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

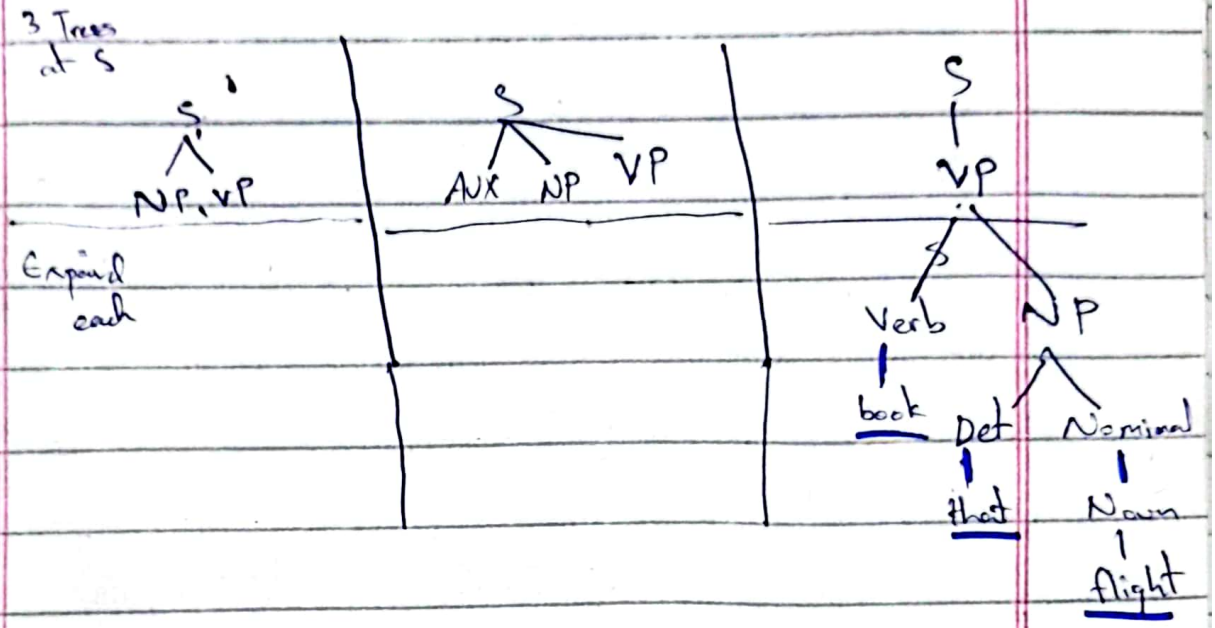
bottom up ↵ ↳ top down.

↳ CFG requires lots of knowledge & rules.

## Top Parser:

Example: (Top Down Parsing).

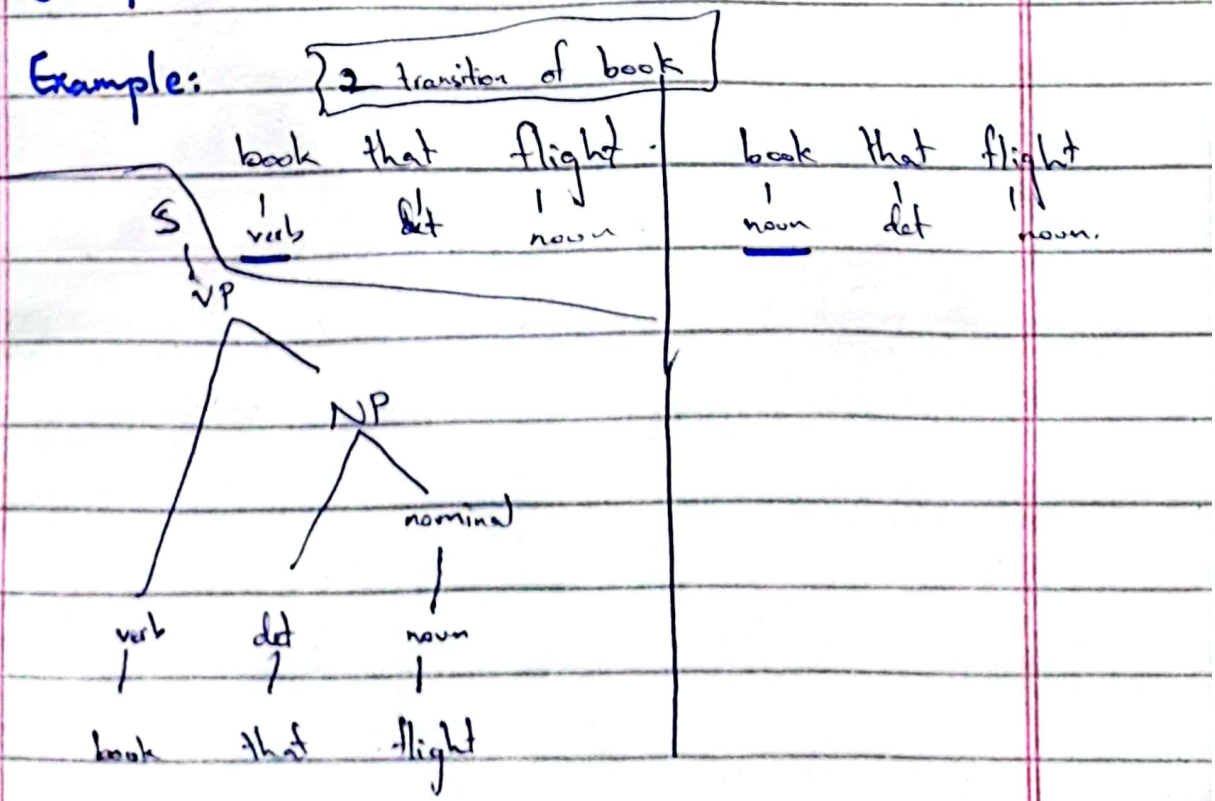
book that flight




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

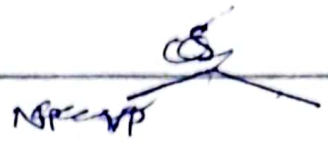
↳ require probability of each production rule.

↳ probabilistic CFG.

↳ prob sum of each non-terminal is 1.

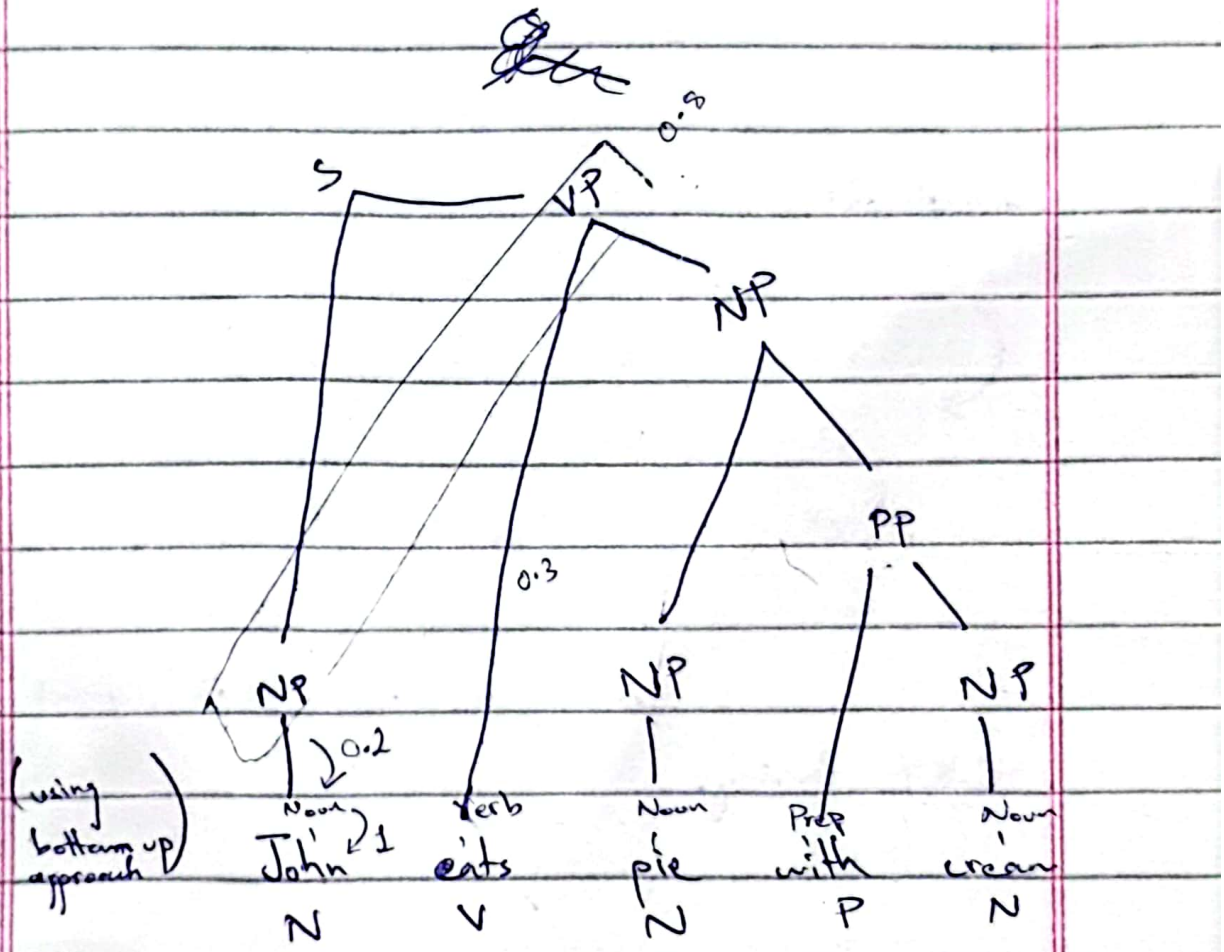
→ prob(1) = $\frac{\text{count (S noun S)}}{\text{count (starting with S)}}$

→ 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

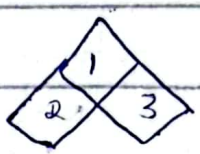
↳ reduce the rules

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<br>V → fish | | | | |
| | 0.5 N → people<br>0.1 V → people | | | |
| | | | N → fish<br>V → fish | |
| | | | | N → tanks<br>V → tanks |

fish people fish tanks

(2)

(1)

| | fish | people | fish | tanks |
|---|---|---|---|---|
| fish | N → fish 0.2<br>V → fish 0.6<br>NP → N 0.7<br>VP → V 0.1 0.6<br>S → VP 0.1*0.6*0.1 | VP → V NP | (1) | (3) |
| people | | N → people 0.5<br>V → people 0.1<br>NP → N<br>VP → V<br>S → VP | | (2) |
| fish | | | N → fish 0.2<br>N → fish 0.6 | |
| tanks | | | | N → tanks 0.2<br>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.

## NLP - 19 (29/11/24)

### Question Answering:

↳ Oldest nlp task.

↳ factoid / narrative.

↳ IR based (doc/content unstructured).
Knowledge based (any type of db).
Hybrid (both IR + Knowledge based)

↳ no context is maintained, all questions are independent.

↳ context in chatbot.

NLP-20 (03/12/24)

## Indexing:

↳ for fast searching (SOLAR, USEEN)

↳ which embeddings for vector conversion.

↳ for customised chatbot.

↳ if info is not factoid, answer retrieval becomes complex.

↳ MRR ⇒ for info retrieval.

↳ knowledge base

## Langchain:

↳ focus on modularity.

↳ Krisma, Redis for db.

↳ data loaded in vector store.

↳ Doc loaders ⇒ structured + unstructured data.

Read from file.

split ⇒ page, further split.

store in db.

Yt GPT

↳ wisper for vid to audio

⇒ Doc Split

↳ overlap required while chunk split.

# NLP-21 (6/12/24)

## Chatbot

sentences/word embeddings.

embed query ⇒ loop on splits.

Q: How to use vector store?

2 divisions of chatbot.

1) creation 2) usage.

⇒ May have repitition in top k.

1) Exact match easily detected.

2) Paraphrase check also.

↳ top relevant & distinct.

max_marginal_relevance instead of similarity_search.

↳ meta_data_filter for specific pdf/doc.

NLP-22 (10/12/24)

## Neural Machine Translation:

↳ Text/Speech

↳ speech → text → speech

speech → speech (Under Research)

↳ seq learning problem, but input & output are not alligned (length can vary)

↳ basic neural network ⇒ latest.

↳ NMT / OpenMT.

↳ Map variable length seq on fixed length memory.

LSTM & GRU.

Encoder → learns embedding of source key.

Decoder → learns embedding of target key.

⇒ CBOW / FASTX / GLOB.

⇒ No of words ⇒ no of LSTM units.

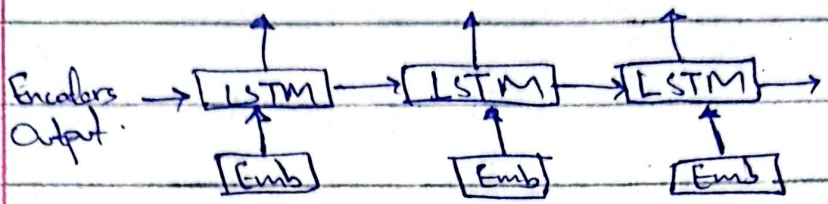
⇒ LSTM recieves current word + previous words (context) max 2/3 prev words.

⇒ Each LSTM decides

↳ reset / modify / retain.

⇒ Encoder has embeddings & 1 layer of LSTMs

⇒ Output of LSTM layer is given to decoder




**Info bottleneck:**

↳ only fixed amount of info to decoder.

↳ context from each LSTM or final LSTM only.

**Attention:**

↳ context to each meaning.

↳ attention layer needs to be trained.

↳ when weighted sum is added, feedback is sent to encoder.

# NLP-23 (13/12/24)

## CBOW Model:

(Input) context vector → output vector.
↳ result (embeddings)

⇒ Corpus for machine translation.
parallel corpus ⇒ source to target.

⇒ Query ⇒ target text (translated).
Sentence translation rather than word.
translation (for learning alignment).

⇒ Eng to urdu don't form diagonal in attention

⇒ no preprocessing for emojis, digits (auto learned).

⇒ 2-3 people translate 1 sentence.
more than 1 person verify.

⇒ Why padding?
for fix dimension of input vector.
Each token has unique (word to index).
↳ embedding for encoder.
Softmax output ⇒ as weights.

NLP- 24 (17/12/24)

## Attention Layer:

↳ Determines

↳ softmax behaves differently.

↳ Encoder ⇒ learns representation of source lang

↳ Decoder ⇒ learns representation of target lang.

↳ teacher forcing ⇒ Don't compute loss in initial epochs (assume output as correct), after that start calculating the loss.
(on pre attention decoder)

↳ Attention layer requires input from encoder + decoder & new weights are sent to decoder

shift Right ⇒ Mark start/end of sentence + padding if required.

logSoftmax ⇒ log probabilities.

RNN has vanishing problem ⇒ so LSTM is used

## BLEU Score:

↳ for evaluating machine translation/ summarization etc.

(Bad) 0 — 1 (Good)

Bilingual Evaluation Understudy.

HT → Ground Truth / Gold Reference / Standard.

MT → Candidate

↳ based on precision.

↳ not subjective but qualitative measure.

↳ If word of candidate sentence is found in any reference, add 1 else 0. Then divide total by length count (candidate).

↳ vanilla BLEU has prob, not good score.

↳ Modified BLEU, delete word from reference if matched with candidate.

Issues:

↳ no semantic meaning.

↳ no sentence structure

⇒ (opposite case ⇒ score is bad but good translation ⇒ reference grammar is diff from training data).

ROUGE: (Recall)

Recall Oriented Understudy for Gisted Evaluation.

⇒ on unigram, bigram level etc.

max of 2 or 3 references.

↳ separate count for each reference

⇒ Opposite (ref is matched with candidate).

**F_Measure:**

$$\frac{2 * BLEU * ROUGE}{BLEU + ROUGE}$$

$$\Rightarrow \frac{2 * 0.5 * 0.4}{0.5 + 0.4} = 0.44$$

**Next Topic**

dictionaries don't have similarities

e.g car vs bicycle

⇒ word sense vs synonyms (tall vs long).

⇒ word net ⇒ contains word sinset

↳ **Homophones:**

only speaking is same.

↳ **Homographs:**

same spells only.

↳ **Sinset** ⇒ used interchangeably.

⇒ **Zeugma Test**

for identifying word sense.

↳ use word in multiple sentences.

(Also merge them and check)

↳ If cannot be concated ⇒ multiple sense.

⇒ wordnet ⇒ (web + api + desktop version).

↳ for senses of words.

↳ not a corpus.

↳ It is a lexical resource.

↳ Hierarchy helps in making anologies.

MeSH (Medical Subject Headings).

→ Word Similarity : Thesarus Method.

How to calculate for wdv?

→ 2 types of Algo.

1) Use lexical resources.

2) use ML & corpus

NLP - 25 (19/12/24)

1) Thesarus based (dict/wornet)

↳ Path based similarity.

edges + 1 → pathlen.

simpath = 1/pathlen.

(sometime not good in similarity as dis-similar things can have good score → can't threshold)

↳ Info content similarity

↳ every word has prob (instead of 1)

↳ prob from corpus

↳ descendent ⇒ sub-childs also

$$\frac{\text{sum of probs}}{N}$$

∵ N = no of words in corpus

↳ log prob ⇒ to avoid overflow issue

$$IC(c) = -\log P(c)$$

↳ lowest common subsumer (LCS)

1) convert all prob to log.

⇒ only LCS or all words?

↳ use all common words

↳ high similarity value of at words on low level of tree.

↳ Lin Similarity:

only LCS used.

$$(A,B) = \frac{2\log P(LCS(c_1,c_2))}{\log P(c_1) + \log P(c_2)}$$

⇒ wordnet required

## The (extended) Lesk Algo

⇒ A thesaurus-based measure that looks at glosses.

↳ uses dict.

↳ get desc of both words from dic & find similarity using dict. description.

↳ 1st match phrase then words

$1 + 2^2 \neq 5$

word phrase

## Embeddings or Cosine similarity.

(previously it was applied on count now applied on embeddings)

## bLSTM

↳ bi-direction (future info also used).