

17/4

N-gram	count
your	883,614
rights	80,891
doorposts	21
your rights	378
your doorposts	0

corpus $520 \cdot 10^6$ words = N

a) stima con MLE di $P(w_i | w_{i-1}) = \frac{P(w_i, w_{i-1})}{N}$

$$P(w_i | w_{i-1}) = \frac{378}{883,614} \left(= \frac{C(w_i, w_{i-1})}{C(w_{i-1}, *)} \right)$$

b) stima $P(w_i | w_{i-1})$; MLE con add-k smoothing
per $k=0.01$; $|V|=1,254,193$

$$P(w_i | w_{i-1}) = \frac{O + k}{N + |V| \cdot k} \Rightarrow C(w_i, w_{i-1}) \text{ aumenta di } k$$

\downarrow

$C(w_i)$ aumenta di k $\forall w \in \text{corpus}$

Argomento extra (non in syllabus)

SINTASSI: regole che governano struttura di frasi -> determinata da mente

CONSTITUENT/PHRASE: gruppo di parole che fungono da unità in struttura gerarchica
(es. "He saw the house on the hill" -> "He", "the house", "on the hill")

Per identificarli, CONSTITUENCY TESTS -> più importanti: noun phrase, verb phrase

Parte fondamentale di NLP: AMBIGUITÀ -> ambiguità in syntactic parsing che influenza interpretazione: problema di PP ATTACHMENT

Si può rappresentare struttura sintattica come albero

19/4

EX1:

The/Det program/N can/N deal/N with/Prep three/Num types/V of/Prop inputs/N ./Period

a) accuracy? $\begin{matrix} \text{verb} \\ \text{aux} \end{matrix}$ $\begin{matrix} \text{verb} \\ \text{verb} \end{matrix}$

$$N=10; 7/10 \text{ corretti} \Rightarrow \text{accuracy} = 7/10 = 70\%$$

b) commento su accuratezza w.r.t. state-of-the-art?

per molti ENG corpora, arrivati a 97% accuratezza
(naive tagging: 91-93%)

EX2:

Training set:

1. the/Det green/Adj bottle/NN leaked/VVD ./Punct
2. the/Det suppliers/NN bottle/VVB water/NN ./Punct
3. green/Adj water/NN suppliers/NN bottle/VVB ./Punct

(clan da Pantry set, per free-structure grammar)

Training set \Rightarrow corretto, no errori

e.g. $\begin{matrix} \text{Verb} \\ \text{V} \\ \text{by} \\ \text{specificatori} \end{matrix}$

a) vogliamo usare HMM per training con questo set
stimare transition e emission prob.?

TRANSITION: 6 label, 6 stati \Rightarrow solo transizioni presenti in set

$$P(\text{Adj}_i | \text{Det}) = \frac{C(\text{Det}|\text{Adj}_i)}{C(\text{Det})} = \frac{1}{2} \quad P(\text{NN} | \text{Adj}_i) = \frac{2}{2} = 1$$

$$P(\text{NN} | \text{Det}) = \frac{1}{2} \quad P(\text{VVB} | \text{NN}) = \frac{2}{5} \quad P(\text{NN} | \text{NN}) = \frac{1}{5}$$

$$\text{EMISSION: } P(\text{the} | \text{Det}) = \frac{C(\text{the} | \text{Det})}{C(\text{Det})} = \frac{2}{2} = 1 \quad P(\text{bottle} | \text{NN}) = \frac{1}{5}$$

$$P(\text{suppliers} | \text{NN}) = \frac{2}{5} \quad P(\text{water} | \text{NN}) = \frac{2}{5}$$

LLM

(Pre)training: fattibile solo da aziende \Rightarrow voi danno modelli con molta conoscenza,
ma non specializzata

allora fine-tuning nel contesto / task

LLM: con molti parametri, IN-CONTEXT LEARNING anziché fine-tuning

storia a LLM per focalizzare modello su task,
poi si fa domanda (PROMPT); modello prende
parola che è risposta che vogliamo
(e.g. "come è la recensione?" \Rightarrow positivo/negativo)

Dopo pretraining: se abbiamo domande, prompting \Rightarrow ma se ne ha troppe, fine-tuning

solli per farlo fare a \Leftarrow molti solli, servono
aziende molti esempi che rivelano
dati importanti (e.g. di
clienti)

e.g. Supervised FT, reinforcement
learning (e.g. chatbots)

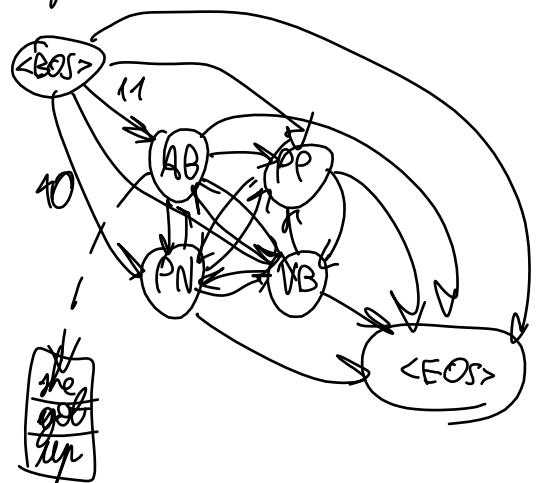
3/S

Esercizio:

	AB	PN	PP	VB	<EOS>
<BOS>	11	10	12	11	2S
AB	11	11	1P	10	1S
PN	11	12	12	10	16
PP	13	11	12	15	18
VB	11	10	10	13	15
	<BOS>	she	get	up	<EOS>
<BOS>	0	-	-	-	-
AB	-	-	-	-	-
PN	-	-	-	-	-
PP	-	-	-	-	-
<EOS>	-	-	-	-	-

	she	get	up	
AB	2S	2S	1S	
PN	13	2S	2S	
PP	2S	2S	13	
VB	2S	1S	1S	

Algoritmo di Viterbi

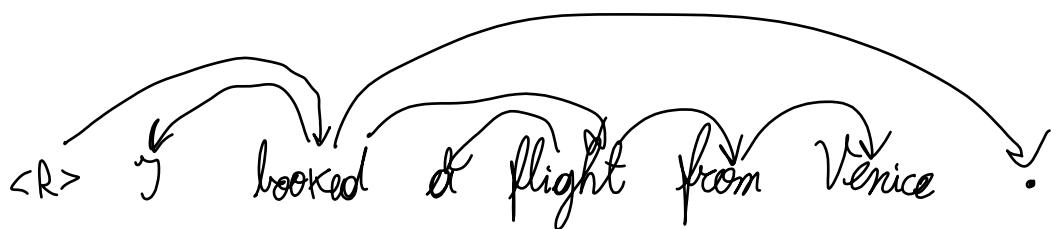


10/5

Ese:

I booked a flight from Venice.

head	booked <R> flight booked flight from booked
dep	I booked a flight from Venice .



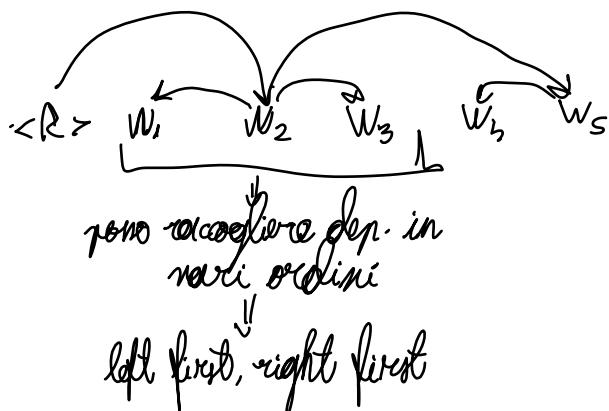
prendere un standard parser, applicare oracle e tradurre allora in sequenza di configurazioni

conf	stack	buffer	action
c ₀	([<R>], [I, booked, ...])		shift
c ₁	([<R>, I], [booked, a, ...])		shift
c ₂	([<R>, I, booked], [a, flight, ...])		leftarc (I ← booked)
c ₃	([<R>, booked], [a, flight, ...])		shift (booked non ha ricevuto tanti dep: no rightarc)
c ₄	([<R>, booked, a], [flight, from, ...])		shift
c ₅	([<R>, booked, a, flight], [from, Venice, ...])		leftarc (a ← flight)
c ₆	([<R>, booked, flight], [from, Venice, ...])		shift
c ₇	([<R>, booked, flight, from], [Venice, ...])		shift
c ₈	([<R>, booked, flight, from, Venice], [...])		rightarc (from → Venice)
c ₉	([<R>, booked, flight, from], [...])		rightarc (flight → from)
c ₁₀	([<R>, booked, flight], [...])		rightarc (booked → flight)
c ₁₁	([<R>, booked], [...])		shift
c ₁₂	([<R>, booked, .], [...])		rightarc (booked → .)
c ₁₃	([<R>, booked], [...])		rightarc (<R> → booked)
c ₁₄	([<R>], [...])		END (configurazione finale)

17/S

head	w_2	$\langle R \rangle$	w_2	w_s	w_s
dep	w_1	w_2	w_3	w_4	w_s

definire spazi di ambiguità
prendiamo arc-standard parser



Procedere sequenze diverse
per costituire albero
e.g. left-dep first, right-dep first

- left-dependent first (economical strategy)

stack	buffer	azione	
$\langle R \rangle$	$[w_1, w_2, \dots]$	sh	\Rightarrow iniziale: sempre stack con solo root
$\langle R \rangle, w_1$	$[w_2, \dots]$	sh	
$\langle R \rangle, w_1, w_2$	$[w_3, \dots]$	la	
$\langle R \rangle, w_2$	$[w_3, \dots]$	sh	
$\langle R \rangle, w_2, w_3$	$[w_4, w_5]$	ra	
$\langle R \rangle, w_3$	$[w_4, w_5]$	sh	
$\langle R \rangle, w_2, w_3, w_4$	$[w_5]$	sh	
$\langle R \rangle, w_2, w_3, w_4, w_5$	$[]$	la	
$\langle R \rangle, w_2, w_5$	$[]$	ra	
$\langle R \rangle, w_2$	$[]$	ra	
$\langle R \rangle$	$[]$	END	

- right-dependent first

stack	buffer	azione
[<R>]	[w ₁ , w ₂ , ...]	sh
[<R>, w ₁]	[w ₂ , ...]	sh
[<R>, w ₁ , w ₂]	[w ₃ , ...]	sh \Rightarrow combro
[<R>, w ₁ , w ₂ , w ₃]	[w ₄ , w ₅]	ra
[<R>, w ₁ , w ₂]	[w ₄ , w ₅]	sh
[<R>, w ₁ , w ₂ , w ₄]	[w ₅]	sh
[<R>, w ₁ , w ₂ , w ₄ , w ₅]	[]	ba
[<R>, w ₁ , w ₂ , w ₅]	[]	ra
[<R>, w ₁ , w ₂]	[]	ba
[<R>, w ₂]	[]	ra
[<R>]	[]	END

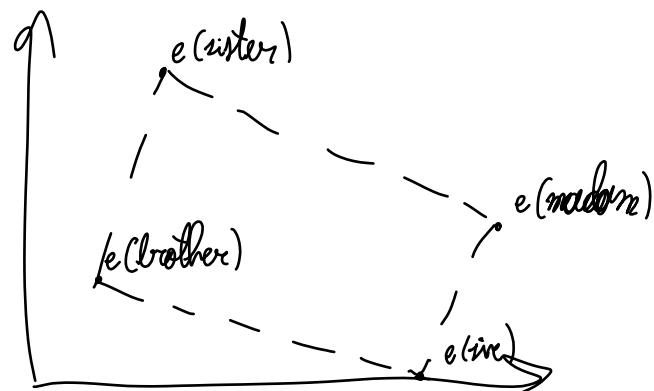
24/S

ES1:

disattore vector semantics

2: sister, brother, madam, sir

dotti embedding per queste parole, proiettarli in 2D, immaginare
scenari



primi tre: poniamo
metterli arbitrariamente
sir: deve formare relazione
con altri

disattore

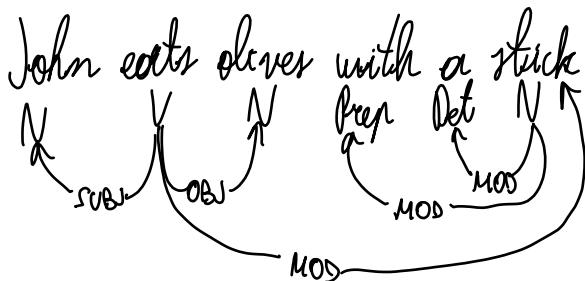
- ③ e(sister) / e(brother) \rightsquigarrow gender (differenza concetto da)
- ⑤ e(madam) / e(sister) \rightsquigarrow sibling rel. property

ES2: (dependency grammar)

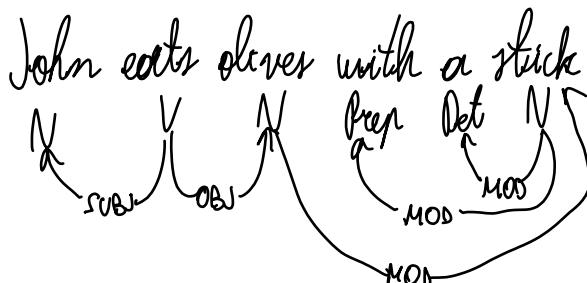
John eats olives with a stick

(frase ambigua, ma noi abbiamo una interpretazione)

fare analisi di due interpretazioni con dep. grammar



introp. 1 (standard)



PP-ATTACHMENT

introp. 2

