An Imitation Learning Approach to Unsupervised Parsing

Bowen Li,\textsuperscript{e} Lili Mou,\textsuperscript{a} Frank Keller\textsuperscript{e}
Why Unsupervised Parsing?

Engineering motivation:

- ~6,000 languages in the world
- Treebanks for ~70 languages (many of them small)
- Syntactic annotation
  - slow and costly
  - relying on expert linguists

We need a way of inducing syntactic knowledge

- Based on simple, crowd-sourcable sentence annotation
- E.g., natural language inference, sentiment
Why Unsupervised Parsing?

Cognitive motivation: how children learn languages?

▶ 18 months: start with two word utterances
▶ By 5 years: generate complex syntax (Brown’s stages):
  ▶ relative clauses, infinitival, gerunds, wh-phrases, passives
▶ No explicit supervision is provided (children don’t see syntax trees)
▶ But they receive indirect feedback: is an utterance understood or not?

To model this, we need a way of inducing syntactic knowledge based on simple semantic labels at the sentence level
Unsupervised Parsing

**Goal:** learn linguistically meaningful syntax (tree structures) without treebank supervision

**Approach:**
- Get training signal from a secondary task:
  - Language modeling
  - Semantically oriented tasks (e.g., natural language inference, sentiment)
- Try to induce meaningful “latent” tree structures
Hard Discrete Parsers

Examples:
- RL-SPINN [Yogatama et al., 2017], Soft-Gating [Maillard et al., 2017],
  Gumbel-Tree-LSTM [Choi et al., 2018]

Advantages:
- Models have grounded parsing actions

Disadvantages: Not differentiable
- Reinforcement learning $\implies$ doubly stochastic gradient descent, poor local optima, low self-agreement
- Dynamic Programming marginalization $\implies$ high time complexity
Soft Continuous Parsers

Very recent work:
- Parsing-reading-predict network [PRPN, Shen et al., 2018]
- Ordered Neurons [ON-LSTM, Shen et al., 2019]

Advantages:
- Relaxing discrete parsing by continuous notions (e.g., structured attention) $\implies$ easy to train by differentiation

Disadvantages:
- Inducing syntax from continuous relaxation is not learnable
- Parsing operations are stipulated externally by heuristics
Combine both Worlds by Imitation Learning

- Is it possible to combine both approaches?
Combine both Worlds by Imitation Learning

- Is it possible to combine both approaches?
- Yes! We can use imitation learning!
Combine both Worlds by Imitation Learning

- Is it possible to combine both approaches?
- Yes! We can use imitation learning!
- Coupling soft continuous parser and hard discrete parser at the intermediate output level (parse tree)
Combine both Worlds by Imitation Learning

$J_{\text{task1}}$

Pretraining

Soft Parser

Induce

Imperfect parsing label

$J_{\text{parse}}$

$J_{\text{task2}}$

Supervised Learning

Policy refinement

Hard Parser

8 / 25
Combine both Worlds by Imitation Learning

- $J_{task1}$
- Pretraining
- Soft Parser
- Induce

- $J_{parse}$
- Imperfect parsing label

- $J_{task2}$
- Supervised Learning
- Policy refinement
- Hard Parser
Combine both Worlds by Imitation Learning

$J_{task1}$

Pretraining

Soft Parser

Induce

Imperfect parsing label

$J_{parse}$

$J_{task2}$

Supervised Learning

Policy refinement

Hard Parser
PRPN as the Soft Parser

Parsing-reading-predict network (PRPN; [Shen et al. 2018])

LSTM with structured attention for LM

Syntactic distance

Induce

Height

[Shen et al., 2018]
PRPN as the Soft Parser

**Syntactic distance**

```
/  /  /  /  /  \\
\ w_1 w_2 w_3 w_4 w_5 /  \\
```

Syntactic Distance

2 3 1 2

**Tree structure**

```
/  /  /  /  /  \  \  \\
\ w_1 w_2 w_3 w_4 w_5 /  /  \  \\
```

Syntactic Distance

2 3 1 2

vanilla induction
PRPN as the Soft Parser

Syntactic distance

Tree structure

vanilla induction
PRPN as the Soft Parser

Syntactic distance

Tree structure

heuristic-based Induction

Right-branching bias

Syntactic Distance

2 3 1 2

2 3 1 2

Syntactic Distance

w_1 w_2 w_3 w_4 w_5

w_1 w_2 w_3 w_4 w_5
PRPN as the Soft Parser

Syntactic distance

Tree structure

heuristic-based Induction

Right-branching bias
Gumbel-Tree-LSTM as the Hard Parser

Tree-LSTM for sentence classification

Learning tree structures by Straight-Through Gumbel Softmax

Probability 0.2 0.1 0.7

Gumbel
Our Approach
Our Approach

Two-stage training:

- Stage 1: step-by-step supervised learning
Our Approach

Two-stage training:

- Stage 1: step-by-step supervised learning
- Stage 2: policy refinement on NLI task
## Experimental Results: Parsing Results on All-NLI

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean $F$</th>
<th>Self-agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left-Branching</td>
<td>18.9</td>
<td>-</td>
</tr>
<tr>
<td>Right-Branching</td>
<td>18.5</td>
<td>-</td>
</tr>
<tr>
<td>Balanced-Tree</td>
<td>22.0</td>
<td>-</td>
</tr>
<tr>
<td>Gumbel-Tree-LSTM</td>
<td>21.9</td>
<td>56.8</td>
</tr>
<tr>
<td>PRPN</td>
<td>51.6</td>
<td>65.0</td>
</tr>
</tbody>
</table>

More settings and analysis in our paper.
### Experimental Results: Parsing Results on All-NLI

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean $F$</th>
<th>Self-agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left-Branching</td>
<td>18.9</td>
<td>-</td>
</tr>
<tr>
<td>Right-Branching</td>
<td>18.5</td>
<td>-</td>
</tr>
<tr>
<td>Balanced-Tree</td>
<td>22.0</td>
<td>-</td>
</tr>
<tr>
<td>Gumbel-Tree-LSTM</td>
<td>21.9</td>
<td>56.8</td>
</tr>
<tr>
<td>PRPN</td>
<td>51.6</td>
<td>65.0</td>
</tr>
<tr>
<td>Imitation (first stage only)</td>
<td>52.0</td>
<td><strong>70.8</strong></td>
</tr>
</tbody>
</table>
### Experimental Results: Parsing Results on All-NLI

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean $F$</th>
<th>Self-agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left-Branching</td>
<td>18.9</td>
<td>-</td>
</tr>
<tr>
<td>Right-Branching</td>
<td>18.5</td>
<td>-</td>
</tr>
<tr>
<td>Balanced-Tree</td>
<td>22.0</td>
<td>-</td>
</tr>
<tr>
<td>Gumbel-Tree-LSTM</td>
<td>21.9</td>
<td>56.8</td>
</tr>
<tr>
<td>PRPN</td>
<td>51.6</td>
<td>65.0</td>
</tr>
<tr>
<td>Imitation (first stage only)</td>
<td>52.0</td>
<td><strong>70.8</strong></td>
</tr>
<tr>
<td>Imitation (two stages)</td>
<td><strong>53.7</strong></td>
<td>67.4</td>
</tr>
</tbody>
</table>

More settings and analysis in our paper
Relationship to Previous Studies

Do latent tree learning models identify meaningful structure in sentences? [Williams et al., 2018]

▶ Our results: Yes, but we need a “good” initialization.
Do latent tree learning models identify meaningful structure in sentences?
[Williams et al., 2018]

- Our results: Yes, but we need a “good” initialization.

Tree-Based Neural Sentence Modeling [Shi et al., 2018]: parse/trivial trees are roughly the same for classification performance

- Our results: same findings in terms of NLI accuracy
One last question

Why does NLI help unsupervised parsing?

- Trivial trees (e.g., left/right-branching)
- True parse trees
- Other not understandable trees
One last question

Why does NLI help unsupervised parsing?

NLI Loss

Tree Space

Trivial trees (e.g., left/right-branching)  True parse trees  Other not understandable trees

Second stage policy refinement  After first stage of imitation learning
Conclusion

- Imitation learning for unsupervised parsing
  - A flexible way of coupling heterogeneous models on the intermediate output level
  - Other applications: semantic parsing [Mou et al., 2017], discourse parsing

- Showing the usefulness of semantic tasks for unsupervised parsing

- More research needed on tasks, models, and combinations in this direction
Thank you!

Q&A