Structured Training for Neural Network Transition-Based Parsing

David Weiss, Chris Alberti, Michael Collins, Slav Petrov

Presented by: Shayne Longpre
What is SyntaxNet?

- 2016/5: Google announces the “World’s Most Accurate Parser Goes Open Source”
- Now supports 40 languages -- Parsey McParseface’s 40 ‘cousins’
What is SyntaxNet?

- 2016/5: Google announces the “World’s Most Accurate Parser Goes Open Source”
- Now supports 40 languages -- Parsey McParseface’s 40 ‘cousins’


+ Unlabelled Data
+ Tune Model
+ Structured Perceptron & Beam Search
+ Global Normalization

SyntaxNet
What is SyntaxNet?

- 2016/5: Google announces the “World’s Most Accurate Parser Goes Open Source”
- Now supports 40 languages -- Parsey McParseface’s 40 ‘cousins’

Chen & Manning (2014)  
Weiss et al. (2015)  
Andor et al. (2016)

+ Unlabelled Data  
+ Tune Model  
+ Structured Perceptron & Beam Search  
+ Global Normalization

SyntaxNet
3 New Contributions

1. Leverage **Unlabelled Data** -- “Tri-Training”

2. **Tuned** Neural Network Model

3. **Final Layer:** **Structured Perceptron w/ Beam Search**
1. Tri-Training: Leverage Unlabelled Data

High Performance Parser (A)

Agree on dependency parse

High Performance Parser (B)

Disagree on dependency parse

Unlabelled Data

“Tri-Training” (Li et al, 2014)

Labelled Data
2. Model Changes

Chen & Manning (2014):
Weiss et al. (2015):

Chen & Manning (2014):
2. Model Changes

Chen & Manning (2014):

Weiss et al. (2015):

- **Perceptron Layer**
  \[
  \hat{y} = \arg\max_{y \in \text{GEN}(x)} \sum_{j=1}^{m} v(y_j) \cdot \phi(x, c_j)
  \]

- **Softmax Layer**
  \[
  P(y) \propto \exp\{\beta^T h_2 + b_y\}
  \]

- **Hidden Layers**
  \[
  h_2 = \max\{0, W_2 h_1 + b_2\}
  \]

- **Embedding Layer**
  \[
  h_0 = [X \cdot E_g] \quad \forall g \in \{\text{word, tag, label}\}
  \]

- **Input**
  - The DT, news NN
  - had VBD
  - little JJ
  - effect NN

- **Features Extracted**
  - \(s_i, b_i\)
  - \(lc_1(s_i), lc_2(s_i)\)
  - \(rc_1(s_i), rc_2(s_i)\)
  - \(rc_1(rc_1(s_i))\)
  - \(lc_1(lc_2(s_i))\)
Problem: Greedy algorithms are unable to look beyond one step ahead, or recover from incorrect decisions.
3. Structured Perceptron Training + Beam Search

**Problem:** Greedy algorithms are unable to look beyond one step ahead, or recover from incorrect decisions.

**Solution:** Look forward -- search the tree of possible transition sequences.

![Diagram of a tree representing possible transition sequences](image)
3. Structured Perceptron Training + Beam Search

**Problem:** Greedy algorithms are unable to look beyond one step ahead, or recover from incorrect decisions.

**Solution:** Look forward -- search the tree of possible transition sequences.

- Keep track of $K$ top partial transition sequences up to depth $m$.
- Score transition using perceptron:

$$\arg\max_{y \in \text{GEN}(x)} \sum_{j=1}^{m} v(y_j) \cdot \phi(x, y_1 \ldots y_{j-1}).$$

*Figura 1 – Árvore de busca utilizando o beam search*
3. Structured Perceptron Training + Beam Search

**Problem:** Greedy algorithms are unable to look beyond one step ahead, or recover from incorrect decisions.

**Solution:** Look forward -- search the tree of possible transition sequences.

- Keep track of $K$ top partial transition sequences up to depth $m$.

- Score transition using perceptron:

$$\arg\max_{y \in \text{GEN}(x)} \sum_{j=1}^{m} v(y_j) \cdot \phi(x, y_1 \ldots y_{j-1}).$$

Figura 1 – Árvore de busca utilizando o beam search
Identify specific flaws in existing models (greedy algorithms) and solve them. In this case, with:
- More data
- Better tuning
- Structured perceptron and beam search

Final step to SyntaxNet: Andor et al. (2016) solve the “Label Bias Problem” using Global Normalization