Statistical NLP
Spring 2011

Berkeley

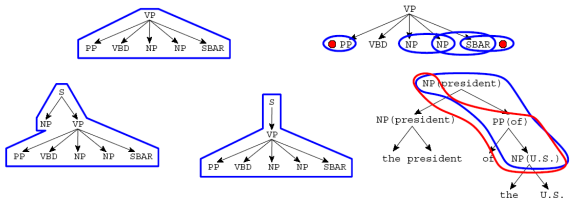
N L P

Lecture 18: Parsing IV

Dan Klein – UC Berkeley

Parse Reranking

- Assume the number of parses is very small
- We can represent each parse T as an arbitrary feature vector φ(T)
  - Typically, all local rules are features
  - Also non-local features, like how right-branching the overall tree is
  - [Charniak and Johnson 05] gives a rich set of features

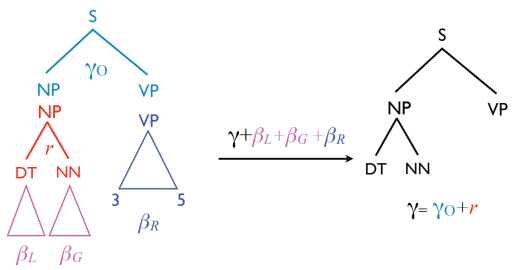


Inside and Outside Scores

Inside and Outside Scores

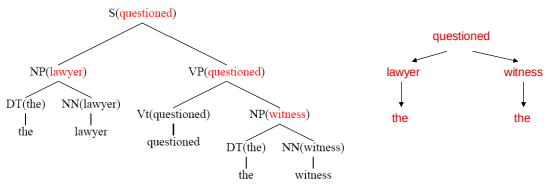
K-Best Parsing

[Huang and Chiang 05, Pauls, Klein, Quirk 10]



Dependency Parsing

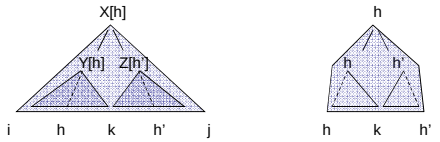
- Lexicalized parsers can be seen as producing *dependency trees*



- Each local binary tree corresponds to an attachment in the dependency graph

Dependency Parsing

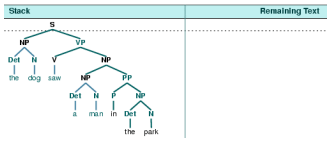
- Pure dependency parsing is only cubic [Eisner 99]



- Some work on *non-projective* dependencies
  - Common in, e.g. Czech parsing
  - Can do with MST algorithms [McDonald and Pereira 05]

Shift-Reduce Parsers

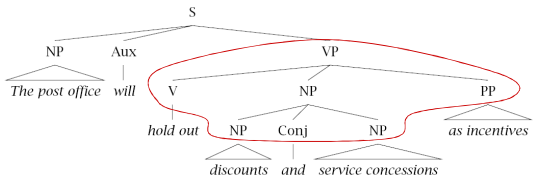
- Another way to derive a tree:



- Parsing
  - No useful dynamic programming search
  - Can still use beam search [Ratnaparkhi 97]

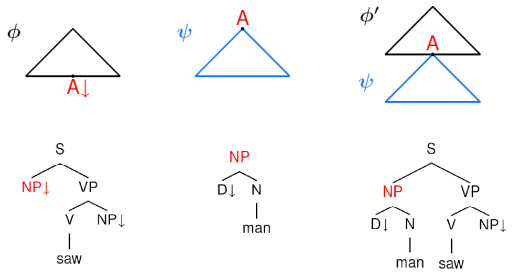
Data-oriented parsing:

- Rewrite large (possibly lexicalized) subtrees in a single step



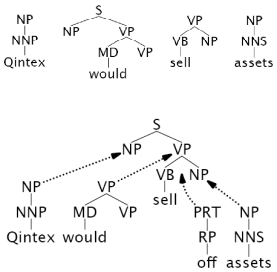
- Formally, a *tree-insertion grammar*
- Derivational ambiguity whether subtrees were generated atomically or compositionally
- Most probable *parse* is NP-complete

TIG: Insertion

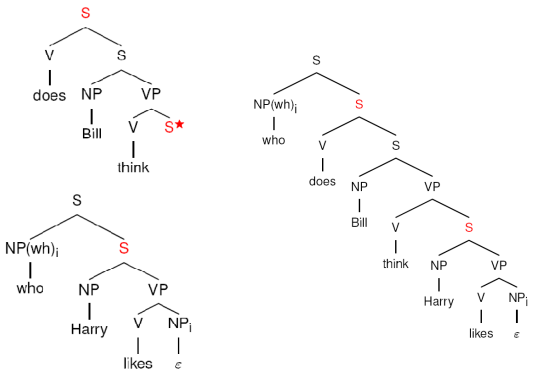


Tree-adjoining grammars

- Start with *local trees*
- Can insert structure with *adjunction* operators
- Mildly context-sensitive
- Models long-distance dependencies naturally
- … as well as other weird stuff that CFGs don't capture well (e.g. cross-serial dependencies)



TAG: Long Distance



CCG Parsing

- Combinatory Categorial Grammar
  - Fully (mono-) lexicalized grammar
  - Categories encode argument sequences
  - Very closely related to the lambda calculus (more later)
  - Can have spurious ambiguities (why?)

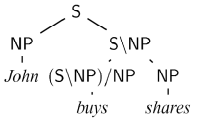
John ⊢ NP

shares ⊢ NP

buys ⊢ (S\NP)/NP

sleeps ⊢ S\NP

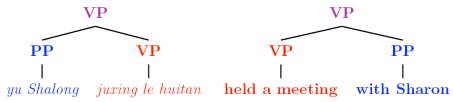
well ⊢ (S\NP)\(S\NP)



Syntax-Based MT

- synchronous context-free grammars (SCFGs)
  - context-free grammar in two dimensions
  - generating pairs of strings/trees simultaneously
  - co-indexed nonterminal further rewritten as a unit

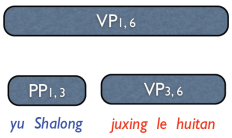
VP → PP(1) VP(2), VP(2) PP(1)
VP → *juxing le huitan*, held a meeting
PP → *yu Shalong*, with Sharon



Translation by Parsing

- translation with SCFGs => monolingual parsing
- parse the source input with the source projection
  - build the corresponding target sub-strings in parallel

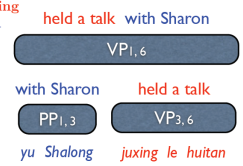
VP → PP(1) VP(2),
VP → *juxing le huitan*,
PP → *yu Shalong*,



Translation by Parsing

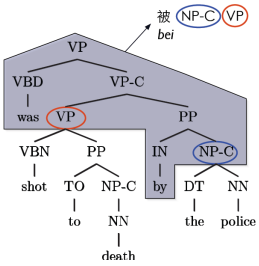
- translation with SCFGs => monolingual parsing
- parse the source input with the source projection
  - build the corresponding target sub-strings in parallel

VP → PP(1) VP(2), VP(2) PP(1)
VP → *juxing le huitan*, held a meeting
PP → *yu Shalong*, with Sharon



Learning MT Grammars

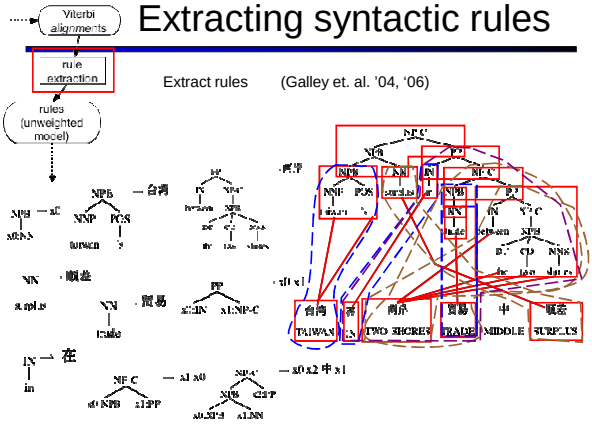
- syntax-directed, English to Chinese (Huang, Knight, Joshi, 2006)
- first parse input, and then recursively transfer



synchronous tree-substitution grammars (STSG)

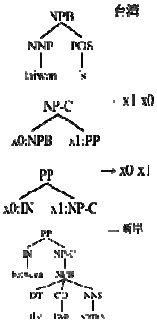
(Galley et al., 2004; Eisner, 2003)

Extracting syntactic rules



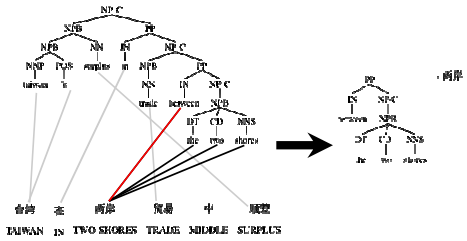
Extract rules (Galley et. al. '04, '06)

Rules can...



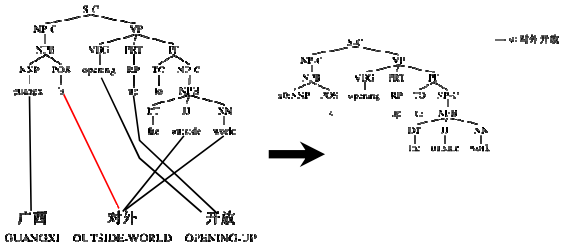
- capture phrasal translation
- reorder parts of the tree
- traverse the tree without reordering
- insert (and delete) words

Bad alignments make bad rules



This isn't very good, but let's look at a worse example...

Sometimes they're really bad



One bad link makes a totally unusable rule!