CSME 206A – Natural Language & Speech Processing **Spring Semester**

Lecture 15: Parsing context-free grammars

Lecturer: K.R. Chowdhary *: Professor of CS*

15.1 Parsing Context-Free Grammars

The parse-trees are useful for:

1. Grammar checking of the sentence,
2. Parsing is an important intermediate stage in semantic analysis.
3. The parsing plays an important role in the following areas of NLP:
   (a) Mechanical translation,
   (b) Question answering
   (c) Information Extraction

A syntactic parser can be viewed as searching through the space of all possible parse-trees to find the correct parse-tree. Before we go through the steps of parsing, let us consider the following rules for grammar, where S is sentence, NP is noun-phrase, VP is verb-phrase, Aux is auxiliary verb, Det is determiner, Nom is nomial, PP is preposition phrase, $Prop\text{-}N$ is proper noun, and $Prep$ is preposition.

15.1.1 Top-Down parsing

Input to the parser is not the actual words of the sentence, but rather the output of the tagger. Thus, in case of *single tagger*, the parser's input is a sequence of tags, one corresponding to each of the words in the sentence. For *multiple tagger*, we have not a single tag corresponding to a word, but instead a set of tags, namely the tags that the multiple tagger reported for that word. In the extreme case, when tagger is not used, each word has all possible tags, this is called *all tagger*.

When parser gets more than one possible tags, it constructs all parses with all possible combinations of tags. That is to say, the parser does not agree on the tags for the words. For example, in the standard sentence, "Time flies like arrow," the first meaning is: "time" as noun, while second is: "time the flies like arrow". Hence, the "time" has sense of verb, and it means "govern or command the time for the flies (noun)", i.e, "flies" is considered as noun. In other words, it commands the listener to measure how long flies take to do some thing. Hence, this tags "time" as verb. It is possible, however, for the parser to (in effect) pickup a single preferred set of tags by choosing those used in the most probable parse. Thus, if "time is fleeting" (lasting for a very short time) then most probable tag for "time" is noun.

The parse-tree based method uses two approaches. One is *top-down* where an edge starting at location l is not produced unless some possible parse of the words w_1 to w_{l-1}, written as $w_{1,l-1}$, could use an edge

starting at location l. In addition, several *would be* edges are collapsed into one when they are identical except for their predictions of subsequent constituents. This reduces the number of edges.

15.1.2 Demonstrating Top-down parsing

The searching is carried out from the root node. The substitutions are carried out, and progressing sentence is compared with the input text sentence to determine whether the sentence generated progressively matches with the original. The figure 15.1 demonstrates the steps for the top-down parsing for the sentence "Book that flight."

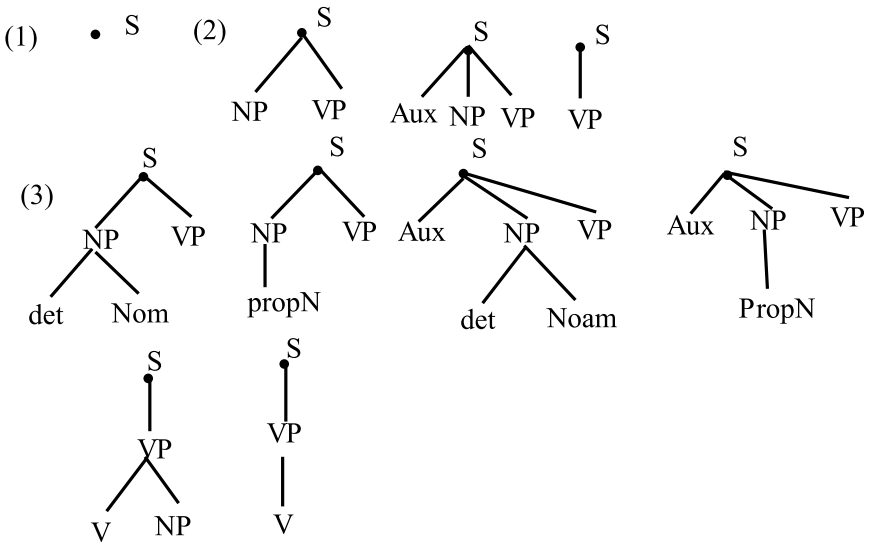


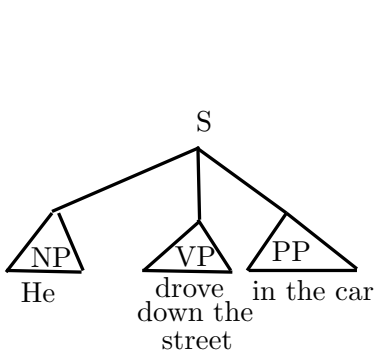
Figure 15.1: Top-down parsing of: "Book that flight"

To carry out the top down parsing, we expand the tree at each level as shown in the figure. At each level, the trees whose leaves fails to match the input sentence, are rejected, leaving behind the trees that represent the successful parses. Going this way, ultimately get the sentence: "Book that flight."

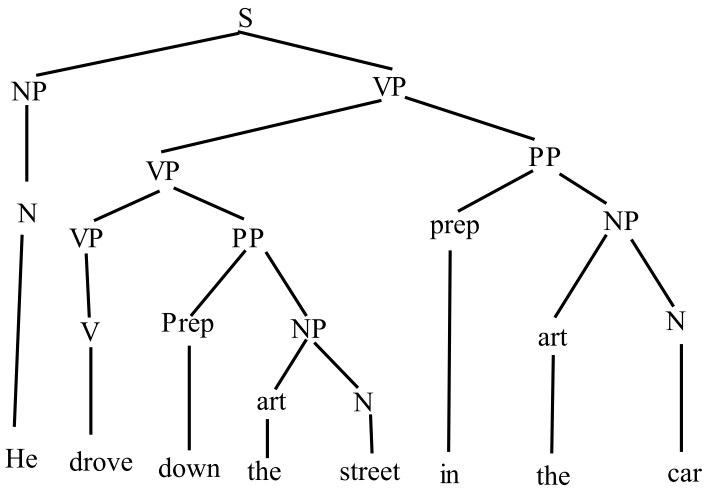
We note that, none of the steps (1) - (3), and (2)a., (2)b., (3)a., (3)b. produce/show producing the beginning of the sentence "Book...". When we try other rewrite rule (4), we find that, (4)a. and subsequently, (4)a.i. successfully generate the sentence "Book that flight". Here, we have not kept the sign "." as part of the sentence.

15.2 Ambiguous Grammars

An ambiguous grammar G has more than one parse-trees for some sentences. However, it is not necessary that every sentence in $L(G)$ has more than one parse-trees. Consider the sentence "He drove down the street in the Car". The two different parse-trees for this sentence are given in Fig. 15.2 and 15.3, respectively. The process of drawing a parse-trees is grouping of the words to realize the structure in the sentence. Fig. 15.2a and 15.2b demonstrate grouping of the words and parsing for one parse-tree, while Fig. 15.3a and 15.3b demonstrate for other parse-tree. Note that the words: *in, at, on, of, to, up, down* are *prepositions*; since the sentence has two prepositions (in and down), hence there are two separate PP subtrees in both the parse-trees.

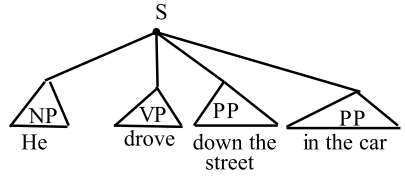


(a) Parsing-1: Grouping of the words for parsing

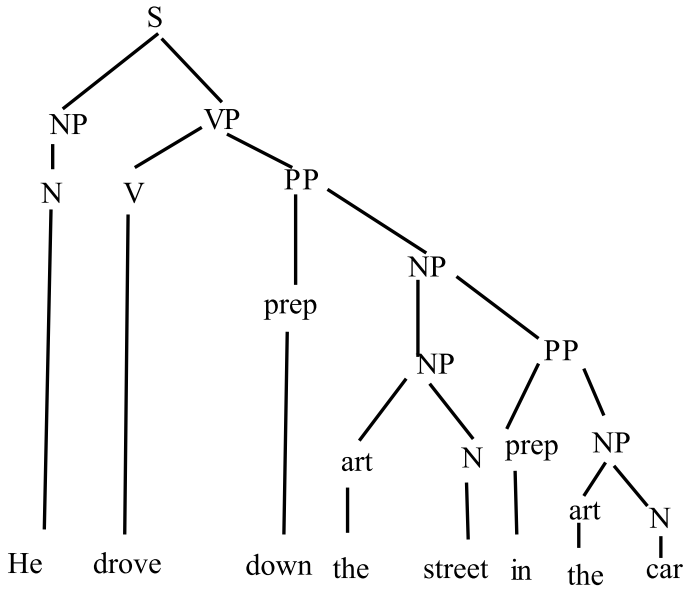


(b) Parsing-1 for : "He drove down the street in the car"

Figure 15.2: Parsing-1 for : "He drove down the street in the car"



(a) Parsing-2: Grouping of the words for parsing



(b) Parsing-2 for : "He drove down the street in the car"

Figure 15.3: Parsing-2 for : "He drove down the street in the car"

Deciding about which tree is correct, can be done by mechanically computing the probability. The correct parse-tree is one that has got higher over all probability.

15.3 Probabilistic Parsing

A probabilistic parser would produce a parse-tree τ for the sentence $w_{1,n}$ as,

$$argmax_{\tau} P(w_{1,n}). \tag{15.1}$$

In Context-free grammar (CFG) this is equivalent to maximizing the product of probabilities of the rules used in the parse. If $\mathcal{P}(\tau)$ are the rules used in the tree τ, then we want to find τ which maximizes,

$$\prod_{r \in \mathcal{P}(\tau)} P(r). \tag{15.2}$$

Here $w_{1,n}$ are the actual words of the sentence.

During the discussion above, we drawn a sharp line between correct and an incorrect parses, which was based on whether a terminal node either matched or did not match the next word in the sentence. According to this parsing, a phrase is either acceptable or not. There are situations under which we can relax these requirements, e.g., when we cannot afford to try alternatives exhaustively. The examples are: analysis of connected speech, text segmentation and identification of words can never be done with complete certainty. At the best, one can say that certain sound is more probable than the other. Consequently, one may associate a fractional number with each terminal node, indicating the probability or quality of nodes below that, in respect of forming grammatically correct sentence.

In the probabilistic parsing, the non-terminal nodes will be assigned some value, which are sophisticated enough to realize that syntactic and semantic restrictions are taken care of. Hence, a parser that aims to deliver the best analysis even if every analysis violates some constraint, must associate a measure of being grammatical (i.e., acceptable) with the analysis of portions of the sentence. The sentence ultimately associates a measure with the analyses of this complete sentence. As a matter of rule, it is possible to generate analysis of every sentence with a non-zero acceptability or matching probability, and then select the one having best analysis.

One simple parser of this type is *best-first* parser, which is based on modified form of standard *top-down serial* parsing algorithm for context-free grammars. The standard algorithm tries to generates one possible parse-tree until it gets stuck. That is, until the generation of a terminal node which does not match the next word in the sentence). In case of stuck, it "backs up" to try another alternative. A *best-procedure*, is like a best-first search that tries all alternatives in parallel. A measure in numerical value is associated with each alternative path that indicates the likelihood, that this analysis matches the sentence processed so far, and it can be extended to the complete sentence analysis. At each progressive step, the path with the highest likelihood is extended. In the process, if the "measure" of current path falls below that of some other path, the parser shifts its attention to that of other path.

A CFG (context-free grammar) consists of,

terminal words $w^1, w^2, ..., w^V$,

non-terminals $N^1, N^2, ..., N^n$,

start symbol N^1, and

production rules $N^i \rightarrow \alpha^j$

where V is length of the sentence w; n is number of non-terminals; and α^j is sequence of terminals and non-terminals.

Given the above, we can define a generative PCFG (probabilistic context-free grammar), where the first three lines are same.

terminal words $w^1, w^2, ..., w^V$,

non-terminals $N^1, N^2, ..., N^n$,

start symbol N^1, and

production rules $N^i \rightarrow \alpha^j$

where α^j is sequence of terminals and non-terminals, and the "rule's probabilities" are:

$$\forall_i \sum_j P(N^i \rightarrow \alpha^j) = 1, \tag{15.3}$$

which shows that for each set of productions having same left side variable (N^i), the sum of probabilities is unity. For example, in the rule $VP \rightarrow \alpha^1 \mid \alpha^2 \mid ... \mid \alpha^n$, the probability of this rule as a whole is unity.

We consider that a sentence is represented as sequence of words $w_1 w_2 ... w_m$, and $w_{ab} = w_a ... w_b$ is a subsequence. Let us assume that, non-terminal N^i dominates sub-sequence $w_a ... w_b$ is represented by N^i_{ab}, .i.e., N_i is root of subtree having children sequence as $w_a ... w_b$. Let $N^i \Rightarrow^* \alpha$. We represent the probability of sentence w_{1n} as,

$$P(w_{1n}) = argmax_t \prod_t P(w_{1n,t}) \tag{15.4}$$

where t is parse tree of sentence w_{1n}. Note that, there can be more than one parse-tree of a sentence when the sentence is ambiguous. The probability rule (15.4) implies that for each sentence, generate all the parse-trees t, do computation of probability for each, and then select the one having highest probability (maximum likely hood that a given parse tree is correct).

Example: Construct a correct parse-tree for the sentence "I saw an astronomer with telescope", for the following PCFG:

Rule	Probability	Rule	Probability
$S \rightarrow NP \;\; VP$	1.0	$NP \rightarrow NP \;\; PP$	0.20
$PP \rightarrow Prep \;\; NP$	1.0	$NP \rightarrow N$	0.45
$VP \rightarrow V \;\; NP$	0.7	$NP \rightarrow Art \;\; N$	0.35
$VP \rightarrow VP \;\; PP$	0.3	$N \rightarrow telescope$	0.25
$Prep \rightarrow with$	1.0	$N \rightarrow astronomer$	0.15
$Art \rightarrow an$	1.0	$N \rightarrow I$	0.60
$V \rightarrow saw$	1.0		

The sentence above can be generated using two different parse-trees: t_1 (see Fig. 15.4) and t_2 (see Fig. 15.5).

Various terminals and non-terminal symbols we shall use are:

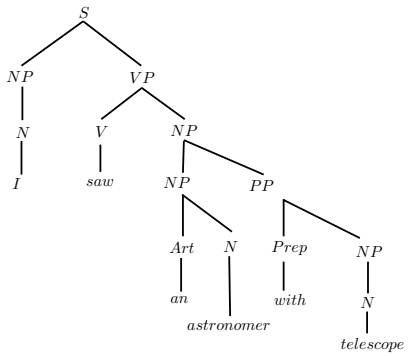


Figure 15.4: Probabilistic parse-tree t_1 for the sentence "I saw an astronomer with telescope"

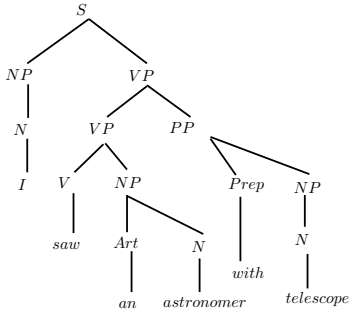


Figure 15.5: Probabilistic parse-tree t_2 for the sentence "I saw an astronomer with telescope"

- Terminals: I, saw, an, astronomer, with, telescope.
- Non-terminals: *S, NP, VP, PP, V, Prep, Art, N*
- Start Symbol: *S*

Probability for parse tree t_1 is product of all the probabilities of rules used, which starting from top, and going down through all the levels is given by:

$$\begin{aligned} P(t_1) =& 1.0 \times 0.45 \times 0.7 \times 0.60 \times 1.0 \times 0.20 \times 0.35 \times 1.0 \times 1.0 \times 0.45 \times 0.25 \\ =& 0.001488375 \end{aligned}$$

Similarly, the probability for parse tree t_2 is given by:

$$\begin{aligned} P(t_2) =& 1.0 \times 0.45 \times 0.3 \times 0.60 \times 1.0 \times 0.35 \times 1.0 \times 0.45 \times 1.0 \times 0.15 \times 0.25 \\ =& 0.00047840625 \end{aligned}$$

Naturally, the parse tree t_1 is correct, as probability of its construction is higher. The tree t_1 indicates that the observer (I) is seeing an astronomer carrying telescope, while t_2 has semantics which indicates that observer (I) is seeing an astronomer with the help of telescope, which obviously is incorrect[1].

[1] One would see the stars through telescope and not the astronomer through a telescope !

The above computations demonstrate that t_1 and t_2 trees both have generated the same sentence. But, which one is semantically true is yet to be decided. However, the first parse tree has higher probability, hence semantics associated with this tree should be true. The semantics says that "telescope is carried by astronomer" is true, which is as per normal conventions. However, this is to be proved logically using tree t_1, using the semantics associated with the syntax. We shall discuss in the semantics in the next section.□

The accuracy of probability of each tree depends on the accuracy of the probabilities of rules used. The probabilities associated with the rules are made available from the the statistics of semantics from large corpus of natural language text.

The benefit of using probability in parsing is that it that it gives a *probabilistic language model*. In addition, the probabilistic parsing provides a solution to ambiguity in language and grammar, as is conclusive from above example. But, this is not necessarily complete, but in partial.

Dependency relation in Parse-trees

In the above example, of astronomer and telescope, the ambiguity is very clear to us, as "weather we are seeing the astronomer through telescope", or "the astronomer is found carrying the telescope"? We try to establish logically, which of these two interpreted sentence correspond to t_1 and which to t_2. The rule VP PP, is like "cut with (knife)", "eat with (fork)", "write with (pen)", such that the verb has dependency with the noun. Hence, the tree t_2 falls in this class. But, t_2 is semantically wrong because one would not use telescope to see an astronomer!

The first tree (t_1), the node $V\ NP$ indicates dependency of V on NP, i.e., "saw an astronomer". The subtree of "with telescope" is distant, and has far less dependency on "saw". Thus, the semantics here is like, "saw man (with pen)", "saw farmer (with turban)", "carried onion (with potatoes)", "did Bsc (with BA)", and so on.

Thus, logically, and through dependency rules, we say, that tree t_1 presents the correct sense of the sentence.