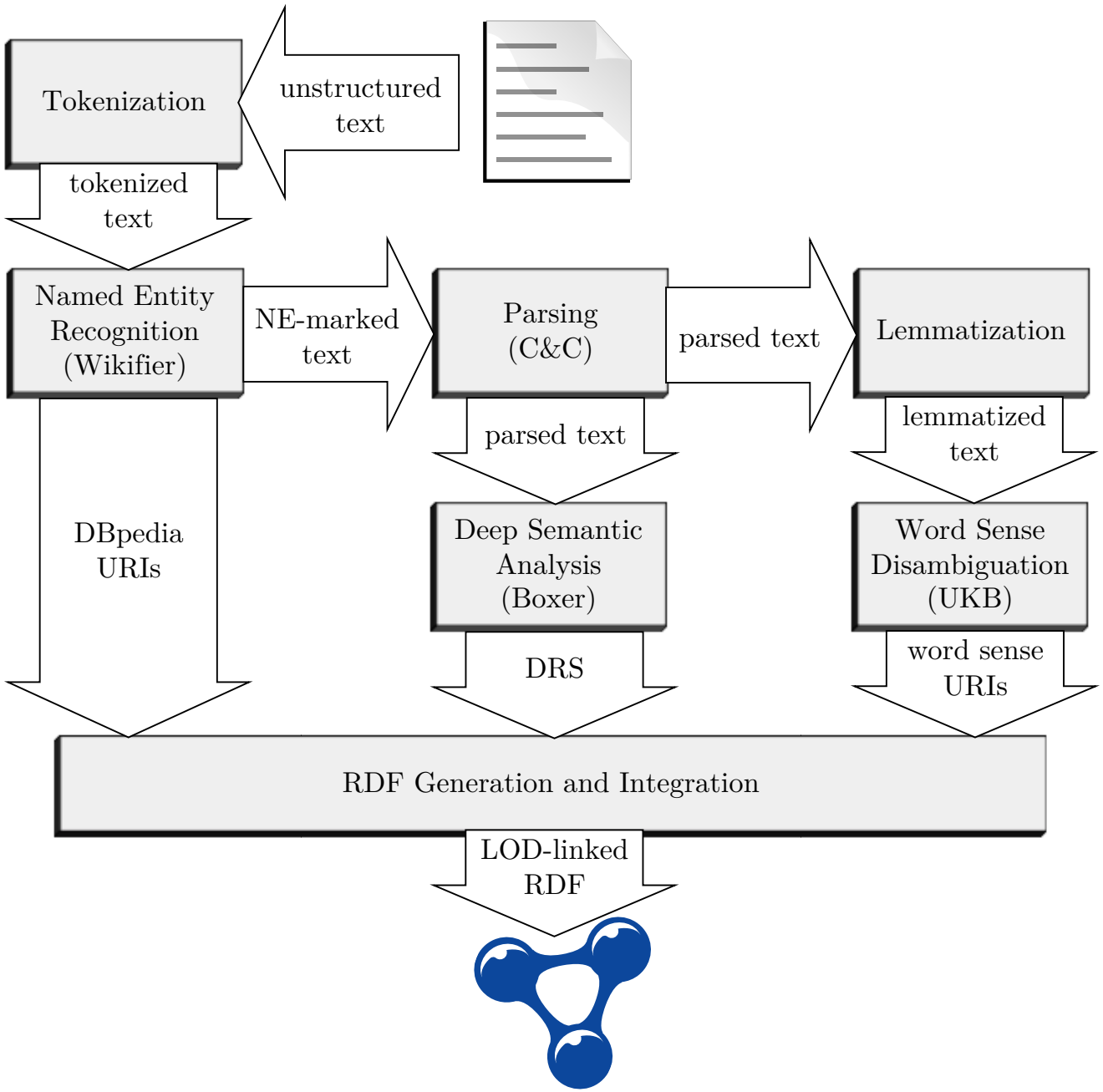
Why?

- Task is easy for structured text, but quite challenging for unstructured text
- Most other approaches are very selective, e.g., extract only pre-defined relations
- Our approach translates the text **in its entirety**
- Possible use cases: Domain-independent scenarios in which no pre-defined schema is available

Introduction

How?

- Use noise-tolerant NLP pipeline to analyze text
- Translate result into RDF representation
- Embed output in the LOD cloud by using vocabulary from DBPedia and Wordnet


Tokenization
unstructured text
tokenized text
Named Entity Recognition (Wikifier)
NE-marked text
Parsing (C&C)
parsed text
Lemmatization
parsed text
lemmatized text
DBpedia URIs
Deep Semantic Analysis (Boxer)
Word Sense Disambiguation (UKB)
DRS
word sense URIs
RDF Generation and Integration
LOD-linked RDF

NLP analysis steps (NER)

Named Entity Recognition (Wikifier [1]):

- Named entities are recognized using Wikifier
- Wikifier finds Wikipedia links for named entities
- Wikipedia URLs are converted to DBPedia URIs

> *The New York Times reported that John McCarthy died.*
> *He invented the programming language LISP.*

Figure: Test sentences

```
[[The New York Times]] reported that [[John McCarthy (computer
scientist)|John McCarthy]] died. He invented the [[Programming
language|programming language]] [[Lisp (programming language)|
Lisp]].
```

Figure: Wikifier output for the test sentences.

NLP analysis steps (WSD)

Word Sense Disambiguation (UKB [2])):

- Words are disambiguated using UKB, an unsupervised graph-based WSD tool
- UKB finds WordNet links for word senses
- UKB output is converted to RDF WordNet URIs

```
ctx_0 w1   00965687-v !! report
ctx_0 w4   00358431-v !! die
ctx_1 w7   01634424-v !! invent
ctx_1 w9   06898352-n !! programming_language
ctx_1 w10  06901936-n !! lisp
```

Figure: UKB output for the test sentences.

[2] http://ixa2.si.ehu.es/ukb/

NLP analysis steps (Parsing and Semantic Analysis)

Parsing (C&C [3]):

- Sentences are parsed with the C&C parser
- The C&C parser is a statistical parser that uses the combined categorial grammar (CCG)
- CCG is a semantically motivated grammar, making it suitable for further semantic analysis

Deep Semantic Analysis (Boxer [4]):

- Boxer builds on the output of C&C and produces discourse representation structures (DRSs)
- DRSs model the meaning of texts in terms of the relevant entities (discourse referents) and relations between them (conditions)

[3] `http://svn.ask.it.usyd.edu.au/trac/candc/`
[4] `http://svn.ask.it.usyd.edu.au/trac/candc/wiki/boxer`

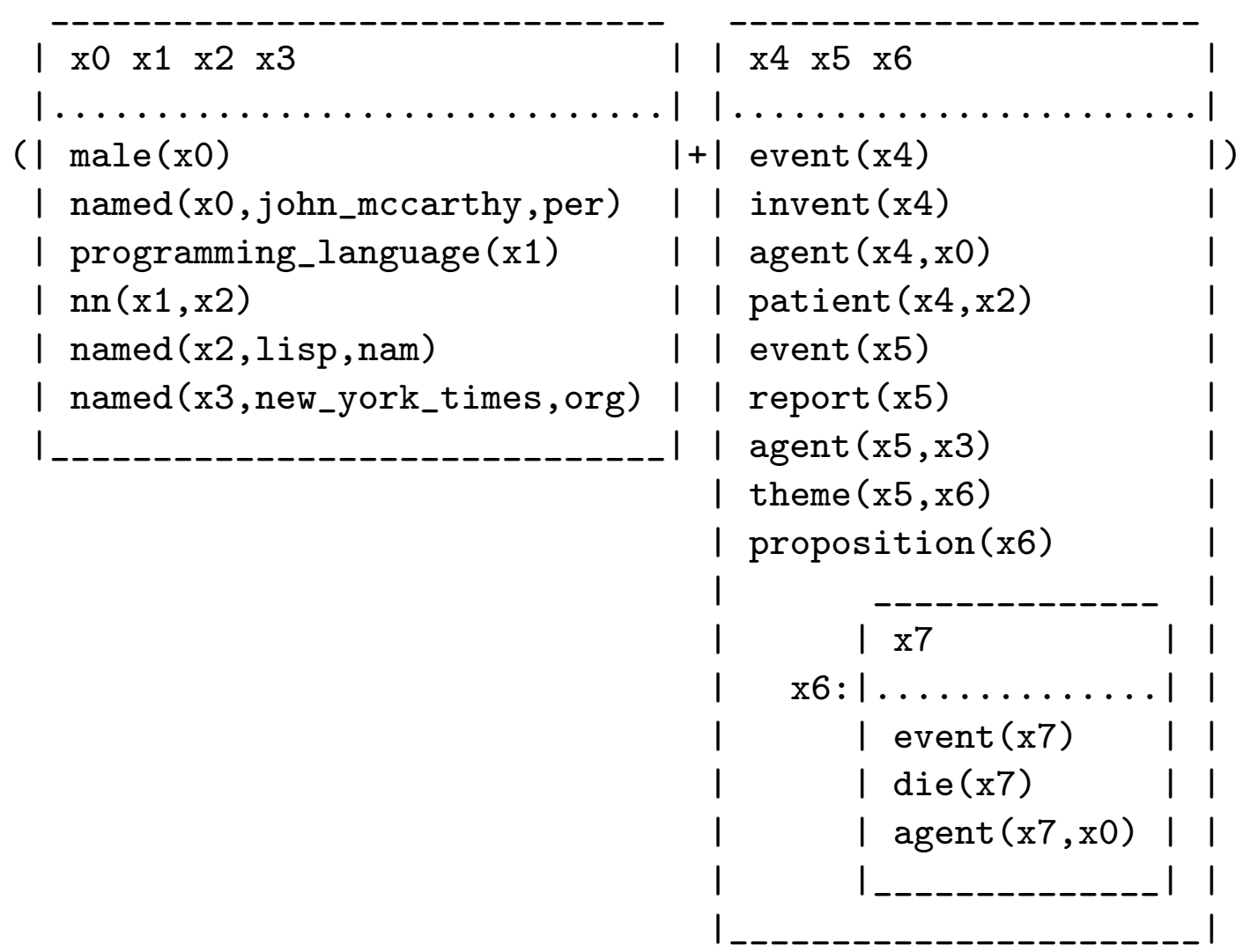


Figure: Boxer output for the example sentences

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

```
_:var0x0 drsclass:named ne:john_mccarthy ;
         rdf:type drsclass:male , foaf:Person ;
         owl:sameAs dbpedia:John_McCarthy_(computer_scientist) .
_:var0x1 rdf:type class:programming_language ;
         owl:sameAs dbpedia:Programming_language .
_:var0x2 drsrel:nn _:var0x1 .
_:var0x2 drsclass:named ne:lisp ;
         owl:sameAs     dbpedia:Lisp_(programming_language) .
_:var0x3 drsclass:named ne:the_new_york_times ;
         owl:sameAs     dbpedia:The_New_York_Times .
_:var0x4 rdf:type drsclass:event ,  wn30:wordsense-invent-verb-2 .
         drsrel:agent _:var0x0 ;  drsrel:patient _:var0x2 .
_:var0x5 rdf:type drsclass:event ,  wn30:wordsense-report-verb-3 ;
         drsrel:agent _:var0x3 ;  drsrel:theme _:var0x6 .
_:var0x6 rdf:type drsclass:proposition ,  reify:proposition ,  reify:conjunction ;
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    drsclass:event .  ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate rdf:type ;
                           rdf:object    wn30:wordsense-die-verb-1 . ]
         reify:conjunct [  rdf:subject   _:var0x7 ;
                           rdf:predicate drsrel:agent ;
                           rdf:object    _:var0x0 .  ]
```

Figure: LODifier output for the test sentences.

Evaluation overview

Task:

- Assess document similarity
- *Story link detection task*, part of topic detection and tracking (TDT) family of tasks

Data:

- TDT-2 benchmark dataset [5]
- Consists of newspaper, TV and radio news in English, Arabic and Mandarin
- Subset: English language, only newspaper articles
- 183 positive and negative document pairs

[5] http://projects.ldc.upenn.edu/TDT2/

Evaluation overview

Method:

- Define various document similarity measures
  - Measures without structural information
  - Measures with structural information
- Documents are predicted to have the same topic if they have a similarity of θ or more
- θ is determined with supervised learning

Evaluation method

Similarity measures without structure:

- **Random baseline**
- **Bag-of-words baseline**: Word overlap between the two documents in a document pair
- **Bag-of-URI baseline**: URI overlap between two RDF documents

Evaluation method

URI class variants:

- Three variants for URI baseline: Take the relative importance of various URI classes into account
  - **Variant 1**: All NEs identified by Wikifier, all words successfully disambiguated by UKB (namespaces `dbpedia:`, `wn30:`)
  - **Variant 2**: Adds all NEs recognized by Boxer (namespace `ne:`)
  - **Variant 3**: Further adds the URIs for all words that were not recognized by Wikifier, UKB or Boxer (namespace `class:`)

Evaluation method

Extended setting:

- Aims at drawing more information from LOD cloud into the similarity computation
- The generated RDF graph is enriched by:
  - DBpedia categories (namespace `dbpedia:`)
  - WordNet synsets and WordNet senses related to the URIs in the generated graph (namespace `wn30:`)

Evaluation method

Similarity measures with structure:

- Full-fledged graph similarity measures, e.g. based on isomorphic subgraphs, are infeasible due to the size of the RDF graphs
- Our similarity measures are based on the shortest paths between relevant nodes
- Our intuition is that short paths between URI pairs denote relevant semantic information and we expect they are related by a short path in other documents as well

Evaluation method

Similarity measures with structure:

$$\text{proSim}_{k,Rel,f}(G_1, G_2) = \frac{\sum_{\substack{a,b \in Rel(G_1) \\ \langle a,b \rangle \in C_k(G_1) \cap C_k(G_2)}} f(\ell(a,b))}{\sum_{\substack{a,b \in Rel(G_1) \\ \langle a,b \rangle \in C_k(G_1)}} f(\ell(a,b))}$$

k: path length, Rel: semantic relations, f: similarity function

- $\text{proSim}_{\text{cnt}}$: Uses $f(\ell) = 1$, i.e., just counts the number of paths irrespective of their length
- $\text{proSim}_{\text{len}}$: Uses $f(\ell) = 1/\ell$, giving less weight to longer paths
- $\text{proSim}_{\text{sqlen}}$: Uses $f(\ell) = 1/\sqrt{\ell}$, discounting long paths less aggressively than $\text{proSim}_{\text{len}}$

Results

Model	normal	extended
Similarity measures without structure		
Random Baseline	50.0	–
Bag of Words	63.0	–
Bag of URIs (Variant 3)	**73.4**	**76.4**
Similarity measures with structure		
$\text{proSim}_{\text{cnt}}$ (k=6, Variant 1)	77.7	77.6
$\text{proSim}_{\text{cnt}}$ (k=6, Variant 2)	79.2	79.0
$\text{proSim}_{\text{cnt}}$ (k=8, Variant 3)	**82.1**	**81.9**
$\text{proSim}_{\text{len}}$ (k=6, Variant 3)	81.5	81.4
$\text{proSim}_{\text{sqlen}}$ (k=8, Variant 3)	81.1	80.9

Summary

Main contributions:

- Service for the LOD community that can be used in different scenarios that can profit from structural representation of text
- Combination of well-established concepts and systems in a deep semantic pipeline
- Linking existing NLP technologies to the LOD cloud
- Evaluation of LODifier in a topic link detection task gives clear evidence of its potential

Possible future work directions:

- Instead of a pipeline approach, simultaneously consider information from the NLP components to allow interaction between them
- Test LODifier in different scenarios

Thank you!