Probabilistic CFGs

Augments each rule in $P$ with a conditional probability:

$$ A \rightarrow \beta [p] $$

where $p$ is the probability that the non-terminal $A$ will be expanded to the sequence $\beta$. Often referred to as

$$ P(A \rightarrow \beta) \text{ or } P(A \rightarrow \beta|A). $$
Why are PCFGs useful?

- Assigns a probability to each parse tree $T$
- Useful in disambiguation
  - Choose the most likely parse
  - Computing the probability of a parse
    
    If we make independence assumptions, $P(T) = \prod_{n \in T} p(r(n))$.
- Useful in language modeling tasks

Where do the probabilities come from?

1. from a treebank:

\[ P(\alpha \rightarrow \beta | \alpha) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)} \]

2. use EM (forward-backward algorithm, inside-outside algorithm)

Parsing with PCFGs

Produce the most likely parse for a given sentence:

\[ \hat{T}(S) = \arg \max_{T \in \tau(S)} P(T) \]

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, $E_1, \ldots, E_n$:

\[ P(E) = P(\text{rule } i \mid C) \ast P(E_1) \ast \ldots \ast P(E_n) \]

- probabilistic CYK (Cocke-Younger-Kasami) algorithm
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.

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

PP can attach to either the NP or the VP:

NP \rightarrow NP \text{PP} \text{ or } VP \rightarrow V NP \text{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 ...).

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.

Example

\[
\begin{array}{c}
S(\text{dumped}) \\
\text{NP(workers)} & \text{VP(dumped)} \\
\text{NNS(workers)} & \text{VBD(dumped)} & N\text{P(sacks)} \\
\text{PP(\text{into})} & \text{NNS(sacks)} & \text{P(\text{into})} \\
\text{DT(a)} & \text{NN(bin)} & \text{NP(bin)} \\
\end{array}
\]

workers dumped sacks into a bin
Probabilistic lexicalized CFGs
View a lexicalized (P)CFG as a simple (P)CFG with a lot more rules.

VP(dumped) → VBD(dumped) NP(sacks) PP(into) [3x10^{-10}]
VP(dumped) → VBD(dumped) NP(cats) PP(into) [8x10^{-10}]
VP(dumped) → VBD(dumped) NP(sacks) PP(above) [1x10^{-12}]
...

Problem?

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 → 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
Producing the most likely 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: 90% recall, 90% precision, 1% crossed bracketed constituents per sentence (WSJ treebank)