A DOP Active Learning Prototype: interactive treebank annotation and grammar learning

Andreas van Cranenburgh

Heinrich Heine Universität Düsseldorf

December 6, 2017
Successes:

▶ relation to Formal Language Theory (TSG),
▶ efficient implementations (many interesting techniques),
▶ robustness (general property of data-driven statistical parsing, pioneered by DOP)
▶ non-local dependencies (no transformations needed with discontinuous constituents)

Scha (1990) Language theory and language technology; (…)
http://www.remkoscha.nl/LeerdamE.html
27 years of DOP research program

Successes:

▶ relation to Formal Language Theory (TSG),
▶ efficient implementations (many interesting techniques),
▶ robustness (general property of data-driven statistical parsing, pioneered by DOP)
▶ non-local dependencies (no transformations needed with discontinuous constituents)

Open questions:

▶ beyond syntax: semantics, discourse etc.
▶ less ambiguous or more grammatical sentences should be easier/faster to process
▶ acquisition of annotated corpus. DOP model of language acquisition and change.

Scha (1990) Language theory and language technology; (...)
http://www.remkoscha.nl/LeerdamE.html
Annotating. 2 down, 40,000 to go ...

2 (S (NP-SBJ Ms. Haag) (VP plays Elianti)) .

3 Rolls-Royce Motor Cars Inc. said it expects its U.S. sales to remain steady at about 1,200 cars in 1990 .

4 The luxury auto maker last year sold 1,214 cars in the U.S.

5 Howard Mosher, president and chief executive officer, said he anticipates growth for the luxury auto maker in Britain and Europe, and in Far Eastern markets .

6 BELL INDUSTRIES Inc. increased its quarterly to 10 cents from seven cents a share .

7 The new rate will be payable Feb. 15 .

8 A record date hasn't been set .

2,39-40 7

%
Annotating. 2 down, 40,000 to go ...

2 (S (NP-SBJ Ms. Haag) (VP plays Elianti).)

3 Rolls-Royce Motor Cars Inc. said it expects its U.S. sales to remain steady at about 1,200 cars in 1990.

4 The luxury auto maker last year sold 1,214 cars in the U.S.

5 Howard Mosher, president and chief executive officer, said he anticipates growth for the luxury auto maker in Britain and Europe, and in Far Eastern markets.

6 BELL INDUSTRIES Inc. increased its quarterly to 10 cents from seven cents a share.

7 The new rate will be payable Feb. 15.

8 A record date has not been set.

2,39-40 7%
How to acquire an annotated corpus

Raw text is cheap, annotation is costly

**Unsupervised:** U-DOP (Bod 2007); learns unlabeled binary trees from distributional properties of raw text.

**Semi-supervised:** improve a supervised parser with unannotated text; e.g., Deoskar et al (2013): Learning Structural Dependencies of Words in the Zipfian Tail.

**Supervised:** Very labor intensive, requires very special set of skills, costly, boring, tedious, etc.

**Active Learning:** Reduce work load without compromising on annotation quality / detail ⇒ this talk
Unsupervised parsing? (U-DOP)

Bod (ACL 2007): Is the end of supervised parsing in sight?
Unsupervised parsing? (U-DOP)

Bod (ACL 2007): Is the end of supervised parsing in sight?

Betteridge’s law of headlines

“Any headline that ends in a question mark can be answered by the word no.”
Unsupervised parsing? (U-DOP)

Bod (ACL 2007): Is the end of supervised parsing in sight?

Betteridge’s law of headlines

“Any headline that ends in a question mark can be answered by the word no.”

Less flippantly . . . Syntax annotation substantially depends on factors beyond raw text:

- annotation choices (typically 100+ pp. guidelines)
- linguistic theory
- world knowledge
Actual treebank annotation practice

Manual correction of automatic parses in GUI

**PTB:** Deterministic parser (Marcus et al 1993, §4.1). Produces only 1 analysis, only provides bracketings it is confident about.

**FTB:** Rule-based shallow parser; does not attach PPs or relative clauses (Abeille et al 2003, §2.2).

**Tiger:** Brants et al (2004, §3)

- Interactive annotation with Cascaded Markov Model; advantage: responds to user feedback.
- LFG parser, non-interactive post-editing/disambiguation; advantage: always syntactically consistent.
How to optimize use of expert annotators

Interactivity:

Semi-automatic annotation: annotator can use candidate parse(s)
Interactive disambiguation: parser can respond to annotation feedback for current sentence

Active Learning:

Incremental parser training: further automatic parses immediately improve from annotation feedback
Prioritization: Annotate sentences in order that minimizes required user interaction ⇒ learning converges faster
Active Learning

1. Select datapoint that model expects to yield the most improvement. (Training Utility Value)
2. Ask expert to annotate datapoint.
3. Re-train the model.
4. Repeat.

i.e., machine teaching instead of machine learning (http://prodi.gy)

Provides substantial annotation speedup: e.g., 80% reduction in annotation time (Baldridge & Osborne, EMNLP 2004)

Why DOP

Memory-based, “training” is conceptually simple & cheap:
new tree ⇒ extract fragments ⇒ update grammar

Incremental model fitting more challenging/expensive with other methods:
  ▶ Split-merge grammars (EM),
  ▶ Bayesian grammars (Gibbs sampling),
  ▶ Deep Learning (SGD).

Bod (1992); Sangati & Zuidema (EMNLP 2011): 2DOP
Active DOP overview

1. Order sentences by uncertainty of parser (uncertainty sampling)
2. Show n-best parse trees w/current grammar
3. Annotator filters n-best trees with constraints: must have this constituent, cannot have that constituent. Alternatively, manual editing of one of the trees
4. Annotator accepts a tree, added to grammar
5. Rinse, repeat
Ranking sentences

Intuition
Disambiguation is hard when a sentence has many analyses with similar probabilities, so use entropy as Training Utility Value (TUV); Maximizes information gain

1. Compute n-best parse trees with probabilities $p_i$ for a sentence
2. Normalize probabilities because we marginalize over a limited number of derivations (exact DOP parse tree probability is NP-hard)
3. Take entropy of probability distribution $p_1 \ldots p_n$: $- \sum_i p_i \log p_i$
4. Normalize by number of parse trees $n$: $\text{TUV(sent)} = \frac{1}{\log n} \cdot - \sum_i p_i \log p_i$

User interface

- present initial n-best trees
- user filters w/constraints
  or: edit tree manually
- user accepts tree;
  grammar is augmented with fragments of this tree before parsing next sentence
Inspecting a derivation

Fragments used in the highest ranked derivation of this parse tree:
rel. freq: 4/810912; weight: 4.93272e-06

```
ROOT
  \_ SENT
    \_ SENT|<l:VN>
    \  \_ SENT|<m:>
       \  \_ SENT|<r:Ssub>
          \_ ADV^SENT  \_ VN  \_ Ssub  \_ PUNC^.
            \_ ...
            \_ ...
            \_ ...
            \_ .
```

rel. freq: 20/2774.65; weight: 0.00720812

```
ADV^SENT
  \_ Aussi
```

rel. freq: 101/201516; weight: 0.000501201

```
VN
  \_ V^VN  \_ CL^VN
    \_ ...
    \_ -t-il
```

rel. freq: 1/31295.4; weight: 3.19536e-05

```
V^VN
  \_ poussa
```
Augmenting the grammar

Given a new tree $T$ and the current grammar $G$, a multiset of tree fragments.

- extract recurring fragments among initial training set and new tree
- **new fragment** compile into new, unique rules
  **existing fragment** increment relative frequency of existing rules
- bookkeeping: re-normalize grammar, re-sort indexes of rules, etc.

Typically takes $< 1$ second to add 1 parse tree to the grammar.

van Cranenburgh (2014): (tree) fragments (in) linear avg. time
Experimental setup

- initial grammar: DOP grammar of FTB (13k sentences *Le Monde* newspaper)
  
<table>
<thead>
<tr>
<th>F1</th>
<th>POS %</th>
</tr>
</thead>
<tbody>
<tr>
<td>79.3</td>
<td>96.3</td>
</tr>
<tr>
<td>79.0</td>
<td></td>
</tr>
</tbody>
</table>


- 50% split of new trees: extra train trees, test set
Observations about annotation / UI

- n-best list not useful: after verifying part of a tree, want to fix that tree instead of playing “spot the differences” w/rest
- When correct annotation is obvious, editing is faster; re-attaching nodes is quick
- Long sentences don’t fit on screen . . .
- REL, PP errors easy to spot
- Long coordinations tricky; spurious ambiguity of where punctuation is attached
## Evaluation

<table>
<thead>
<tr>
<th>Model, train set</th>
<th>Test set</th>
<th>F1</th>
<th>EX</th>
</tr>
</thead>
<tbody>
<tr>
<td>2DOP, FTB</td>
<td>FTB</td>
<td>79.3</td>
<td>19.9</td>
</tr>
<tr>
<td>2DOP, FTB</td>
<td>Bovary</td>
<td>77.7</td>
<td>22.9</td>
</tr>
<tr>
<td>2DOP, FTB + 100 Bovary trees</td>
<td>Bovary</td>
<td>78.9</td>
<td>23.8</td>
</tr>
</tbody>
</table>

![Graph showing F1 and EX scores](image)

- out-of-domain effect is small: 7% rel. error increase
- 5% relative error reduction from just 100 new trees
Possible improvements

General:
- Better ranking heuristics / sentence selection
- Gamification: maximize inter-annotator agreement
- Efficient workflow; keyboard-based UI

Ideas from previous work:
- Osborne & Baldridge (EMNLP 2004):
  - Use diverse ensemble of parsers
  - Reduce n-best list to a decision tree of annotation choices
- Baldridge & Palmer (EMNLP 2009):
  - Model annotator expertise/fallibility
  - Model cost of annotation given sentence
- Mirroshandel & Nasr (IWPT 2011):
  - Rank per-token uncertainty instead of by sentence
Wild ideas

- Bootstrap a new treebank when no initial grammar is available? (endangered / low-resource languages)
- Add new levels of annotation to an existing treebank? e.g.,
  - discontinuous constituents,
  - multi-word expressions
- Joint annotation of constituency and dependency structures?
- Grammar engineering instead of treebank annotation; e.g., LTAG, RRG
Yes, we can ...
Conclusion

Yes, we can . . .

speed up annotation w/DOP

- Encouraging results:
  - Literary, out-of-domain text parsed relatively well
  - Small number of annotations already improve accuracy

- More comprehensive experiments needed to see to what extent incremental learning really helps

Code will be made available at
http://github.com/andreasvc/disco-dop