NLP - 14 (05/11/24)

Syntatic Parsing:

↳ Morphology is based on independent words, while syntax deals with combinations of words.

↳ Morphology is often irregular (ablaut/umlaut)

↳ Some sentences are syntactically/grammar wise correct but have no semantic meaning.

⇒ constituents.

↳ group of words that give meaning when they go together.

↳ one phrase can have multiple phrases.

↳ noun phrase/preposition phrase/clause

CFG

↳ terminal → vocab

↳ Det → Determiner

↳ Parsing → using parse tree

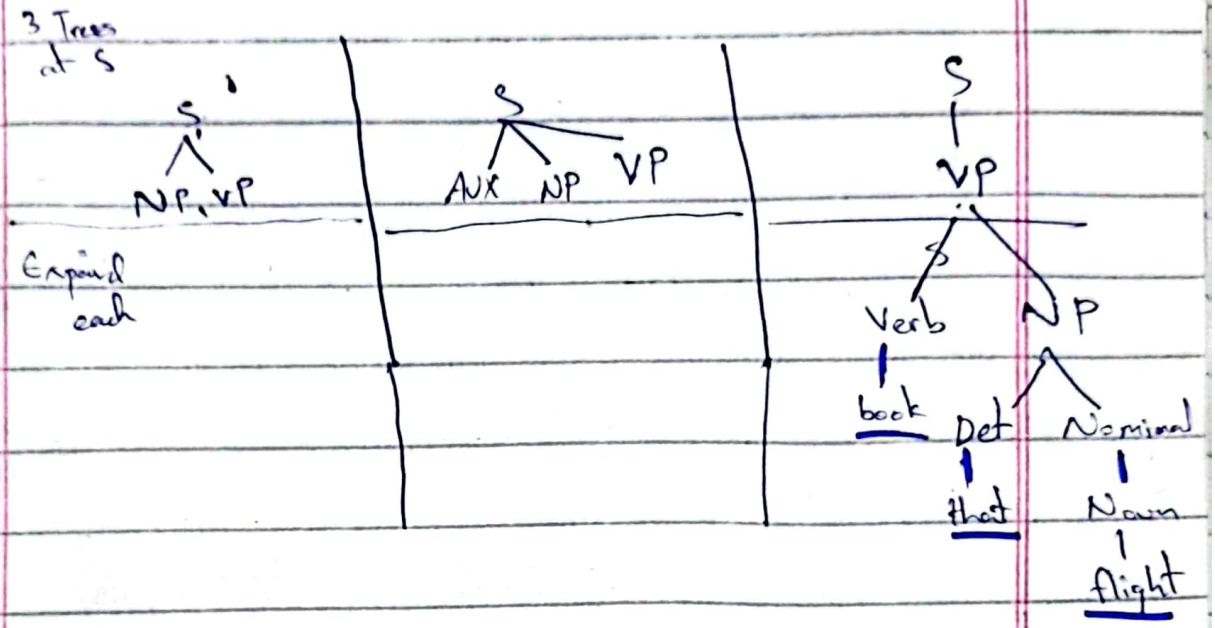
bottom up ↵ ↳ top down.

↳ CFG requires lots of knowledge & rules.

Top Parser:

Example: (Top Down Parsing).

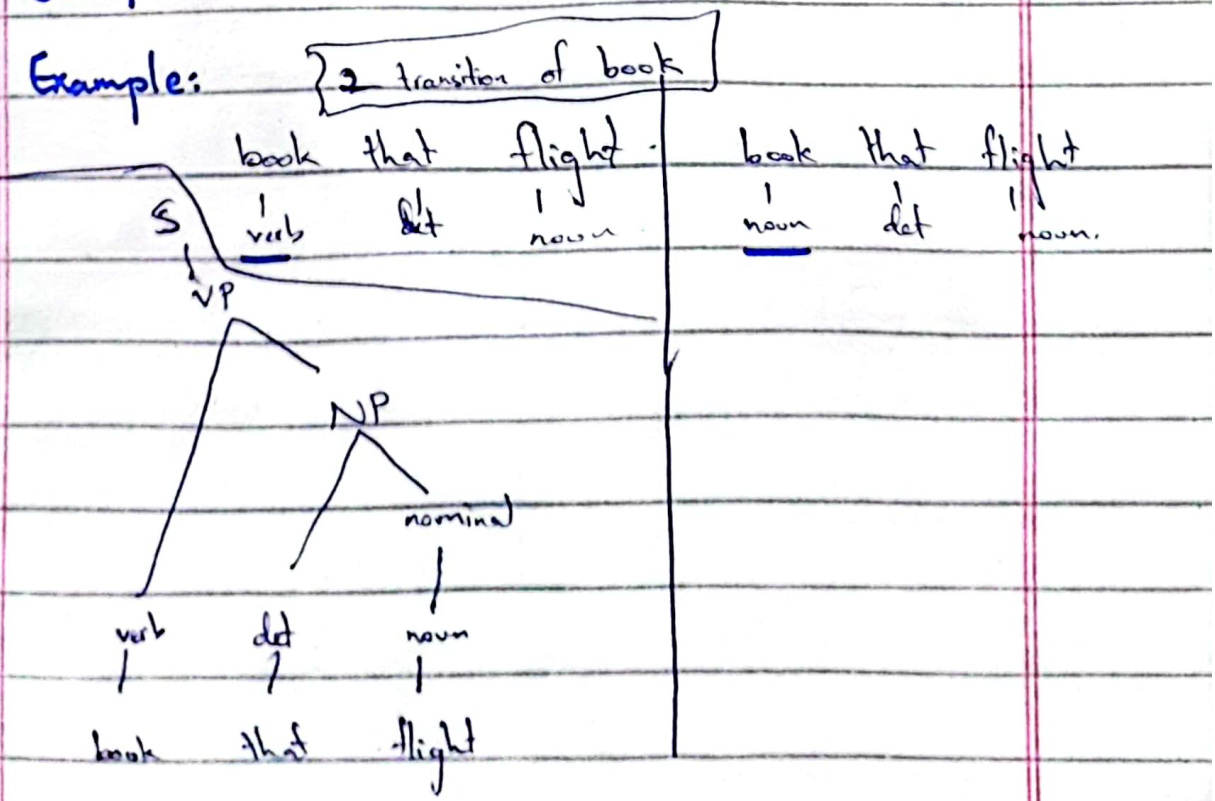
book that flight



If one sentence has more than one possible parse trees ⇒ ambigious sentence.

Bottom Up Parsing:

Example:



In bottom up, start with terminal & check production rules.

↳ check pairs else single words.

↳ shift reduce parsing. (bottom up parsing)

↳ for training parser, manually annotated corpus is required

↳ Sentactic ambiguity leads toward semantic ambiguity.

↳ Pantreebank.

NLP-15 (08/11/24)

↳ If multiple parse trees, which one to use?

↳ Use probability

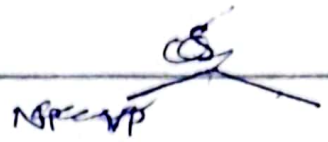
↳ require probability of each production rule.

↳ probabilistic CFG.

↳ prob sum of each non-terminal is 1.

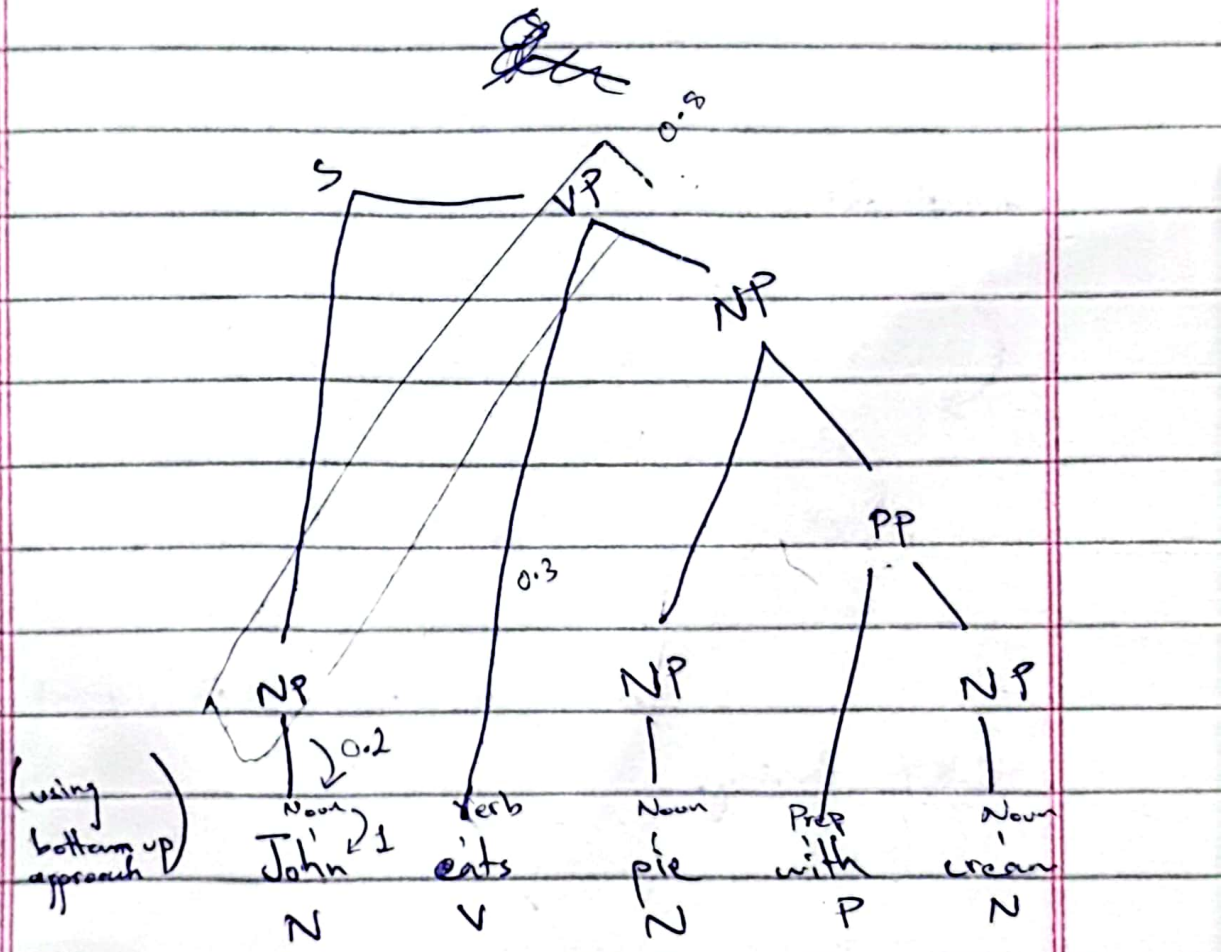
→ prob(1) = $\frac{\text{count (S noun S)}}{\text{count (starting with S)}}$

→ prob(2) = $\frac{\text{count (only Noun as NP)}}{\text{count (NP)}}$



Example:

⇒ John/N eats/V pie/N with/P cream/N.



↳ computing prob of parse tree.

S → NP VP 0.8.

NP → Noun 0.2.

-

prob = (0.8 * 0.2 * 1) * (0.3 * 1.0) *

Workflow → flow diagram + written flow.

desired output ← for evaluation 15-20 docs | fallback mechanism. | llama / OpenAI | beautiful soup. winhd

↳ CFG must be converted to CNF

NO more than 2 siblings on right side.

Example:

$X \rightarrow ABC \Rightarrow X \rightarrow AX_2.$

$X_2 \rightarrow BC.$

↳ new production rule added.

↳ new symbol added to vocab.

Example 2:

$VP \rightarrow VBD\ NP\ PP\ PP$

$X_2 \rightarrow NP\ PP.$

$X_3 \rightarrow X_2\ PP.$

$VP \rightarrow VBD\ X_3.$

NLP-16 (12/11/24)

CKY Parsing:

↳ bottom up parser.

↳ grammar should be in CNF.

↳ fill table with unaries.

↳ if prob, add also.

↳ Ambiguity is handled by probability.

↳ top cell has atleast one rule starting with S else not valid.

No → AB, C

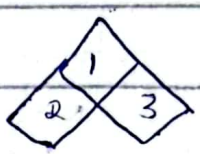
↳ reduce the rules

V → people 0.1
VP → V 0.1

reduced as ↳

VP → V 0.1 * 0.1

NP → N 0.5 * 0.7



↳ cell 1 will have combinations of 2 & 3

↳ discard invalid combinations,

S → NP VP (0.9 * 0.35 * 0.06)

In computer we use matrix

Example:

	fish	People	Fish	Tank
N → fish V → fish				
	0.5 N → people 0.1 V → people			
			N → fish V → fish	
				N → tanks V → tanks

fish people fish tanks

(2)

(1)

	fish	people	fish	tanks
fish	N → fish 0.2 V → fish 0.6 NP → N 0.7 VP → V 0.1 0.6 S → VP 0.1*0.6*0.1	VP → V NP	(1)	(3)
people		N → people 0.5 V → people 0.1 NP → N VP → V S → VP		(2)
fish			N → fish 0.2 N → fish 0.6	
tanks				N → tanks 0.2 N → tanks 0.3

(4)

1) [fish people] [fish]

2) [fish] [people fish]

(2)

1) [people] [fish tanks]

2) ——— ———

(3)

1) fish people fish tanks

2) ——— ——— ——— X

3) ——— ——— ——— X

4) ——— ——— ——— ✓

5) ——— ——— ✓

6) ——— ——— ——— X

Due to restriction of CNF (binaries only)

NLP-17 (15/11/24)

Embeddings:

↳ numbers are fast for processing. ('by arranging in order then assign num')

↳ One hot encode ⇒ does not give info about similar/opposite semantic words.

↳ Word Embeddings provide similarity & has low dimensions.

↳ min/ distance b/w similar words.

↳ corpus required (words in d/f contexts)

↳ low dimension → data loss (low acc)

↳ high dimension → no data loss (high acc).

↳ BERT ⇒ attention mechanism.

↳ train embedding from scratch/ or train/finetune for specific context.

↳ m ⇒ multilingual. (mBERT)

↳ ~~red~~ predict next word depending upon basic information.

↳ center word prediction

↳ prev info + next info used.

↳ possible words from corpus.

↳ word with lowest vector distance.

↳ <s> <s> ____

(for starting)

↳ In CBOW, embedding is bi-product of model.

NLP-18 (19/11/24)

Words to Vector.

CBOW → basic purpose ⇒ center word (pred)

↳ biproduct ⇒ embeddings.

noun/4 → value can't go beyond 1.

⇒ input vector size ⇒ vocab size.

* (V*1) (col) * (1*V) (row)

Embedding Dimension size ← N * V ⇒ 100 * 20.

V: Vocab size.

⇒ Batch

↳ usually 2^{power}, for faster computation.

↳ does not greatly impact on accuracy.

ReLU

↳ max(0, val)

Softmax.

↳ convert nums to probabilities.

(0-1) ⇒ sum = 1.

↳ $\frac{exp(value)}{sum\ of\ values}$.

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{V} e^{z_j}}$$

↳ If pred vs actual are not same weights & bais are updated ⇒ back prop

↳ KNN ⇒ not fixed activation function (RELU) cross-entropy.

↳ CBOW ⇒ shallow network (2W + 2 Bias).

NLP - 19 (22/11/24)

↳ stop training when loss not decreasing.
  ↳ manually / flags in code.

↳ CBOW → prediction (main)
  embeddings (biproduct).

↳ extract embeddings from hidden layer.
  w_1 → col matrix , w_2 → row matrix
  ↳ transpose to add.

  N * V → dimensions.

BERT Model:

↳ attention layer.

↳ mostly for embedding only (bi-prod).

↳ for classification, need to add layer to end.

→ for training from scratch.

↳ large data + time req.

→ Extrinsic Evaluation:

↳ after using in external task

→ Intrinsic Evaluation:

↳ without using in external task.