Q 3.7

Given <s> I am Sam </s>
<s> Sam I am </s>
<s> I am Sam </s>
<s> I do not like green eggs and Sam </s>.

$N = 25$ (considering <s> and </s>)

$V = 11$ $\lambda_1 = 1/2$ $\lambda_2 = 1/2$.

using linear interpolation

$$\hat{P}(Sam/am) = \lambda_1 P(Sam/am) + \lambda_2 (Sam)$$

$$= \frac{1}{2} \frac{C(am\ Sam)}{C(am)} + \frac{1}{2} \frac{C(Sam)}{N}.$$

$$= \frac{1}{2} \times \frac{2}{3} + \frac{1}{2} \times \frac{4}{25}$$

$$= 0.33 + 0.08.$$

$$\boxed{\hat{P}(Sam/am) = 0.41}$$

* CKY Algorithm

\# Conversion to CNF.

Q. $A \rightarrow BCD$
$K \rightarrow e$

$\Rightarrow$ $A \rightarrow BC'$
$C' \rightarrow C''D$
$C'' \rightarrow e$
$K \rightarrow e$

eg. Book the flight through houston

0 Book 1 the 2 flight 3 through 4 Houston 5.

	1	2	3	4	5
0	S, VP, Verb nominal, noun 01	- 02	VP, S, X2 03	- 04	S, X2, VP 05
1	11	Det 12	NP 13	- 14	NP 15
2	21	22	nominal noun 23	- 24	nominal 25
3	31	32	33	prep 34	PP 35
4	41	42		44	NP, proper noun 45
					55

for [02] we see [01] [12].

i.e $x_{02} \longrightarrow x_{01}\ x_{12}$

check for a production $X \longrightarrow$ (anyone entry of (0,1) (Det))

Now for x_{13}

$x_{13} = x_{12}\ x_{23}$.

NP $\longrightarrow$ Det. nominal

for x_{03}.

$x_{03} \longrightarrow x_{01}\ x_{13}$

$x_{02}\ x_{23}$. ϕ.

for x_{14}

$x_{14} \longrightarrow x_{12}\ x_{24}$ ϕ

$x_{13}\ x_{34}$

for x_{04}

$x_{04} \longrightarrow x_{01}\ x_{14}$ ϕ

$x_{02}\ x_{24}$ ϕ

$x_{03}\ x_{34}$

for x_{05}

$x_{05} \longrightarrow x_{01}\ x_{15}$ | $x_{04}\ x_{45}$

$x_{02}\ x_{25}$

$x_{03}\ x_{35}$

Q. "A pilot likes flying planes" for CNF CFG :-

S → NP VP	NN → pilot
VP → VBG NNS	VBZ → likes
VP → VBZ NP	VBG → flying
NP → DT NN	JJ → flying
NP → JJ NNS	NNS → planes.
DT → a	

⇒ problem with CYK.

→ So CYK algo is not a parser but a recognizer (We can make the parse tree with some modifications)

→ Problem of ambiguity is still ~~unresolved~~ unresolved.

For solving the problem of ambiguity, we use :-

Probabilistic CFG.

Syntactic ambiguity is of 2 types:-

→ Attachment

arises with the ~~a~~ attachment of multiple productions

eg the boy saw the girl with Telescope

→ Coordination

eg old men and women.

Probabilistic CFG is a 5 ~~typ~~ tuple (T, N, S, R, P)

where TNSR have same meaning

~~P~~ P(R) ⇒ probability of each rule such that

$\forall x \in N \quad \sum_{x \to x \in R} P(x \to \gamma) = 1$

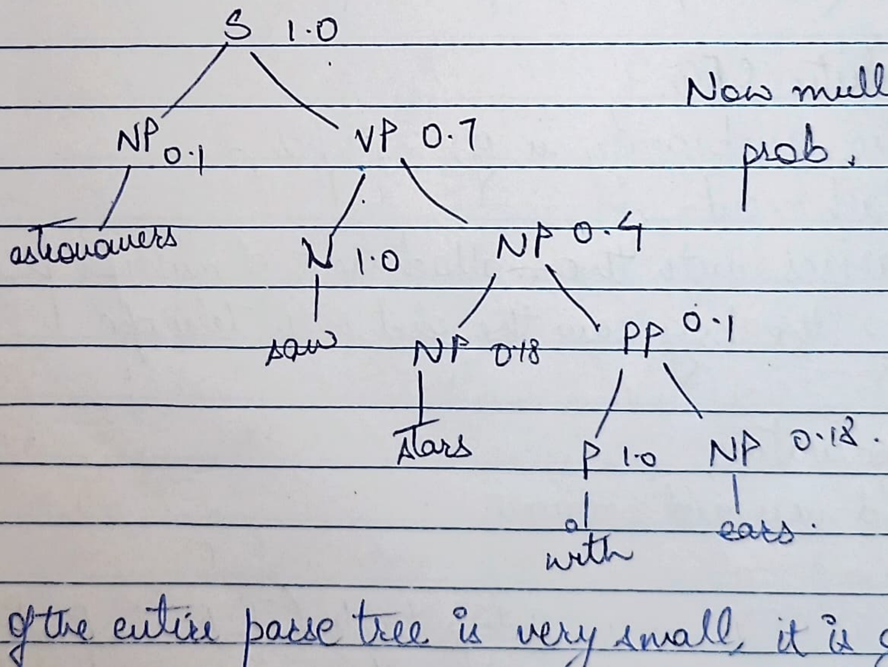
P(t) ⇒ probability of tree. It is the product of probabilities of all the rules

Ch. 12 & 13   Some grammar rules of english → book.
Treebank

eg.

S → NP VP	1.0
VP → V NP	0.7
VP → VP PP	0.3
PP → P NP	1.0
P → with	1.0
V → saw	1.0
NP → NP PP	~~1.0~~ 0.4
NP → astronomers	0.1
NP → ears	0.18
NP → saw	0.4
NP → stars	0.18
NP → telescope	0.1

→ Astronomers saw stars with ears.



Now multiply all prob.

⇒ If probability of the entire parse tree is very small, it is grammatically incorrect.

→ Shorter trees will have more probability as compared to longer trees.

Intermediate to surface ~~lexical~~ orthographic rule:
If plural 's' follows a morpheme ending in 'x', 'z' or 's', insert e.

Unit 3.

→ Read 12.3 from book.

Treebank.

Sufficiently robust grammars can be used to assign a parse tree to any sentence. This means that it is possible to build a corpus where every sentence in the collection is paired with a corresponding parse tree. Such a syntactically annotated corpus is called a treebank.

Ambiguity

Structural Ambiguity occurs when the grammar can assign more than one parse to a sentence.
For eg: the sentence "I shot an elephant in my pajamas"
the phrase 'in my pajamas' can be a part of Noun phrase (NP) headed by elephant or a part of verb phrase headed by 'shot'.

Types

1) Attachment Ambiguity
A sentence has AA if a particular constituent can be attached to the parse tree at more than one place. see eg.

2) Coordination Ambiguity
In CA, phrases can be conjoined by by a conjunction like 'and'.
eg the phrase old men and women can be bracketed as [old [men and women]] refering to old men and old women.
or as [[old man] and [women]]

See video for CYK and PCYK. & learn algo from Pg 264
(agar time ho to book meiṇ highlighted pe nazar marle)

Probabilistic CFGr.

See mam's notes + Read a proper def from net if you don't understand that. Only def impt.

⇒ Problems in PCFGrs as parsing models.

There are 2 main issues with PCFGr:-

1) Lack of sensitivity to lexical information (main).
2) Lack of sentivity to structural preferences.

→ Lack of sensitivity to lexical information

Because under the PCFGr model, probabilities of nodes are multi-plied we can say that the choice of each word in a string is conditionally independent of entire tree, once we have conditioned on the POS directly above the word.

This can cause all sorts of problems. For eg. prepositional phrase (PP) attachment ambiguity (have not impt).

So for a particular sentence, the parser will choose.

$VP \to VP\ PP$ over $NP \to NP\ PP$ if

$q(VP \to VP\ PP) > q(NP \to NP\ PP)$

and vice-a-versa

Now this decision is entirely independent of any lexical informa-tion (the words) in the i/p sequence while the lexical items involved can give strong evidence about whether to attach a noun or a verb.

→ Lack of sensitivity to structural preferences.

Highlighted but can't understand the difference b/w close attachment & far attachment

Lexicalization of treebank.

The key idea here is to identify for each context free rule of form

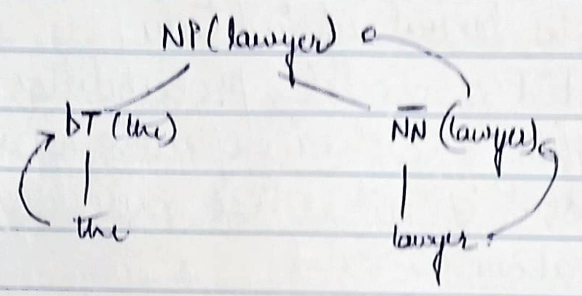
$$X \rightarrow Y_1 Y_2 \ldots Y_n$$

an index $h \in \{1, 2 \ldots n\}$ that specifies the head of the rule. The head of the context free rule intuitively corresponds to the center or the most important child of the rule.

So for $S \rightarrow NP\ VP$ the head would be VP.

Once the head of each context free rule has been identified, lexical information can be propagated bottom up through parse tree in the treebank.

So in the example:-



Part of speech such as DT or NN recieve the lexical item below them as head words. Non terminals higher in the tree recieve the lexical items from their head child.

Lexicalised PCFG.

In Lexicalised PCFG, :-

→ each rule will have an associated parameter for example the above rule would have

$$q(S(\text{examined}) \rightarrow NP(\text{lawyer})\ VP(\text{examined}))$$

→ each rule has a non terminal with head word on left side of rule

→ each rule has 2 children & one of the 2 children must have the same lexical item as left hand side

Notation

$$S(\text{examined}) \rightarrow_2 NP(\text{lawyer})\ VP(\text{examined})$$

$$PP(\text{in}) \rightarrow_1 PP(\text{in})\ PP(\text{in})$$

formally a Lexicalized PCFG (in CNF) is a 6 Tuple $G = (N, \Sigma, R, S, q, \gamma)$ where :-

→ N is a finite set of non terminals in the grammar
→ Σ is a finite set of lexical items in grammar
→ R is set of rules of the form :-

- $X(h) \rightarrow_1 Y_1(h)\ Y_2(m)$
- $X(h) \rightarrow_2 Y_1(m)\ Y_2(h)$ } $X, Y_1, Y_2 \in N,\ h, m \in \Sigma$
- $X(h) \rightarrow h$ where $X \in N,\ h \in \Sigma$.

→ S is the start symbol
→ For each rule $r \in R$ there is an associated parameter $q(r)$

s.t. $q(r) \geq 0$ & $\forall$ for any $X \in N,\ h \in \Sigma$,

$$\sum_{r \in R : LHR(r) = X(h)} q(r) = 1$$

(transition (of sorts) probability)

where $LHS(r)$ is LHS or any rule r

→ For each $X \in N,\ h \in \Sigma$, there is a parameter $\gamma(X, h)$. We have $\gamma(X, h) \geq 0$ & $\sum_{X \in N, h \in \Sigma} \gamma(X, h) = 1$ (for root)

Feature Structures

imagine we have a grammar for the sentence :-
" the gangster dies " as

S → NP VP
NP → Det N
VP → IV
Det → the
N → gangster
IV → dies

Now if we want to include the sentence "the gansters die" then we will have to add new productions to the grammar.

As we try to include more forms of verbs, more pronouns, more verbs, the grammar would explode.

So ~~th~~ what we can do is instead of ~~talking~~ considering the non terminals as atomic category symbols, we consider them as a set of properties. Certain rules can then be imposed to constraint on these properties. These constraints can force a certain property to have a certain value, 2 properties to have the same value no matter what that value is.

Using this we can write the grammar as:—

S → NP VP. number of NP = number of VP
NP → Det N
VP → IV
Det → the
N → gangster   number of N = singular
N → gangsters   number of N = plural
IV → dies   number of IV = singular
IV → die   " " " = plural.

Such sets of properties are commonly represented as feature structures.

Feature structures are often written as attribute value ~~pair~~ matrices.

For eg the attribute value matrix expressing that something is singular and 3rd person is:

$$\begin{bmatrix} \text{NUMBER} & \text{sg} \\ \text{PERSON} & 3 \end{bmatrix}$$

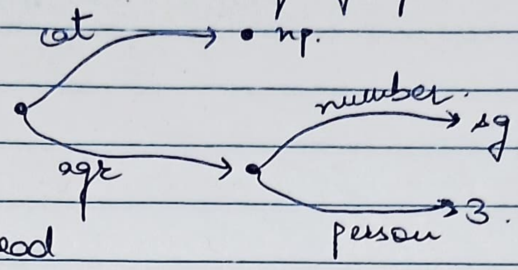
→ This is also valid:

$$\begin{bmatrix} \text{cat} & \text{np} \\ \text{agr} & \begin{bmatrix} \text{number} & \text{sg} \\ \text{person} & 3 \end{bmatrix} \end{bmatrix}$$

This makes it possible to group features of common type together

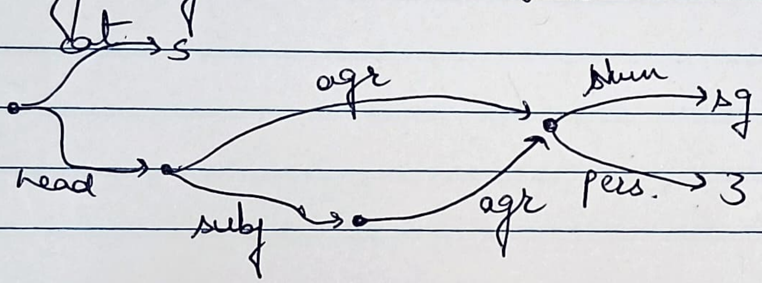
→ Feature structure ~~will~~ can be represented through graphs like:-

$$\begin{bmatrix} \text{cat} & \text{np} \\ \text{agr} & \begin{bmatrix} \text{number} & \text{sg} \\ \text{person} & 3 \end{bmatrix} \end{bmatrix}$$



path in graph → features that lead through the feature structure to some value.

→ The graph we saw was tree-like i.e. each node had only 1 incoming edge. This need not be the case:-



This property of feature structures that several features can share 1 value is called reentrancy

This can be represented in matrix via indexing

$$\begin{bmatrix} \text{cat} & \text{S} \\ \text{head} & \begin{bmatrix} \text{agr} & (1)\begin{bmatrix} \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix} \\ \text{SUBJ} & \begin{bmatrix} \text{Agr} & (1) \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

~~Dependency Parsing~~ Unit 4
Semantic Analysis

Properties of knowledge base.
→ Verifyable (easy to map)
→ no ambiguity
→ no vagueness.
→ Should be in canonical form (multiple representation of same meaning are mapped to a single place).

→ non logical vocablury: is an open ended sets containing various set of objects & its properties.
→ Logical vocablury is a closed set that consists of operators, ~~quan~~ quantifiers etc.

For eg: For a restaurant's application
dishes, cuisines → non logical vocab.
add dish, remove dish → " "

eg.
Domain → a, b, c, d, e, f, g, h, i, j.

Matthew (a), Fanco (b), Katie (c), Carolina (d) customer

Franca (e), Med (f), Rio (g) Restaurant

Italian (h), Mexican (i), Indian (j). cuisine.

Properties
Noisy {e, f, g}

Relation

Mathew likes the Med

Kattie likes Med & Rias

likes = {(a, f), $\underbrace{(c, f), (c, g)}_{\text{write them seperately}}$}

Functions

Location of (Frasca)

these types of functions which map to appostrophe
" Frasca's location "

serves (Frasca, Indian)

for now:

I only have 5 dollar and I don't have a lot of time

in FOL is

have $\underbrace{\text{(speaker, 5 dollars)}}$ ^ ¬have (speaker, lot of time)

↳ 2nd argument is applied on 1st. always

⇒ Modus Ponens

→ Form of inference.

If given: α

and $\alpha \longrightarrow \beta$.

then β is also true

read d reduction

WST → nptel

Word Similarity

2 methods

→ wordnet

→ Distribution Model

In distribution model, we say that 2 words are similar if their distribution is same.

Distribution semantics means representation of a word in multiple dimensions (different contexts)

eg

Target voc: vocablury to be represented

Context words/term vocablury: the different contexts that form the multiple dimensions.

eg

Target vocablury: automobile, car, soccer, football

Context words: wheel transport passenger tournament goal

concurrency/Dist. matrix	wheel	transport	passenger	tournament	goal
automobile	1	1	0	0	0
car	1	2	1	0	0
soccer	0	0	0	1	1
football	0	0	1	1	2

"2" means that car & transport occur in any order (may even not be together) in context window twice in corpus.

Q. Convert in FOL.

1) All birds fly.

$\forall x \; Bird(x) \rightarrow fly(x)$

2) All man respect their parents.

$\forall x \; Man(x) \rightarrow respect(x, parent)$

3) Some boys play cricket

$\exists x : Boy(x) \rightarrow plays(x, cricket)$

(7)

4) ~~All~~ not all students like ~~books~~ both, maths & science.

~~$\forall x$: Student(x)~~

$\exists x$: student(x) $\longrightarrow$ (likes(x, math) XOR likes(x, science))

5) Moddus Pounens:

Marcus was a man.

Marcus was a pompian

All pompians were Romans.

Ceasar was a ruler

All pompians were either loyal to Ceaser or hated him.

Everyone is loyal to someone

People only try to assasinate rulers. They are not loyal too.

Marcus tried to assasinate Ceaser

Infer if Marcus was loyal to Ceaser.

a) Man(Marcus)

Pompian(Marcus)

$\forall x$: Pompian(x) $\longrightarrow$ Roman(x)

Ruler(Ceaser)

$\forall x$ Pompian(x) $\longrightarrow$ (loyal(x, Ceaser) $\vee$ Hate(x, Ceaser))

$\forall x \exists y$ Loyal(x, y)

~~$\forall x \forall y$ Assasinate(x, y) $\longrightarrow$ $\neg$ loyal(x, y)~~ what is x & y?

~~Assasinate(Marcus, Ceasar)~~

$\forall x \forall y$ Person(x) $\wedge$ Ruler(y) $\wedge$ assasinate(x, y) $\rightarrow$ $\neg$ loyal(x, y)

add a rule Person(x) $\rightarrow$

Man(x) $\rightarrow$ Person(x)

Thematic Roles: Ch 19 19.1 (Pg 382 of whole pdf)
374 of book!

~~Lexical~~ Lexicalised PCFG. Till 4th point pdf on class.

⇒ Weakness of PCFG as parsing method

→ Lack of sensitivity to Lexical Information

→ " " " Structural preferences.

Feature structures and unification of feature structure 13.1, 2, 3, 4 class

Unification of features structure

Before you combine 2 feature structure, you have to first check if they are compatible

link in comment of above ppt