

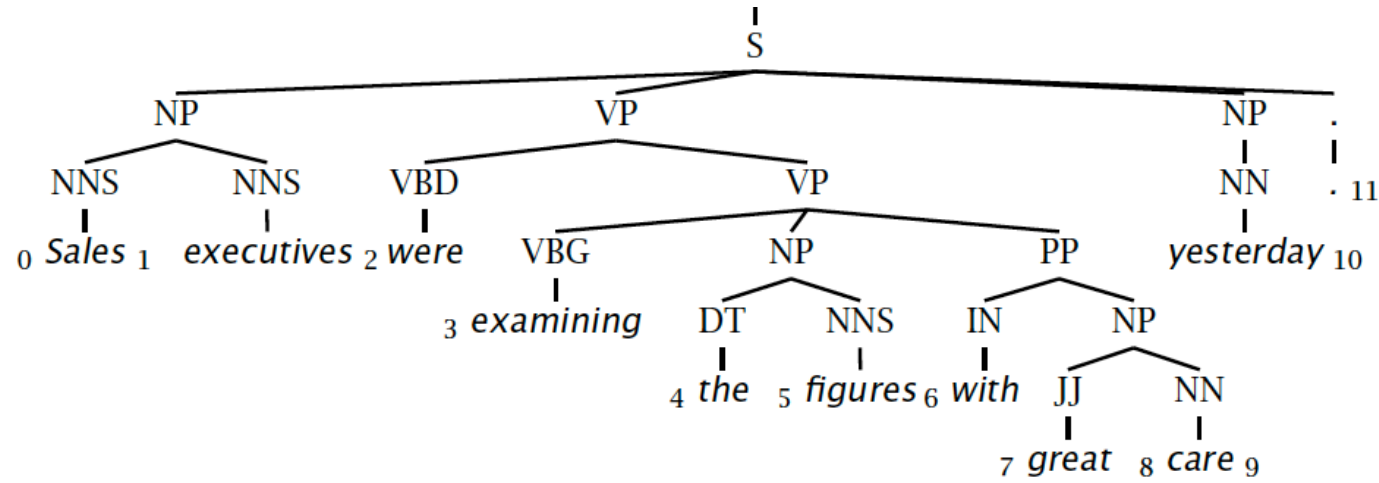


Constituency Parser Evaluation

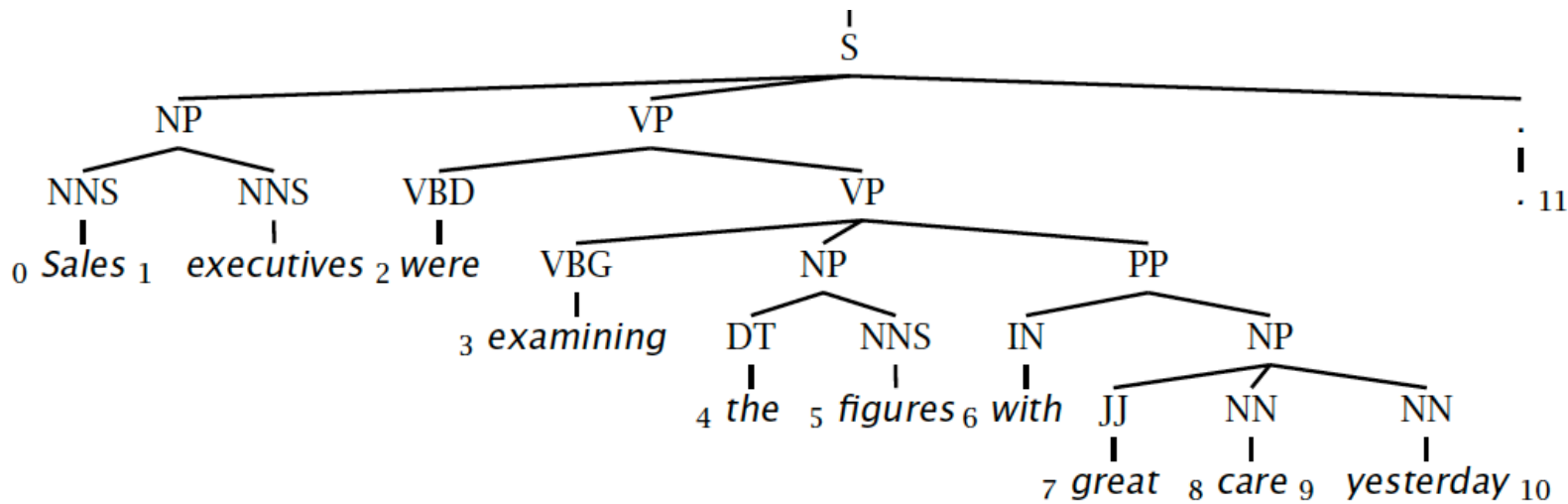


Evaluating constituency parsing

Gold standard brackets: S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), NP-(4:6), PP-(6:9), NP-(7,9), NP-(9:10)



Candidate brackets: S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), NP-(4:6), PP-(6:10), NP-(7,10)





Evaluating constituency parsing

Gold standard brackets:

S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), **NP-(4:6)**, PP-(6-9), NP-(7,9), NP-(9:10)

Candidate brackets:

S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), **NP-(4:6)**, PP-(6-10), NP-(7,10)

Labeled Precision	$3/7 = 42.9\%$
Labeled Recall	$3/8 = 37.5\%$
LP/LR F1	40.0%
Tagging Accuracy	$11/11 = 100.0\%$



How good are PCFGs?

- Penn WSJ parsing accuracy: about 73% LP/LR F1
- Robust
 - Usually admit everything, but with low probability
- Partial solution for grammar ambiguity
 - A PCFG gives some idea of the plausibility of a parse
 - But not so good because the independence assumptions are too strong
- Give a probabilistic language model
 - But in the simple case it performs worse than a trigram model
- The problem seems to be that PCFGs lack the lexicalization of a trigram model



Constituency Parser Evaluation