Transition-Based Parsing

Based on a Tutorial at COLING-ACL, Sydney 2006
with Joakim Nivre

Sandra Kübler, Markus Dickinson

Indiana University
E-mail: skuebler,md7@indiana.edu
Dependency Parsing

- grammar-based parsing
- data-driven parsing

- learning: given a training dataset $C$ of sentences (annotated with dependency graphs), induce a parsing model $M$ that can be used to parse new sentences
- parsing: given a parsing model $M$ and a sentence $S$, derive the optimal dependency graph $G$ for $S$ based on $M$
Deterministic Parsing

- **Basic idea:**
  - Derive a single syntactic representation (dependency graph) through a deterministic sequence of elementary parsing actions
  - Sometimes combined with backtracking or repair

- **Motivation:**
  - Psycholinguistic modeling
  - Efficiency
  - Simplicity
Covington’s Incremental Algorithm

- Deterministic incremental parsing in $O(n^2)$ time by trying to link each new word to each preceding one:

\[
\text{PARSE}(x = (w_1, \ldots, w_n))
\]
\[
1 \quad \text{for } i = 1 \text{ up to } n
\]
\[
2 \quad \text{for } j = i - 1 \text{ down to } 1
\]
\[
3 \quad \text{LINK}(w_i, w_j)
\]

\[
\text{LINK}(w_i, w_j) = \begin{cases} 
E \leftarrow E \cup (i, j) & \text{if } w_j \text{ is a dependent of } w_i \\
E \leftarrow E \cup (j, i) & \text{if } w_i \text{ is a dependent of } w_j \\
E \leftarrow E & \text{otherwise}
\end{cases}
\]

- Different conditions, such as Single-Head and Projectivity, can be incorporated into the LINK operation.
Shift-Reduce Type Algorithms

- Data structures:
  - Stack \([\ldots, w_i]_S\) of partially processed tokens
  - Queue \([w_j, \ldots]_Q\) of remaining input tokens

- Parsing actions built from atomic actions:
  - Adding arcs \((w_i \rightarrow w_j, w_i \leftarrow w_j)\)
  - Stack and queue operations

- Left-to-right parsing in \(O(n)\) time

- Restricted to projective dependency graphs
Yamada’s Algorithm

- Three parsing actions:
  - **Shift**
    
    $\begin{array}{c}
    \ldots s \quad [w_i, \ldots] Q \\
    [\ldots, w_i] s \quad [\ldots] Q
    \end{array}$

  - **Left**
    
    $\begin{array}{c}
    [\ldots, w_i, w_j] s \quad [\ldots] Q \\
    [\ldots, w_i] s \quad [\ldots] Q \\
    \quad w_i \rightarrow w_j
    \end{array}$

  - **Right**
    
    $\begin{array}{c}
    [\ldots, w_i, w_j] s \quad [\ldots] Q \\
    [\ldots, w_j] s \quad [\ldots] Q \\
    \quad w_i \leftarrow w_j
    \end{array}$

- Algorithm variants:
  - Originally developed for Japanese (strictly head-final) with only the **Shift** and **Right** actions
  - Adapted for English (with mixed headedness) by adding the **Left** action
  - Multiple passes over the input give time complexity $O(n^2)$
Nivre’s Algorithm

- Four parsing actions:

  **Shift**
  \[
  \begin{array}{c}
  \ldots \text{s} \quad [w_i, \ldots] \text{Q} \\
  \ldots, w_i \text{s} \quad [\ldots] \text{Q}
  \end{array}
  \]

  **Reduce**
  \[
  \begin{array}{c}
  \ldots, w_i \text{s} \quad [\ldots] \text{Q} \\
  \exists w_k : w_k \rightarrow w_i
  \end{array}
  \]

  **Left-Arc**
  \[
  \begin{array}{c}
  \ldots, w_i \text{s} \quad [w_j, \ldots] \text{Q} \\
  \neg \exists w_k : w_k \rightarrow w_i
  \end{array}
  \]

  **Right-Arc**
  \[
  \begin{array}{c}
  \ldots, w_i, w_j \text{s} \quad [\ldots] \text{Q} \\
  \neg \exists w_k : w_k \rightarrow w_j
  \end{array}
  \]

- Characteristics:
  - Integrated labeled dependency parsing
  - Arc-eager processing of right-dependents
  - Single pass over the input gives time complexity \( O(n) \)
Economic news had little effect on financial markets.
Example

[\text{root} \ Economic]_S \ [\text{news had little effect on financial markets} \ .]_Q

Shift
Economic news had little effect on financial markets.
Example

[nmod

 locally Economic news[[S had little effect on financial markets .]]Q

Shift
Example

\[
\begin{array}{c}
\text{[root]}_s \quad \text{Economic news} \quad [\text{had little effect on financial markets}.]_Q \\
\end{array}
\]

Left-Arc_{sbj}
Economic news had little effect on financial markets.
Example

Economic news had little\[effect on financial markets .\]
Example

Economic news had little effect on financial markets.

Left-Arc$_{nmod}$
Economic news had little effect on financial markets.

Right-Arc$_{obj}$
Economic news had little effect on financial markets.
Economic news had little effect on financial markets.
Example

Economic news had little effect on financial markets.
Economic news had little effect on financial markets.

Right-Arc\textsubscript{pc}
Economic news had little effect on financial markets.
Example

Economic news had little effect on financial markets.

Reduce
Economic news had little effect on financial markets.
Economic news had little effect on financial markets.
Economic news had little effect on financial markets.
Classifier-Based Parsing

Data-driven deterministic parsing:
- Deterministic parsing requires an oracle.
- An oracle can be approximated by a classifier.
- A classifier can be trained using treebank data.

Learning methods:
- Memory-based learning (MBL) (Nivre et al. 2004)
- Maximum entropy modeling (MaxEnt) (Cheng et al. 2005)
Feature Models

Learning problem:
- Approximate a function from parser states, represented by feature vectors to parser actions, given a training set of gold standard derivations.

Typical features:
- Tokens:
  - Target tokens
  - Linear context (neighbors in $S$ and $Q$)
  - Structural context (parents, children, siblings in $G$)
- Attributes:
  - Word form (and lemma)
  - Part-of-speech (and morpho-syntactic features)
  - Dependency type (if labeled)
  - Distance (between target tokens)
There is an ideal shit, of which all the shit that happens is but an imperfect image.

(Plato)
There is an ideal shit, of which all the shit that happens is but an imperfect image.

(Plato)
There is an ideal shit, of which all the shit that happens is but an imperfect image.

(Plato)