

1 Accurate Parsing is Computationally Challenging

A CFG for English might generate the phrase “that grand old house on the hill” using the following grammar.

$$\begin{aligned} NP &\rightarrow Det\ N \\ Det &\rightarrow \text{that} \\ Det &\rightarrow \text{the} \\ N &\rightarrow AP\ N \\ AP &\rightarrow \text{grand} \\ AP &\rightarrow \text{old} \\ N &\rightarrow N\ PP \\ N &\rightarrow \text{house} \\ PP &\rightarrow P\ NP \\ P &\rightarrow \text{on} \\ N &\rightarrow \text{hill} \end{aligned}$$

But the parsing accuracy of PCFGs can be improved by lexicalizing the nonterminals. This means a nonterminal is specified by a pair of a syntactic category and a particular word. For example NP_{house} is a nonterminal which can generate an noun phrase whose head word is the word “house”. The concept of “head word” is not formally defined, but intuitively the head word is the “most important word” in a phrase. In a lexicalized PCFG the phrase “that grand old house on the hill” might be generated using the following productions.

$$\begin{aligned} NP_{\text{house}} &\rightarrow Det_{\text{that}} N_{\text{house}} \\ Det_{\text{that}} &\rightarrow \text{that} \\ N_{\text{house}} &\rightarrow AP_{\text{grand}} N_{\text{house}} \\ AP_{\text{grand}} &\rightarrow \text{grand} \end{aligned}$$

$$\begin{aligned}
N_{\text{house}} &\rightarrow AP_{\text{old}}N_{\text{house}} \\
AP_{\text{old}} &\rightarrow \text{old} \\
N_{\text{house}} &\rightarrow N_{\text{house}}PP_{\text{on}} \\
N_{\text{house}} &\rightarrow \text{house} \\
&\vdots
\end{aligned}$$

Parsing a lexicalized PCFG is challenging. Even if we restrict the head words to the words that occur in the sentence, the average sentence length in the New York times is 25 words. Combining that with 20 syntactic categories gives 460 nonterminals. The viterbi algorithm considers all possible pairs of adjacent phrases — this approximately 25^3 times 460^2 — about 4 billion phrase pairs. Many parsers use even more information in the nonterminals making complete Viterbi parsing impossible.

2 Probabilities to Weights

Consider a probabilistic context free grammar (PCFG) in Chomsky normal form. We have productions of the form $X \rightarrow YZ$ and $X \rightarrow a$ and for each nonterminal X we have $\sum_{\beta} P(X \rightarrow \beta) = 1$. We define the weight a production as follows.

$$w(X \rightarrow \beta) = \log_2 \frac{1}{P(X \rightarrow \beta)}$$

We can think of $w(X \rightarrow \beta)$ as a number of bits. The weight of a parse tree is the sum of the weights of the productions used in that parse tree. The weight of a tree can be thought of as the number of bits it takes to name (or code for) that tree. We are interested in finding the Viterbi parse tree — the most probably tree, or equivalently, the lightest weight tree.

3 Dijkstra Lightest Derivation

The efficiency of Parsing can be improved by using algorithms similar to Dijkstra shortest path and A*.

We consider the following inference rule for deriving weighted phrases from weighted phrases.

$$\begin{array}{l}
X \rightarrow YZ, w_r \\
Y_{i,j}, w_1 \\
Z_{j,k}, w_2 \\
\hline
X_{i,k}, w_1 + w_2 + w_r
\end{array}$$

Please compare the following algorithm to the Dijkstra shortest path algorithm.

1. Initialize Q to be the set containing all pairs $\langle X_{i,i+1}, w \rangle$ where the grammar contains $X \rightarrow a$ with weight w and the i th word in the input string is a .
2. Initialize S to be the empty set.
3. If Q is empty terminate with failure (there is no path from s to g).
4. Remove a pair $\langle Y_{i,j}, w \rangle$ from Q with minimum weight w (over all elements of Q).
5. If S already contains a weight for $Y_{i,j}$, i.e., if S contains $\langle Y_{i,j}, w' \rangle$ for some w' , then continue again from step 3.
6. If $Y_{i,j} = S_{1,n}$ then terminate and output w as the weight of the lightest weight derivation.
7. Otherwise, add $\langle Y_{i,j}, w \rangle$ to S .
8. Add to Q all new weighted phrases $\langle X_{i,k}, w \rangle$ or $\langle X_{k,j}, w \rangle$ that can be derived from $\langle Y_{i,j}, w \rangle$ using the above rule and other phrases in S .
9. repeat from step 4.

Note that if removing $\langle Y_{i,j}, w \rangle$ from Q results in adding $\langle X_{u,v}, w' \rangle$ then, because weights of grammar rules are non-negative, we have that $w' \geq w$. This implies that the weights removed from Q monotonically increase. This implies that when a pair $\langle X_{u,v}, w \rangle$ is added to S , w is the weight of the lightest weight derivation of $X_{u,v}$. Note that if the given word string has a light weight parse tree then heavy weight phrases are never considered.