Dependency Parsing and Semantic Role Labeling

Steven Bethard
for Jinho D. Choi
University of Colorado Boulder
Dependency-based SRL

• Semantic role labeling
  - Task of identifying arguments of each predicate and labeling them with semantic roles in relation to the predicate.

• Dependency-based semantic role labeling
  - Advantages over constituent-based semantic role labeling.
    • Dependency parsing is faster (2.29 milliseconds / sentence).
    • Dependency structure is more similar to predicate argument structure.
  - Labels headwords instead of phrases.
    • Still can recover the original semantic chunks for the most of time (Choi and Palmer, LAW 2010).
Dependency-based SRL

- Constituent-based vs. dependency-based SRL

He opened the door with his foot at ten

Diagram showing the dependency structure with labels for Agent, Theme, Instrument, and Temporal.
Dependency-based SRL

- Constituent-based vs. dependency-based SRL

He opened the door with his foot at ten.

ARG_0: He
ARG_1: the door
ARG_2: with his foot
SBJ: He
OBJ: the door
ADV: at
TMP: ten
Transition-based SRL

- Parsing states
  - \((\lambda_1, \lambda_2, p, \lambda_3, \lambda_4, A)\)
  - \(p\) - index of the current predicate candidate.
  - \(\lambda_1\) - indices of lefthand-side argument candidates.
  - \(\lambda_4\) - indices of righthand-side argument candidates.
  - \(\lambda_{2,3}\) - indices of processed tokens.
  - \(A\) - labeled arcs with semantic roles

- Initialization: \(([\phantom{1}], [\phantom{1}], 1, [\phantom{1}], [2, ..., n], \emptyset)\)

- Termination: \((\lambda_1, \lambda_2, \emptyset, [\phantom{1}], [\phantom{1}], A)\)
Transition-based SRL

- Transitions
  - **No-Pred** - finds the next predicate candidate.
  - **No-Arc** ← - rejects the lefthand-side argument candidate.
  - **No-Arc** → - rejects the righthand-side argument candidate.
  - **Left-Arc** ← - accepts the lefthand-side argument candidate.
  - **Right-Arc** → - accepts the righthand-side argument candidate.
Features

- Baseline features
  - N-gram and binary features (similar to ones in Johansson and Nugues, EMNLP 2008).
  - Structural features.

Subcategorization of “wants”

Path from “John” to “buy”

Depth from “John” to “buy”

1 ↑ LCA ↓ 2
Features

• Dynamic features
  - Derived from previously identified arguments.
  - Previously identified argument label of $w_{arg}$.
  - Label of the very last predicted numbered argument of $w_{pred}$.
  - These features can narrow down the scope of expected arguments of $w_{pred}$.
Summary

• Introduced a transition-based SRL algorithm, showing near state-of-the-art results.
  - No need to design separate systems for argument identification and classification.
  - Make it easier to develop a joint-inference system between dependency parsing and semantic role labeling.

• Future work
  - Several techniques, designed to improve transition-based parsing, can be applied (e.g., dynamic programming, k-best ranking)
  - We can apply more features, such as clustering information, to improve labeling accuracy.