Last Class:

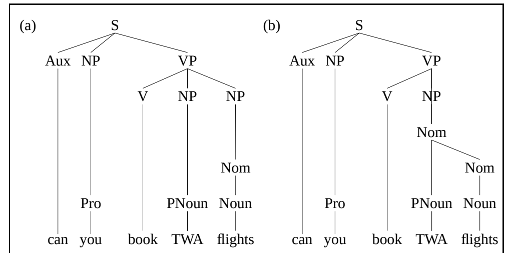
1. The Earley Algorithm
2. Intro to Probabilistic Parsing

Today:

1. Parsing with PCFG's
2. Intro to Question Answering

Slide CS474–1

Example



Rules			P
S	→	Aux NP VP	.15
NP	→	Pro	.40
VP	→	V NP NP	.05
NP	→	Nom	.05
NP	→	PNoun	.35
Nom	→	Noun	.75
Aux	→	Can	.40
NP	→	Pro	.40
Pro	→	you	.40
Verb	→	book	.30
PNoun	→	TWA	.40
Noun	→	flights	.50

Rules			P
S	→	Aux NP VP	.15
NP	→	Pro	.40
VP	→	V NP	.40
NP	→	Nom	.05
Nom	→	PNoun Nom	.05
Nom	→	Noun	.75
Aux	→	Can	.40
NP	→	Pro	.40
Pro	→	you	.40
Verb	→	book	.30
Pnoun	→	TWA	.40
Noun	→	flights	.50

Slide CS474–2

Parsing with PCFGs

Produce the most likely parse for a given sentence:

$$\hat{T}(S) = argmax_{T \in \tau(S)} P(T)$$

where $\tau(S)$ is the set of possible parse trees for S.

- Augment the Earley algorithm to compute the probability of each of its parses.

  When adding an entry E of category C to the chart using rule i with n subconstituents, $E_1, \ldots, E_n$:

  $P(E) = P(rule\ i \mid C) * P(E_1) * \ldots * P(E_n)$
- probabilistic CYK (Cocke-Younger-Kasami) algorithm

Slide CS474–3

Problems with PCFGs

Do not model *structural dependencies.*

Often the choice of how a non-terminal expands depends on the location of the node in the parse tree.

E.g. Strong tendency in English for the syntactic subject of a spoken sentence to be a pronoun.

- 91% of declarative sentences in the Switchboard corpus are pronouns (vs. lexical).
- In contrast, 34% of direct objects in Switchboard are pronouns.

Slide CS474–4

Problems with PCFGs

Do not adequately model *lexical dependencies*.

Moscow sent more than 100,000 soldiers into Afghanistan...

PP can attach to either the NP or the VP:
NP $\rightarrow$ NP PP or VP $\rightarrow$ V NP PP?

Attachment choice depends (in part) on the verb: *send* subcategorizes for a destination (e.g. expressed via a PP that begins with *into* or *to* or ...).

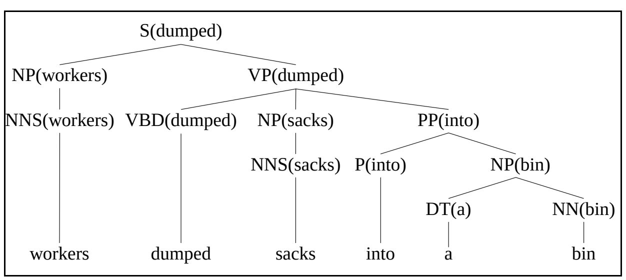
Slide CS474–5

Probabilistic lexicalized CFGs

- Each non-terminal is associated with its head.
- Each PCFG rule needs to be augmented to identify one rhs constituent to be the head daughter.
- Headword for a node in the parse tree is set to the headword of its head daughter.

Slide CS474–6

Example



Slide CS474–7

Probabilistic lexicalized CFGs

View a lexicalized (P)CFG as a simple (P)CFG with a lot more rules.

VP(dumped) $\rightarrow$ VBD(dumped) NP(sacks) PP(into) [$3\text{x}10^{-10}$]
VP(dumped) $\rightarrow$ VBD(dumped) NP(cats) PP(into) [$8\text{x}10^{-10}$]
VP(dumped) $\rightarrow$ VBD(dumped) NP(sacks) PP(above) [$1\text{x}10^{-12}$]
...

Problem?

Slide CS474–8

Incorporating lexical dependency information

Incorporates lexical dependency information by:

1. relating the heads of phrases to the heads of their constituents;
2. including syntactic subcategorization information.

Syntactic subcategorization dependencies:

Probability of a rule r of syntactic category n:
p(r(n) | n, h(n)).

Example: probability of expanding VP as VP $\rightarrow$ VBD NP PP will be p (r | VP, dumped).

Incorporating lexical dependency information

Condition the probability of a node n having a head h on two factors:

1. the syntactic category of the node n
2. the head of the node's mother $h(m(n))$

p(h(n) = word_i | n, h(m(n)))

Computing the probability of a parse

Computing the probability of a particular parse for a given sentence changes from:

$P(T) = \prod_{n \in T} p(r(n))$

to

$P(T) = \prod_{n \in T} p(r(n)|n,h(n)) * p(h(n)|n,h(m(n)))$

Evaluation Measures and State of the Art

- labeled recall: # correct constituents in candidate parse of s / # correct constituents in treebank parse of s
- labeled precision: # correct constituents in candidate parse of s / total # of constituents in candidate parse of s
- crossing brackets: the number of crossed brackets

State of the art: 91-92% recall/, 1% crossed bracketed constituents per sentence (WSJ treebank)