Some terms

Machine translation - predicting which word is used is more frequent

Improvement in perplexity often correlates with improvement in speech recognition performance

Edit distance

Do row by row if letter is different, add substitution cost

min edit distance at any cell is the cost + 1 from the left cell

if letter is same take no cost from i-1, j-1 (diagonal)

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*'s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

	word	POS tag		word	POS tag
(a)	**Mary**	NNP	(f)	**Mr.**	NNP
(b)	**also**	RB	(g)	**resolved**	VBN
(c)	**bought**	VBD	(h)	**must**	MD
(d)	**10**	CD	(i)	**not**	RB
(e)	**apples**	NNS	(j)	**Every**	DT

Personal pronoun

PRP - He

Auxiliary

AUX - Does

Determiners (DT)

the, a, some, that, this

E.g. Does that

particles (RP) can appear after object

prepositions (IN) can appear only after verb

E.g. of, off, on

I thought that/IN

adjectives (JJ)

E.g. other, grand, already married/JJ

Verbs

VBD - commented

VBP - do, have

VBZ - is

There/EX are 70 children there/RB

Modal

MD - could

Personal Pronouns

PRP - you

PRP$ - your

Binary classification

sign(x) = 1 for x $\geq$ 0, -1 for x $<$ 0 Feature extraction so that there is a good mapping of x

Softmax

[2 -1 5]

$\frac{e^2}{e^2+e^{-1}+e^5}$ into another vector of 3 numbers where it sum to 1

Loss functions

$L_{cross-entropy}(\hat{y}, y) = -\sum_i ylog(\hat{y})$

or $-log(\hat{y}) forhardclassification$

$\hat{y}$ is 1 then it can minimize the loss function

$\hat{y}$ requires softmax for transformation

Ranking loss

$L_{ranking}(x, x') = max(0, 1 - (f(x) - f(x')))$

Gradient descent

$w_i \leftarrow w_i - \alpha \frac{\partial_L}{\partial w_i}$ for $\alpha > 0$

Successive iterative approximation, can't plug in a value like $w_{1,1}$

if L(w) is convex (single min point), then it will converge to global minimum

Expected number of bits

$\sum bits * P(bits)$ =1 $\times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8}$ ... = 2 Entropy is the number of bits needed to encode

Per word cross entropy

Entropy rate = $\frac{1}{n} H(w_1) = -\frac{1}{n}(\sum p(W) log(p(W))$

Smoothing

$P(w|c_i) = \frac{\#times\ w\ occur\ in\ texts\ of\ class\ c_i}{\sum \#timesw\ occurs\ in\ texts\ of\ class\ c_i}$

$P(w|c_i) = \frac{\#times\ w\ occur\ in\ texts\ of\ class\ c_i\ +1}{\sum \#times\ w\ occurs\ in\ texts\ of\ classc_i + V}$

$C^*(w_0 w) = \{C(w_0 w) + 1\} \times \frac{C(w_0)}{C(w_0)+V}$

Witten Bell

If C*(w$_x w_i$) > 0, $C(w_x) \times \frac{C(w_x w_i)}{C(w_x)+T(w_x)}$

If C*(w$_x w_i$) $= 0$, $\frac{c(w_x)T(w_x)}{Z(w_x)(c(w_x)+T(w_x))}$

Stochastic POS tagging

$P(T, W) = P(< s >, t_1, w_1, t_2, w_2 ... < s >)$

$= P(< s >) \cdot P(t_1 | < s >) \cdot P(w_1 | < s >, t_1)$

$P(T|W) = \frac{P(T,W)}{P(W)} = P(T, W)$

Markov assumption

w_k only depends on the previous n - 1 words

$P(w_k | w_1, .., w_{k-1}) \approx$P(w$_k | w_{k-1}$)

Vertibi

v(tag, word) = P(w$_i | t_i$) $\times P(t_i | t_{i-1}) \times P(t_{i-1})$

$Trigram P(t_i | t_{i-1} t_{i-2}) = P(t_i | t_{i-1} t_{i-2}) + P(t_i | t_{i-1}) + P(t_i)$

Forward computation

1. Look at the number of input nodes
2. Compute the s node which is the value before there is actually h_1(non-linear activation function)

$s_i = w_x i_i + w_{x+1} i_{i+1} ... + w_k i_k + ... b_i$
For the hidden layers i_i will be h_i

$h_i = \frac{1}{1+e^{-s_i}}$
Final value h_i will be o_i

$L = \frac{1}{2}[(o_1 - t_1)^2 + (o_2 - t_2)^2]$

Backward computation

Base case

1. Take the s values previously computed
2. Calculate $\frac{\partial_L}{\partial_{w_m}} = \frac{\partial_L}{\partial_{s_1}} \frac{\partial_{s_1}}{\partial_{w_m}}$

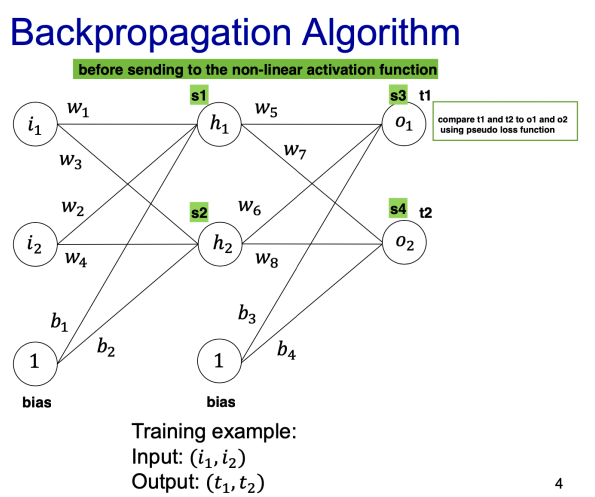
E.g. $\frac{\partial_L}{\partial_{w_6}}$

$\frac{\partial_L}{\partial_{s_3}} = (o_1 - t_1) \times o_1(1 - o_1)$
$\frac{\partial_{s_3}}{\partial_{w_6}} = h_2$

Recursive case

E.g. $\frac{\partial_L}{\partial_{w_2}}$

$\frac{\partial_L}{\partial_{w_2}} = \frac{\partial_L}{\partial_{s_1}} \times i_2$
$\frac{\partial_L}{\partial_{s_1}} = [\frac{\partial_L}{\partial_{s_3}} \times w_5 + \frac{\partial_L}{\partial_{s_4}} \times w_7] \times h_1(1 - h_1)$



Answer to Question 4:

$v(V, \text{water}) = v_1(1) = P(\text{water} \mid V) \times P(V \mid \text{<s>}) \times P(\text{<s>}) = \frac{1}{20} \times \frac{3}{4} \times 1 = \frac{3}{80}$

$v(N, \text{water}) = v_1(2) = P(\text{water} \mid N) \times P(N \mid \text{<s>}) \times P(\text{<s>}) = \frac{1}{50} \times \frac{1}{4} \times 1 = \frac{1}{200}$

Since

$v(V, \text{plants}) = v_2(1)$
$= P(\text{plants} \mid V) \times \max\{P(V \mid V) \times v1(1), P(V \mid N) \times v1(2))\}$
$= \frac{1}{50} \times \max\{\frac{2}{5} \times \frac{3}{80}, \frac{1}{6} \times \frac{1}{200}\}$
$= \frac{3}{10000}$

Therefore, edge from V_1 and V_2 is chosen.

Since

$v(N, \text{plants}) = v_2(2)$
$= P(\text{plants} \mid N) \times \max\{P(N \mid V) \times v1(1), P(N \mid N) \times v1(2))\}$
$= \frac{1}{10} \times \max\{\frac{1}{5} \times \frac{3}{80}, \frac{2}{3} \times \frac{1}{200}\}$
$= \frac{3}{4000}$

Therefore, edge from V_1 and N_2 is chosen.

Since

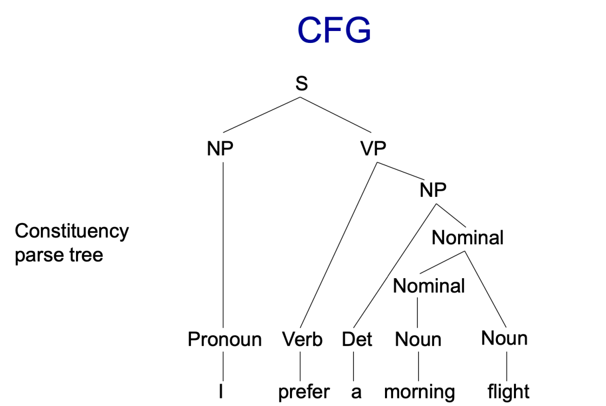
$\max\{P(\text{</s>} \mid V) \times v_2(1), P(\text{</s>} \mid N) \times v_2(2)\} = \max\{\frac{2}{5} \times \frac{3}{10000} \times 1, \frac{1}{6} \times \frac{3}{4000} \times 1\}$
$= \frac{1}{8000}$

Therefore, edge between N_2 and </s> is chosen.

Hence, path: <s> → V_1 → N_2 → </s> is the chosen path.

Therefore, the optimal sequence of part-of-speech tags is **<s>, V, N, </s>**

CFG



Bracketed notation

$[_S[_{NP}[_{Pronoun}\ \text{I}]][_{VP}[_{Verb}\ \text{prefer}][_{NP}[_{Det}\ \text{a}][_{Nominal}\ [_{Nominal}[_{Noun}\ \text{morning}]]\ [_{Noun}\ \text{flight}]]]]]$

Det → that | this | a
Noun → book | flight | meal | money
Verb → book | include | prefer
Pronoun → I | she | me
Proper-Noun → Houston | NWA
Aux → does
Prep → from | to | on | near | through

CKY - Bottom up

1. Define base case where non-terminal expands into a word
2. For a computed cell, try every combination for e.g. [1, 5] is there a rule at cell [1, 5] that expands to [1, 2] and [2, 5], [1, 3] and [3, 5]

'a flight through' is not a constituent then there would be no symbol assigned

Earley algorithm

1. Write down in chart[0] where RHS is non-terminal symbol

!Note! Remember to add in • in front of POS tag to make it a dotted rule

2. Add the single rule where the RHS matches the word at the location

For e.g. Book that flight
Add Verb ← book • Chart[1]
Include recursively where LHS is found in the RHS of the grammar rule

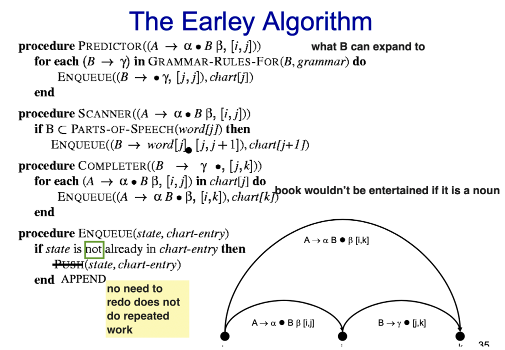
Do not add repeated rules to same chart
In the same chart expand all grammar rules that are after •
Combine rules from previous chart and current chart for combined rules matches CFG

3. Terminate when [0..n] and $S \leftarrow$ some rule •

The Earley Algorithm

does not require CNF

```
function EARLEY-PARSE(words, grammar) returns chart
  kick start the whole process, looking for a S
  ENQUEUE((γ → • S, [0,0]), chart[0])
  for i ← from 0 to LENGTH(words) do
   for each state in chart[i] do
     if INCOMPLETE?(state) and
              NEXT-CAT(state) is not a part of speech then
          PREDICTOR(state)   pos expands to the word
     elseif INCOMPLETE?(state) and   that symbol to right of dot is non-terminal symbol
              NEXT-CAT(state) is a part of speech then
          SCANNER(state)
     else
          COMPLETER(state)
   end
  end
  return(chart)
```

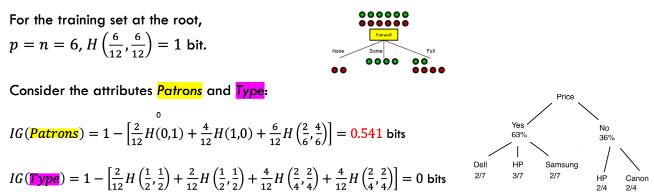


Grammar rules, non-terminal symbol
NP - Noun phrase
VP - Verb phrase

Either have 2 non-terminal symbols or 1 terminal symbol (words)

Garden path sentences
Reduced relative clause (which was)
The horse raced past the barn fell
raced used in the passive voice

Decision tree



Every regular language is context free
Union: $R = R_1 + R_2$
$S \rightarrow S_1 | S_2$
Then $S \rightarrow S_1$ and $S \rightarrow S_2$

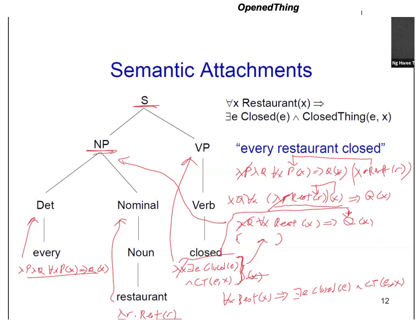
Concat: R = $R_1 R_2$
S $\rightarrow S_1 S_2$

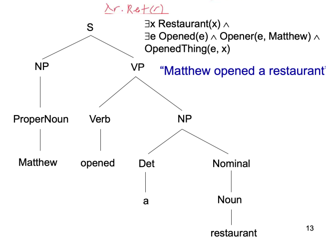
Kleene star:
$S \rightarrow S_1 S_1 | \epsilon$

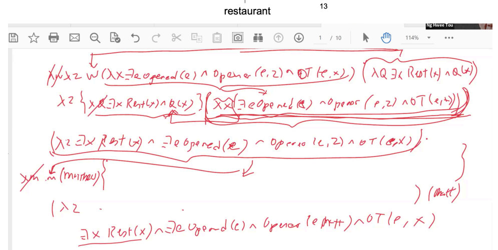
Perplexity
$P(w_1, ..., w_n)^{-\frac{1}{n}}$
If P(hello) = 1/4, P(there) = 1/4 and there are 30000 names with probability 1/20
Then $P(\frac{1}{4} + \frac{1}{4} + \frac{1}{20})^{-\frac{1}{3}}$

Semantic Representation
If CFG, then what is RHS becomes the LHS of the semantic representation
1. Determine the POS tags of each word
2. Propagate up if there is only 1 RHS
3. Apply rules in order, remove first lambda and apply wherever f appears E.g.
NP.sem(VP.sem) then λ f · f(Fransca) (λ x ∃ e Closed(e) ...)
(λ x ∃ e Closed(e) ...) (Frasca)

Grammar Rule	Semantic Attachment
$S \rightarrow NP\ VP$	$\{NP.sem(VP.sem)\}$
$NP \rightarrow Det\ Nominal$	$\{Det.sem(Nominal.sem)\}$
$NP \rightarrow ProperNoun$	$\{ProperNoun.sem\}$
$Nominal \rightarrow Noun$	$\{Noun.sem\}$
$VP \rightarrow Verb$	$\{Verb.sem\}$
$VP \rightarrow Verb\ NP$	$\{Verb.sem(NP.sem)\}$ use RHS semantic to determine LHS
$Det \rightarrow every$	$\{\lambda P.\lambda Q.\forall x P(x) \Rightarrow Q(x)\}$
$Det \rightarrow a$	$\{\lambda P.\lambda Q.\exists x P(x) \wedge Q(x)\}$
$Noun \rightarrow restaurant$	$\{\lambda r.Restaurant(r)\}$
$ProperNoun \rightarrow Matthew$	$\{\lambda m.m(Matthew)\}$
$ProperNoun \rightarrow Franco$	$\{\lambda f.f(Franco)\}$
$ProperNoun \rightarrow Frasca$	$\{\lambda f.f(Frasca)\}$ expecting something to close
$Verb \rightarrow closed$	$\{\lambda x.\exists e Closed(e) \wedge ClosedThing(e,x)\}$
$Verb \rightarrow opened$	$\{\lambda w.\lambda z.w(\lambda x.\exists e Opened(e) \wedge Opener(e,z) \wedge \cancel{Opened}(e,x))\}$ OpenedThing



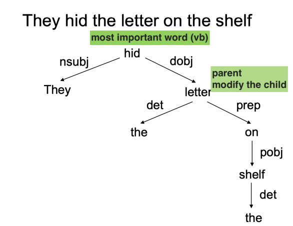




Constituents Parse trees
Head word are propagated up the parse tree (RHS to LHS)
$NP \rightarrow DT\underline{NN}$ RHS will always be non-terminal symbol unless there is POS tag on LHS
E.g. S $\rightarrow$ NP VP
Sometimes there are no direct objects after the verb
"he disappeared" instead of "he disappeared mary"
Verb decides what types of non-constituents

Head finding
1. Find the POS tag that expands to each word
2. For each grammar rule, add the head child with LHS to be the head
3. Add the head word in () beside the new head

Typed Dependency Parse
Parent is modified by the child
!Note! Do not forgot the edge has an arrow pointing from parent to child
Who hid?
The nominal subj is they and direct obj is letter
on the shelf are more auxiliary things



Argument Dependencies	Description
nsubj	nominal subject
csubj	clausal subject
dobj	direct object
iobj	indirect object
pobj	object of preposition
Modifier Dependencies	**Description**
tmod	temporal modifier
appos	appositional modifier
det	determiner
prep	prepositional modifier

Untyped Dependency Parse
1. Find all the head words
2. For every RHS, determine the most important head word then add the other symbols as the children