ASP-based Inductive Logic Programming applied to Phrase Chunking: Challenges and Improvements

Peter Schüller

Knowledge Based Systems Group
TU Wien, Wien, Austria

Computer Engineering Department
Marmara University, Istanbul, Turkey
Motivation

- Participation in SemEval 2016 iSTS (Interpretable Semantic Similarity) (Agirre et al., 2016)
- Rule-based (ASP) system for aligning phrases of 2 sentences
- ASP-based ad hoc encoding for Phrase Chunking
  - better than baseline
  - much worse than CRF-based systems
  - 3 very different datasets with 756/750/330 sentences
Motivation

- Participation in SemEval 2016 iSTS (Interpretable Semantic Similarity) (Agirre et al., 2016)
- Rule-based (ASP) system for aligning phrases of 2 sentences
- ASP-based ad hoc encoding for Phrase Chunking
  - better than baseline
  - much worse than CRF-based systems
  - 3 very different datasets with 756/750/330 sentences

⇒ Let’s learn rules for chunking from datasets!
ILP Formulation of Chunking

- Instance:
  \[\text{pos}(\text{Pos}, \text{Token})\]

- Background Knowledge:
  \[\text{postype}(\text{Pos}) \leftarrow \text{pos}(\text{Pos}, \text{Token}).\]
  \[\text{token}(\text{Token}) \leftarrow \text{pos}(\text{Pos}, \text{Token}).\]
  \[\text{nextpos}(\text{Pos}, \text{Token} - 1) \leftarrow \text{pos}(\text{Pos}, \text{Token}), \text{token}(\text{Token} - 1).\]

- Potential Head Atoms in Learned Program (#modeh):
  \[\text{split}(+\text{token})\]

- Potential Body Atoms in Learned Program (#modeb):
  \[\text{pos}($\text{postype}$, +\text{token})\]
  \[\text{nextpos}($\text{postype}$, +\text{token})\]
ILP Examples for Chunking

▶ Example Sentence

[Former₁ Nazi₂ death₃ camp₄ guard₅ Demjanjuk₆] [dead₇] [at₈ 91₉]

▶ Background Knowledge:

\[ \text{pos}(c_{NNP}, 1), \text{pos}(c_{NNP}, 2), \text{pos}(c_{NN}, 3) \]
\[ \text{pos}(c_{NN}, 4), \text{pos}(c_{NN}, 5), \text{pos}(c_{NNP}, 6) \]
\[ \text{pos}(c_{VBD}, 7), \text{pos}(c_{IN}, 8), \text{pos}(c_{CD}, 9) \]
\[ \text{goodchunk}(1) \leftarrow \text{not split}(1), \text{not split}(2), \text{not split}(3), \]
\[ \text{not split}(4), \text{not split}(5), \text{split}(6) \]
\[ \text{goodchunk}(7) \leftarrow \text{split}(6), \text{split}(7) \]
\[ \text{goodchunk}(8) \leftarrow \text{split}(7), \text{notsplit}(8) \]

▶ Examples:

#example goodchunk(1).
#example goodchunk(7).
#example goodchunk(8).
Attempts with Existing Solvers

- **XHAIL (Ray, 2009):**
  open source, old ASP solver technology
  feasible with 50 examples (<1 day for finding hypothesis)

- **ILED (Katzouris et al., 2015):**
  open source, incremental solving
  fails for noisy input, several problematic algorithm corner cases

- **ILASP2 (Law et al., 2015):**
  closed source, most advanced capabilities (learning guesses)
  feasible with 5 examples (<1 day for finding hypothesis)
Attempts with Existing Solvers

- XHAIL (Ray, 2009):
  open source, old ASP solver technology
  feasible with 50 examples (<1 day for finding hypothesis)

- ILED (Katzouris et al., 2015):
  open source, incremental solving
  fails for noisy input, several problematic algorithm corner cases

- ILASP2 (Law et al., 2015):
  closed source, most advanced capabilities (learning guesses)
  feasible with 5 examples (<1 day for finding hypothesis)

- Infeasible with ASP-based ILP technology
XHAIL Principle (Ray, 2009)

Examples
Background Knowledge
Head Mode Bias

Abduction

Kernel Set

→ which potential heads $\Delta$ make examples true?

Body Mode Bias

Deduction

ground Kernel Program

→ which potential rules $K$ make $\Delta$ true?

Generalisation

replace constants with variables

non-ground Kernel Program

Induction

Hypothesis

→ smallest subset of $K'$ that satisfies $E$

ASP-based Inductive Logic Programming for NLP 6
XHAIL Principle (Ray, 2009)

Examples
Background Knowledge
Head Mode Bias

Abduction

Kernel Set

Deduction

Body Mode Bias

Generalisation

ground Kernel Program

which potential rules $K$ make $\Delta$ true?

replace constants with variables

non-ground Kernel Program

Induction

Hypothesis

Cheap ASP Optimization

$\rightarrow$ which potential heads $\Delta$ make examples true?

Expensive ASP Optimization

$\rightarrow$ smallest subset of $K'$ that satisfies $E$
XHAIL Pruning (Kazmi et al., 2017)

Examples
Background Knowledge
Head Mode Bias

Abduction
Kernel Set

Deduction

Body Mode Bias

Generalisation

Ground Kernel Program \( K \)

Non-ground Kernel Program \( K' \)

Induction

Hypothesis

Generalisation (counting)

\( K' + \) support counter for \( K \rightarrow K' \)

Pruning

coarse-grained \( K'' \subseteq K' \)
XHAIL Pruning (Kazmi et al., 2017)

- Examples
- Background Knowledge
- Head Mode Bias

Abduction

Kernel Set

Deduction

Body Mode Bias

Generalisation

ground Kernel Program $K$

Generalisation (counting)

$K' +$ support counter for $K \rightarrow K'$

non-ground Kernel Program $K'$

Replaced

Added

Induction

Hypothesis

- $K'' = \text{nonground rules that contribute to } \geq P \text{ examples}$

ASP-based Inductive Logic Programming for NLP
<table>
<thead>
<tr>
<th>Examples</th>
<th>Pruning</th>
<th>Suboptimality</th>
<th>Train F1</th>
<th>Test F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>172.8</td>
<td>70.7</td>
<td>60.5</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>10.9</td>
<td>69.3</td>
<td>62.4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.3</td>
<td>66.5</td>
<td>62.7</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.0</td>
<td>65.1</td>
<td>61.6</td>
</tr>
<tr>
<td>500</td>
<td>0</td>
<td>31954.4</td>
<td>39.2</td>
<td>38.9</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>17855.1</td>
<td>40.7</td>
<td>39.7</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6238.5</td>
<td>53.0</td>
<td>49.4</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4260.9</td>
<td>59.2</td>
<td>54.7</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1598.6</td>
<td>67.1</td>
<td>59.7</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1117.2</td>
<td>73.3</td>
<td>62.1</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>732.6</td>
<td>71.7</td>
<td>62.7</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>561.4</td>
<td>71.2</td>
<td>63.6</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>475.0</td>
<td>71.8</td>
<td>64.0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>312.7</td>
<td>71.2</td>
<td>63.3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>220.3</td>
<td>73.4</td>
<td>63.7</td>
</tr>
</tbody>
</table>

Resource Limits: 30 min, 5 GB

Pruning = adjustment

LoD vs. Optimality
Wrapup

- Pruning and modern ASP algorithms make learning feasible.
- Hypotheses have provided insights about dataset issues.
- F1 score is significantly better than with manual rules.
Pruning and modern ASP algorithms make learning feasible.
Hypotheses have provided insights about dataset issues.
F1 score is significantly better than with manual rules.
Contribution: parameter for level of detail vs. optimality.
Limitation: mode bias very simple.
Possible Future Directions

- Incremental learning with ASP + noise.
- Pruning for multi-model hypothesis ILP algorithms.
- Combining Prolog-based and ASP-based ILP techniques.
- Find logic programming fragments that permit scalability and provide (limited) nonmonotonicity.
Possible Future Directions

- Incremental learning with ASP + noise.
- Pruning for multi-model hypothesis ILP algorithms.
- Combining Prolog-based and ASP-based ILP techniques.
- Find logic programming fragments that permit scalability and provide (limited) nonmonotonicity.

Thank you for your attention!

contact Peter Schüller
schueller.p@gmail.com
Knowledge Based Systems Group
Faculty of Informatics
TU Wien, Vienna, Austria

