Inflection-Tolerant Ontology-Based Named Entity Recognition for Real-Time Applications

Christian Jilek  Markus Schröder  Rudolf Novik
Sven Schwarz  Heiko Maus  Andreas Dengel

Parts of this work were funded by the DFG in the SPP on Intentional Forgetting in Organizations.
Motivation

Forgetful & Self-Organizing Information Systems
(to support information management & knowledge work)

↓

continuous information value assessment

↓

continuous user activity tracking and evidence processing

↓

information extraction in (near) real-time
Problem of Inflections

DBpedia Spotlight:

### Related Work

<table>
<thead>
<tr>
<th>Inflection-Tolerant NER</th>
<th>Real-Time Capable NER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Savary &amp; Piskorski (2010) → IE platform SProUT, Polish, explicitly listing all inflected forms</td>
<td>Dlugolinsky, Nguyen et al. (2013/2014) → several gazetteer-based approaches</td>
</tr>
<tr>
<td></td>
<td>Al-Jumaily et al. (2013) → NER for Arabic text mining, no details on performance given</td>
</tr>
</tbody>
</table>
Approach

The approach described involves an NE recognizer, which acts as a combination of several multi-layer finite state transducers having different tolerance levels. This recognizer connects to knowledge graph(s) [instance labels, types, …] and allows access to language information [word types, flections, …].

Christian Jilek – 13th DBpedia Community Meeting 2019, Leipzig
Multi-Layer FST with High Tolerance

Layer 1: Character Layer

```
D→E→U→T→S→C→H→E
S→R→N
M→N
O→R→S→C→H→U→N→G→S→Z→E→N→T→R
U→M→S
E→N
```

Input:

Deutsches Forschungszentrum für Künstliche Intelligenz

Layer 2: Word Layer

```
Ø→w1→w2→w3→w4→w5
```

“Deutsches Forschungszentrum für Künstliche Intelligenz“ (German Research Center for Artificial Intelligence)
Multi-Layer FST with Low Tolerance

Input: Christian_Jilek

Layer 1: Character Layer

Layer 2: Word Layer

„Christian“ [w1]

„Jilek“ [w2]
Evaluation Setting

• idea:
  • use the German Wikipedia as a large set of texts written by different people
  • use DBpedia types to decide whether to apply low or high inflection tolerance
  • use Wikipedia annotations as a „silver standard“
    • term used (often inflected form) manually annotated with its article name (often basic form)
      
      |
      | [[ Haus | Häuser ]]
      | [[ Junktor | Junktoren ]]

• problems:
  • independent term-links-combinations
  • adjective-noun-combinations

  → use Levenshtein distance (LD) to identify samples (typically LD<=4)

  • ambiguities (e.g. >1000 instances of „Jewish Cemetery“)
  • terms not annotated in „their own“ article (e.g. „Berlin“ in article about „Berlin“)

• benefit: 3.9M articles having 50.4M annotations
Results: Recall

![Graph showing recall results for different LD values with four methods: amount (%), HMT/CST, MLFST, StemFST.](image-url)
Results: Measuring Precision

„A commercial personal information management tool is used in the project.“

- $P_O^*$: only overlapping terms as false positives, ambiguities disregarded
- $P_O$: only overlapping terms including ambiguities as false positives
- $P_A^*$: all other terms as false positives, ambiguities disregarded
- $P_A$: all other terms including ambiguities as false positives
Results: Precision

<table>
<thead>
<tr>
<th></th>
<th>HMT/CST</th>
<th>MLFST</th>
<th>StemFST</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO*</td>
<td>80%</td>
<td>40%</td>
<td>20%</td>
</tr>
<tr>
<td>PO</td>
<td>60%</td>
<td>30%</td>
<td>10%</td>
</tr>
<tr>
<td>PA*</td>
<td>40%</td>
<td>20%</td>
<td>10%</td>
</tr>
<tr>
<td>PA</td>
<td>20%</td>
<td>10%</td>
<td>5%</td>
</tr>
</tbody>
</table>
Results: Processing Speed & Memory Consumption

- **HMT**: 15084 char/ms
- **CST**: 4075 char/ms
- **MLFST**: 3281 char/ms
- **StemFST**: 5048 char/ms
- **OpenNLP (basic PL)**: 350 char/ms
- **CoreNLP (basic PL)**: 6 char/ms

**Memory Consumption**

- **HMT**: 4 GiB
- **CST**: 3 GiB
- **MLFST**: 1 GiB
- **StemFST**: 1 GiB

**Total runtime for performing NER on German Wikipedia**

- **HMT**: 60 min
- **CST**: 45 min
- **MLFST**: 30 min
- **StemFST**: 15 min

**basic PL:**
- tokenizer
- sentence splitter
- POS tagger
Conclusion

• presented inflection-tolerant and real-time capable OB NER approach based on
  • Trie-based string matching
  • finite state cascades
  • exhaustive inflection listing
  • exploiting ontological background information

• comparably fast as available high speed methods

• outperforming them in recognizing terms that lexically vary slightly (e.g. inflection)

• narrowing the gap to more sophisticated but slower NLP pipelines without losing too much runtime performance
Outlook

• incorporate disambiguation mechanisms (exploiting user context)

• add more layers to scan for patterns (ToDos, appointments, Hearst patterns, …)

• improve language capabilities (rules, heuristics, multi-language support, …)

• incorporate StemFST into MLFST for multi-word terms (slightly better precision)
Selected References


Thanks for your attention! 😊

Parts of this work were funded by the DFG in the SPP on Intentional Forgetting in Organizations.