Broad-Coverage Transition-Based UCCA Parsing

Daniel Hershcovich$^{1,2}$ & Omri Abend$^2$ & Ari Rappoport$^2$

$^1$Edmond and Lily Safra Center for Brain Sciences
$^2$School of Computer Science and Engineering
Hebrew University of Jerusalem

Learning Club
November 17, 2016
Outline

1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions
Semantic Annotation Schemes

Represent semantic structure of text as a graph. Used by NLP applications for features and structure, providing information such as *who did what to whom?*

Examples:
- Semantic Role Labeling
- Semantic Dependencies
- Abstract Meaning Representation
- Universal Conceptual Cognitive Annotation
Semantic Role Labeling (SRL)

Annotate predicates and their arguments as a flat structure. Examples:

**PropBank**

After graduation, John moved to Paris

- **AM-TMP**: Time
- **A1**: Theme
- **move.01**: Motion
- **A2**: Goal

**FrameNet**

- **After**: Motion Time Theme
- **graduation**: Motion
- **John**: Theme
- **moved**: Motion
- **to Paris**: Goal
Semantic Dependency Parsing (SDP)

Graph on the text tokens, including internal structure of arguments. Examples:

**DELPH-IN MRS-derived bi-lexical dependencies (DM)**

After graduation, John moved to Paris

**Prague Dependency Treebank tectogrammatical layer (PSD)**
Graph on knowledge resource entries inferred from the tokens.

move-01

ARG0

ARG2

person

city

name

op1

Paris

name

name

op1

John

after

time

graduate-01

ARG0

ARG0

Daniel Hershcovich

Broad-Coverage Transition-Based UCCA Parsing
Universal Conceptual Cognitive Annotation (UCCA)

- Cross-linguistically applicable semantic representation scheme.
- Demonstrated applicability to English, French, German & Czech.
- Support for rapid annotation.
- Semantic stability in translation [Sulem et al., 2015].
- Proven useful for machine translation evaluation [Birch et al., 2016].
- Applicability has been so far limited by the absence of a parser.
(1) non-terminal nodes, (2) reentrancy, (3) discontinuity
## UCCA Corpora

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Wiki</th>
<th>Test</th>
<th>20K Leagues</th>
</tr>
</thead>
<tbody>
<tr>
<td># passages</td>
<td>300</td>
<td>34</td>
<td>34</td>
<td>154</td>
</tr>
<tr>
<td># sentences</td>
<td>4267</td>
<td>453</td>
<td>518</td>
<td>506</td>
</tr>
<tr>
<td># nodes</td>
<td>298,665</td>
<td>33,263</td>
<td>37,262</td>
<td>29,315</td>
</tr>
<tr>
<td>% terminal</td>
<td>42.95</td>
<td>43.62</td>
<td>42.89</td>
<td>42.09</td>
</tr>
<tr>
<td>% non-term.</td>
<td>58.30</td>
<td>57.46</td>
<td>58.31</td>
<td>60.01</td>
</tr>
<tr>
<td>% discont.</td>
<td>0.53</td>
<td>0.51</td>
<td>0.47</td>
<td>0.81</td>
</tr>
<tr>
<td>% reentrant</td>
<td>2.31</td>
<td>1.76</td>
<td>2.18</td>
<td>2.03</td>
</tr>
<tr>
<td># edges</td>
<td>287,381</td>
<td>32,015</td>
<td>35,846</td>
<td>27,749</td>
</tr>
<tr>
<td>% primary</td>
<td>98.29</td>
<td>98.81</td>
<td>98.75</td>
<td>97.73</td>
</tr>
<tr>
<td>% remote</td>
<td>1.71</td>
<td>1.19</td>
<td>1.25</td>
<td>2.27</td>
</tr>
</tbody>
</table>

Average per non-terminal node

| # children       | 1.67  | 1.68  | 1.66  | 1.61 |

Excluding root node, implicit nodes, and linkage nodes and edges.
Outline

1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions
Parse sentence $w_1 \ldots w_n$ to graph $G = (V, E, \ell)$ incrementally, using buffer $B$ and stack $S$. Classifier determines transition to apply at each step. Transition-based parsers work by applying a \textit{transition} at each step to the parser state, defined using a buffer $B$ of tokens and nodes to be processed, a stack $S$ of nodes currently being processed, and a graph $G = (V, E, \ell)$ of constructed nodes and edges. A classifier selects the next transition based on the current state’s features. It is trained by an oracle based on gold-standard annotations.

| $S$ | After | graduation | $B$ | John | moved | to | Paris |

\[ G \]

\[ \begin{array}{c}
L \\
H \\
\rightarrow \downarrow \\
After \\
\rightarrow \downarrow \\
graduation
\end{array} \]
Transition-Based Parsing

Transitions: **Shift, Right-Edge_L, Reduce, Shift,**

Figure: Example for intermediate state during transition-based parsing.
Our parser supports the structural properties of UCCA.¹

<table>
<thead>
<tr>
<th>Before Transition</th>
<th>After Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stack</strong></td>
<td><strong>Stack</strong></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>S</td>
<td>x</td>
</tr>
<tr>
<td>S</td>
<td>x</td>
</tr>
<tr>
<td>S</td>
<td>x</td>
</tr>
<tr>
<td>S</td>
<td>y</td>
</tr>
<tr>
<td>S</td>
<td>x</td>
</tr>
<tr>
<td>S</td>
<td>y</td>
</tr>
<tr>
<td>S</td>
<td>x</td>
</tr>
<tr>
<td>S</td>
<td>x</td>
</tr>
<tr>
<td>[root]</td>
<td>∅</td>
</tr>
</tbody>
</table>

**Table:** TUPA transitions. (·, ·)ₓ denotes a primary X-labeled edge, and (·, ·)ₓ a remote X-labeled edge.
TUPA Classifiers

We experiment with three classifiers:

- $\text{TUPA}_{\text{sparse}}$: Perceptron, sparse features: words, POS tags & edge label combinations.
- $\text{TUPA}_{\text{dense}}$: Perceptron, dense embedding features: word2vec [Mikolov et al., 2013] for words, else random.
- $\text{TUPA}_{\text{NN}}$: 2-layer MLP, learned embedding features, logistic activation + dropout.

For all classifiers, inference is performed greedily, i.e., without beam search.
Outline

1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions
Experimental Setup

We conduct our main experiment on the UCCA Wikipedia corpus, and use the English part of the UCCA *Twenty Thousand Leagues Under the Sea* English-French parallel corpus as out-of-domain data.
Evaluation

We report two variants of labeled precision, recall and F-score: one where we consider only primary edges, and another for remote edges. Given graphs $G_p = (V_p, E_p, \ell_p)$ and $G_g = (V_g, E_g, \ell_g)$ over terminals $W = \{w_1, \ldots, w_n\}$, the yield $y(e) \subseteq W$ of an edge $e = (u, v)$ in either graph is the set of terminals in $W$ that are descendants of $v$. The *mutual edges* between the graphs are:

$$M(G_p, G_g) = \{(e_1, e_2) \in E_p \times E_g \mid y(e_1) = y(e_2) \land \ell_p(e_1) = \ell_g(e_2)\}$$

and we define

$$LP = \frac{|M(G_p, G_g)|}{|E_p|} \quad LR = \frac{|M(G_p, G_g)|}{|E_g|} \quad LF = 2 \cdot LP \cdot LR$$
Experiments

Baselines

Since there are no existing UCCA parsers, we use bilexical DAG parsers:

1. Convert UCCA into bilexical dependencies.
2. Train parsers on the resulting training set.
3. Apply trained parsers to the test set.
4. Reconstruct UCCA graphs.

Figure: Bilexical approximation for UCCA graphs.
### Results

$\text{TUPA}_{\text{NN}}$ obtains the highest scores in nearly all metrics:

<table>
<thead>
<tr>
<th></th>
<th>Wiki (in-domain)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Primary</td>
<td></td>
<td>Remote</td>
</tr>
<tr>
<td></td>
<td>LP</td>
<td>LR</td>
<td>LF</td>
</tr>
<tr>
<td>Bilexical Approximation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upper Bound</td>
<td>93.4</td>
<td>83.7</td>
<td>88.3</td>
</tr>
<tr>
<td>DAGParser [Ribeyre et al., 2014]</td>
<td>63.7</td>
<td>56.1</td>
<td>59.5</td>
</tr>
<tr>
<td>TurboParser [Almeida and Martins, 2015]</td>
<td>60.2</td>
<td>47.4</td>
<td>52.9</td>
</tr>
<tr>
<td>Direct Approach</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{TUPA}_\text{sparse}$</td>
<td>64</td>
<td>55.6</td>
<td>59.5</td>
</tr>
<tr>
<td>$\text{TUPA}_\text{dense}$</td>
<td>55</td>
<td>54.8</td>
<td>54.9</td>
</tr>
<tr>
<td>$\text{TUPA}_\text{NN}$</td>
<td><strong>65</strong></td>
<td><strong>62.5</strong></td>
<td><strong>63.7</strong></td>
</tr>
</tbody>
</table>
Tree Approximation

For completeness, we also explore lossily converting UCCA into trees, resulting in a simplified task for the underlying parser, in addition to the maturity of tree-based parsers. Although remote edges are of pivotal importance, exploring tree approximation methods can inform the future development of DAG parsers in general and of UCCA parsers in particular.

<table>
<thead>
<tr>
<th>Constituency Tree Approximation</th>
<th>Upper Bound</th>
<th>100</th>
<th>100</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPARSE [Maier and Lichte, 2016]</td>
<td>63</td>
<td>64.7</td>
<td>63.7</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dependency Tree Approximation</th>
<th>Upper Bound</th>
<th>93.7</th>
<th>83.6</th>
<th>88.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaltParser [Nivre et al., 2007]</td>
<td>64.9</td>
<td>57.9</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>LSTM Parser [Dyer et al., 2015]</td>
<td><strong>74.9</strong></td>
<td><strong>66.4</strong></td>
<td><strong>70.2</strong></td>
<td></td>
</tr>
</tbody>
</table>

Direct Tree Parsing

| TUPA sparse − REMOTE | 65.5 | 57.5 | 61.3 |
Outline

1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions
UCCA-Based Distributed Representation

Vector representation for sentences and documents, based on recursive composition on the UCCA graph.

Impact:
- General automatic semantic feature extractor for text.
- Accurate measure for text similarity.
- Understand the semantic contribution of different elements.
Outline

1. Semantic Annotation Schemes
2. Transition-based UCCA Parsing
3. Experiments
4. Future Work
5. Conclusions
Conclusions

We present TUPA, the first parser for UCCA, and evaluate it in both in-domain and out-of-domain settings, showing it surpasses bilexical DAG parsers on the task of UCCA parsing. Future work will incorporate LSTMs into TUPA, and apply the parser to more languages such as German, demonstrating the importance of broad-coverage parsing. We will also improve the conversion-based methods and explore different target representations. A UCCA parser will enable using the scheme for representation in NLP tasks.

- UCCA exhibits formal properties important for semantic representation.
- We present the first parser for UCCA and the first to support these properties.
Universal Conceptual Cognitive Annotation (UCCA).
In Proc. of ACL, pages 228–238.

Lisbon: Evaluating TurboSemanticParser on multiple languages and out-of-domain data.


Cognitive linguistics.
Cambridge University Press.

Basic linguistic theory.

Transition-based dependency parsing with stack long short-term memory.

Discontinuous parsing with continuous trees.
References II

Efficient estimation of word representations in vector space.

MaltParser: A language-independent system for data-driven dependency parsing.
*Natural Language Engineering, 13(02):95–135.*

Alpage: Transition-based semantic graph parsing with syntactic features.
In *Proc. of SemEval,* pages 97–103, Dublin, Ireland.

Conceptual annotations preserve structure across translations: A French-English case study.
In *Proc. of S2MT,* pages 11–22.