
ANNO ACCADEMICO 2024/2025

Tecnologie del Linguaggio Naturale

Teoria

Altair's Notes



**UNIVERSITÀ
DI TORINO**



DIPARTIMENTO DI INFORMATICA

CAPITOLO 1**INTRODUZIONE ALLE TECNOLOGIE DEL LINGUAGGIO NATURALE** **PAGINA 5**

1.1	Prologo	5
	La Complessità del Linguaggio Naturale — 6 • I Livelli di Conoscenza del Linguaggio — 8 • Strutture Linguistiche e Ambiguità — 9 • Lo Stato dell'Arte — 10	
1.2	I Livelli Linguistici	13
	Da Frase a Significato — 13 • Il Livello Morfologico e l'Analisi Lessicale — 15 • Il Livello Sintattico — 18 • Il Livello Semantico — 20 • Il Livello Pragmatico e del Discorso — 23	

CAPITOLO 2**SEQUENCE TAGGING****PAGINA 25**

2.1	Part of Speech (PoS) Tagging	25
	Perché studiare PoS? — 25 • Analisi Basata su Regole — 27 • ENGTWOL Lexicon — 27	
2.2	PoS Tagger Statistici	28
	HMM — 28 • MEMM e CRF — 31	

Premessa

Licenza

Questi appunti sono rilasciati sotto licenza Creative Commons Attribuzione 4.0 Internazionale (per maggiori informazioni consultare il link: <https://creativecommons.org/licenses/by/4.0/>).



Formato utilizzato

Box di "Concetto sbagliato":

Concetto sbagliato 0.1: Testo del concetto sbagliato

Testo contenente il concetto giusto.

Box di "Corollario":

Corollario 0.0.1 Nome del corollario

Testo del corollario. Per corollario si intende una definizione minore, legata a un'altra definizione.

Box di "Definizione":

Definizione 0.0.1: Nome delle definizioni

Testo della definizione.

Box di "Domanda":

Domanda 0.1

Testo della domanda. Le domande sono spesso utilizzate per far riflettere sulle definizioni o sui concetti.

Box di "Esempio":

Esempio 0.0.1 (Nome dell'esempio)

Testo dell'esempio. Gli esempi sono tratti dalle slides del corso.

Box di "Note":

Note:-

Testo della nota. Le note sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive.

Box di "Osservazioni":

Osservazioni 0.0.1

Testo delle osservazioni. Le osservazioni sono spesso utilizzate per chiarire concetti o per dare informazioni aggiuntive. A differenza delle note le osservazioni sono più specifiche.

1

Introduzione alle Tecnologie del Linguaggio Naturale

1.1 Prologo

La prima parte del corso sarà incentrata sulla linguistica computazionale generale, in cui ci si soffermerà sugli aspetti più tradizionali e linguistici¹. In questa parte verrà anche trattato il parsing. Nella seconda parte si andranno a studiare la semantica lessicale e le ontologie. Infine, nella terza parte del corso si andrà a studiare NLP statistico e distribuzionale.

Parte prima: keywords

-
- NLP
 - CL
 - Lexicon
 - Morphology
 - Syntax
 - semantics
 - Conversational Interface
 - Conversational agent
 - Dialogue System
 - Parsing
 - NLG
 - MT
 - Grammar
 - Treebank
 - NL ambiguity
 - BOT
 - LLM

Figure 1.1: Il giorno prima dell'esame bisogna sapere cosa significano tutte queste parole :3

¹Libro di riferimento: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. La prima e la seconda edizione, perché Jurafsky non riesce a finire il draft della terza :(

Le 4 ere della linguistica computazionale:

1. 1940 - 1969: primi tentativi.
2. 1970 - 1992: formalizzazione.
3. 1993 - 2012: apprendimento automatico.
4. 2013 - 2018: deep learning.

Note:-

Tutto cambiò nel 2018, quando NLP fu il primo successo su larga scala di rete neurale autosupervisionata.

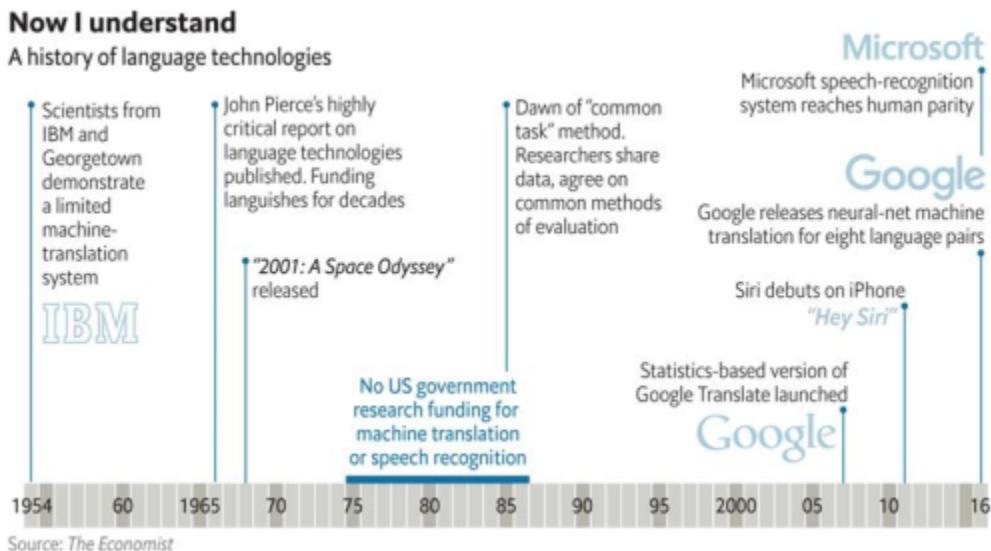


Figure 1.2: Il passato delle tecnologie del linguaggio naturale.

1.1.1 La Complessità del Linguaggio Naturale

C'è un legame tra linguaggio umano e intelligenza. Già Turing sosteneva che se si potesse parlare in un certo modo si fosse intelligenti (test di Turing). La differenza tra il linguaggio umano e un linguaggio di programmazione è l'*ambiguità*: C o Java non sono ambigui.

Il linguaggio umano:

- *Discretezza* (esistenza di elementi):
 - Api: Ritmo, orientamento, durata.
 - Esseri umani: Fonemi, morfemi, parole.
- *Ricorsività*:
 - Scimpanze: Gesti atomici.
 - Uomo: Gianni vede Pietro, Maria vuole che Gianni veda Pietro, Paolo crede che Maria voglia che Gianni veda Pietro.
- *Dipendenza dalla struttura*:
 - Non "una parola dietro l'altra" ma c'è una struttura: La ragazza parte, I ragazzi di cui mi ha parlato la ragazza partono.
- *Località*:
 - Gianni lo ha guardato.
 - Gianni ha detto che Pietro lo ha guardato.

Intelligenza e linguaggio nel il test di Turing:

- Possono le macchine pensare?
- Se riesco a parlare come un essere umano allora penso.
- Gioco dell'imitazione: un giudice deve capire se quello che ha davanti è un uomo oppure un computer.

Note:-

Ci sono una serie di obiezioni a questo test: teologia, matematica, coscienza, etc.

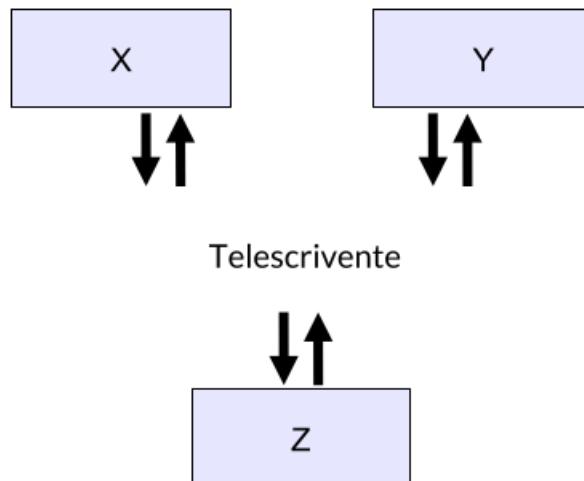


Figure 1.3: Il gioco dell'imitazione.

Nel 1966, Weizenbaum crea Eliza. Una macchina in grado di "comprendere" e ingannare gli esseri umani.

Note:-

Il punto debole del test di Turing e di Eliza è il giudice: se è coinvolto emotivamente potrebbe far passare un computer per un essere umano^a.

^aBlade runner moment

Definizione 1.1.1: Winograd Schema

Evoluzione del Turing test: un test a scelta multipla che utilizza domande con una specifica struttura. In questi test gli esseri umani sono molto bravi a rispondere, i computer no.

Note:-

Rimuove il giudizio, quindi tecnicamente più accurato.

Corollario 1.1.1 Captcha

Un test di Turing inverso per capire se l'interlocutore è umano. Non c'è linguaggio, ma riconoscimento cognitivo.

Corollario 1.1.2 Voight-Kampff Test

Test in Blade runner basato sulle emozioni, evoluzione del test di Turing.

1.1.2 I Livelli di Conoscenza del Linguaggio

HAL 9000, in "2001: Odissea nello spazio" mostra un esempio di comunicazione.

Domanda 1.1

Come fa HAL a rispondere?

- Riconoscimento vocale.
- Comprensione del linguaggio naturale.
- Generazione del linguaggio naturale.
- Sintesi vocale.
- Recupero ed estrazione di informazioni.
- Inferenza.

Livelli della conoscenza:

1. Il suono: HAL deve essere in grado di analizzare e produrre dei segnali audio che contengono le parole: foni e fonemi.
2. Le parole: HAL deve essere in grado di riconoscere le singole parole.
3. Raggruppare le parole: HAL deve essere in grado di distinguere la struttura della frase.
4. Significato: HAL deve conoscere il significato delle singole parole e deve essere in grado di comporre questi significati per trovare il significato complessivo della frase.
5. Contesto e scopi: HAL deve avere delle conoscenze del mondo che gli permettono usare il linguaggio in maniera contestuale: *I'm afraid, I can't* invece di *I won't*.
6. Conversazione: HAL deve avere deve essere in grado di conversare, dando delle risposte e facendo delle domande pertinenti al discorso.

A ogni livello corrisponde una parte del linguaggio:

1. Fonetica e Fonologia: lo studio del suono della lingua.
2. Morfologia: lo studio delle parti significative delle parole.
3. Sintassi: lo studio sulla struttura e sulle relazioni tra le parole.
4. Semantica: lo studio del significato.
5. Pragmatica: lo studio di come il linguaggio è usato per compiere goal. Il passivo serve per mettere in luce/enfatizzare alcune parti della frase.
6. Discorso: lo studio delle unità linguistiche rispetto alla singola dichiarazione.

Note:-

Jurafsky è un chad nerd.

1.1.3 Strutture Linguistiche e Ambiguità

Analizzando i vari livelli si trovano diverse *strutture linguistiche*.

Definizione 1.1.2: Struttura Linguistica

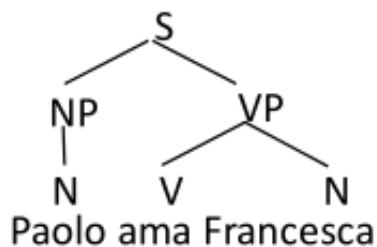
Una struttura è un insieme su cui è definita una relazione:

- Relazione fonetico-fonologica sull'insieme dei foni-fonemi.
- Relazione morfologica sull'insieme dei morfemi.
- Relazione sintattica sull'insieme delle parole.
- Relazione semantica sull'insieme dei significati delle parole.
- Relazione pragmatica sull'insieme dei significati delle parole e sul contesto.
- Relazione “discorsale” sull'insieme delle frasi.

Note:-

Ci sono relazioni tra i componenti della frase. Inoltre le relazioni cambiano a seconda della lingua.

Esempio 1.1.1 (Struttura Sintattica)



Definizione 1.1.3: Ambiguità

Il linguaggio naturale presenta frasi che possono essere interpretate in modi differenti.

Esempio 1.1.2 (Ambiguità)

"I made her duck"

- Ho cucinato una papera per lei.
- Ho cucinato una papera che apparteneva a lei.
- Ho creato una papera con la stampante 3D e gliel'ho data a lei.
- Ho fatto abbassare la sua testa.
- In Harry Potter^a: Ho trasformato lei in una papera.

^aRowling merda.

Osservazioni 1.1.1

- Le parole "duck" e "her" sono morfologicamente ambigue nella loro parte del discorso. "Duck" può essere un verbo o un nome, "her" può essere un pronome dativo o possessivo.
- Il verbo "make" è sinteticamente ambiguo: può essere transitivo o intransitivo.
- Inoltre "make" è anche semanticamente ambiguo: può significare creare o cucinare.
- In una frase parlata c'è un altro livello per cui "her" può essere udito come "eye" e "make" come "maid".

Note:-

Essere ambigui permette di essere brevi e concisi.

Altre proprietà notevoli del linguaggio:

- Linguaggio non standard, evolve nel tempo.
 - Scialla bros → chill → è easy.



- Segmentazione.
 - Il treno Torino San Remo.
- Locuzioni, spesso l'interpretazione non è compositiva.
 - Pollica verde.
- Neologismi.
 - Twettare²
- Conoscenza del mondo.
 - Lucia e Carola erano sorelle.
 - Lucia e Carola erano madri.
- Meta-linguaggio.
 - La prima cosa bella ha avuto un grandissimo successo.

1.1.4 Lo Stato dell'Arte

- 1976: In Canada un sistema riesce a stampare due bollettini meteo in due lingue diverse.
- BabelFish, di Yahoo, era un sistema "a regole" di trascrizione automatica, basato su Systran.
- 2011: IBM costruisce un supercomputer per battere un essere umano a Jeopardy, Watson.
- Tecnologie vocali: Speech Recognition, TextToSpeech, HTML5 Speech API (pagine web vocali).

²Musk merda.

Note:-

Dopo sette milioni e mezzo di anni Pensiero Profondo fornisce la risposta: "42"^a.

^aGuida Galattica per gli Autostoppisti.

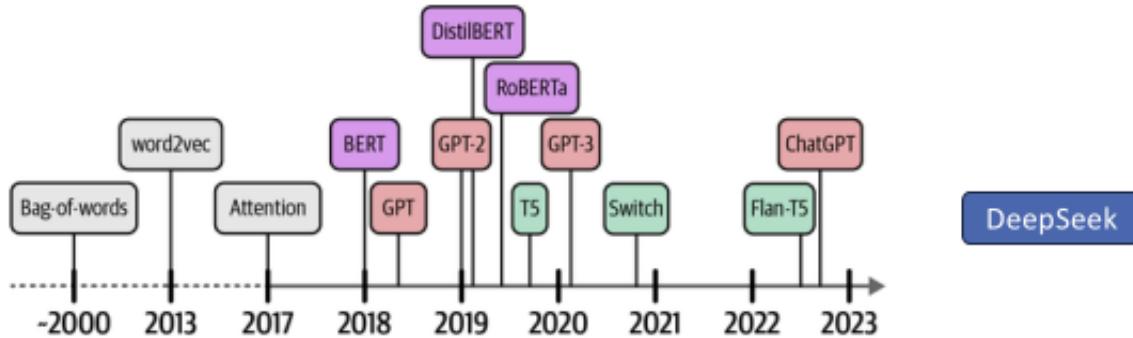


Figure 1.4: LLM. Tratto da "Hands-On Large Language Models", uscito nel Dicembre del 2024.

Note:-

Well, Deepseek è open source e funziona meglio di ChatGPT (a patto che non chiedi cosa sia successo a piazza Tienanmen nel 1989).

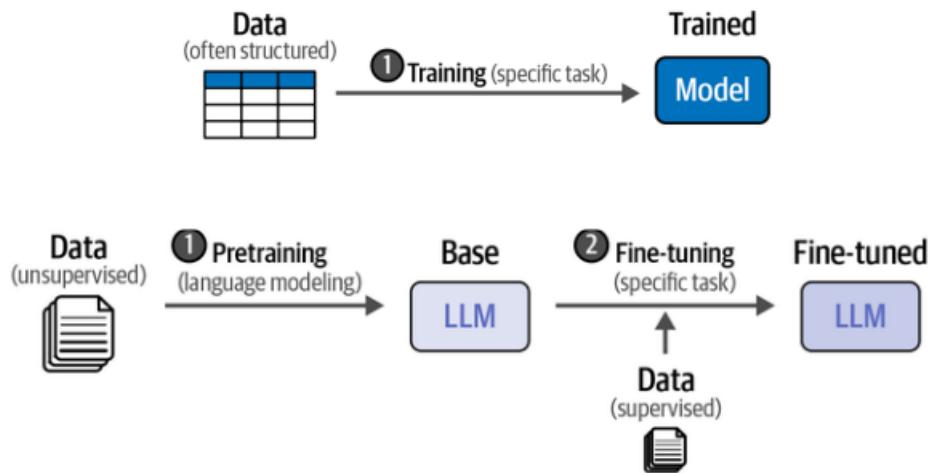


Figure 1.5: Shifting di paradigma dovuto al Machine Learning.

Definizione 1.1.4: AI Generativa

Modello di linguaggio di reti neurali multi-task basate sui transformer addestrati su una grande quantità di dati utilizzando self training e feedback umano.

- Modello di Linguaggio: Text prediction → T9.
- Multi-task: Google Translator, Siri.

Domanda 1.2

Come fare un LLM (M. Lapata)?

1. Collezionare una grande quantità di dati.
2. Chiedere al LLM di predire la nuova parola in una frase.
3. Ripetere il tutto.

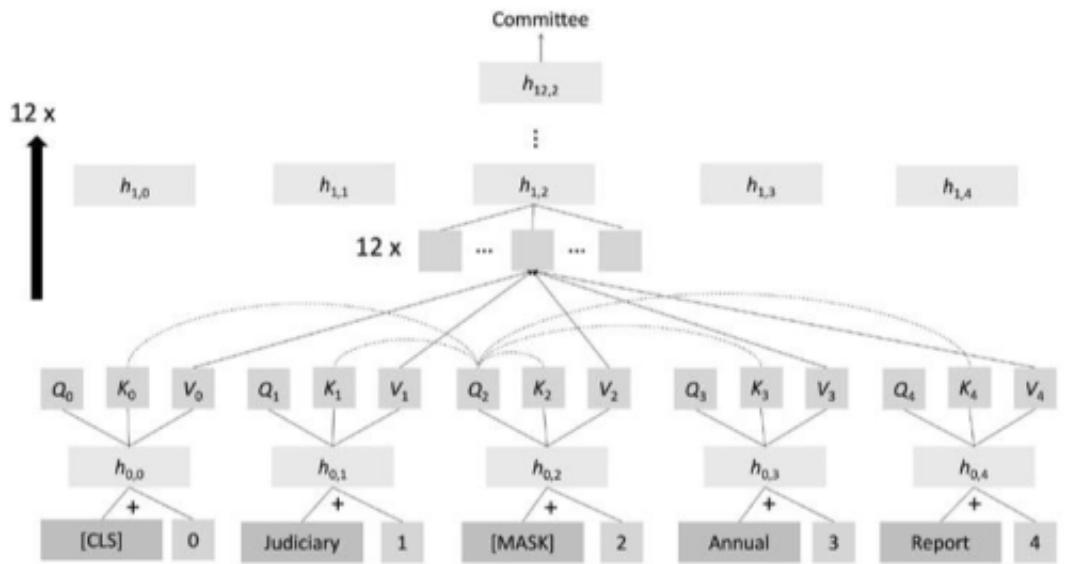


Figure 1.6: Auto addestramento di una rete neurale.

Domanda 1.3

Come usare un LLM?

- Sintonizzazione a grana fine.
- Prompting.

Si può usare un LLM per:

- Search Engine.
- Writer/Code assistant.

Note:-

Noam Chomsky odia questi sistemi. Secondo lui servono per evitare l'apprendimento.

DeepSeek:

- Apprendimento rinforzato automatico (senza essere umani).
- Meno costoso → politicamente importante.

Il problema fondamentale: Convertire una frase o un testo in una forma che permetta l'applicazione di meccanismi di ragionamento automatico.

1.2 I Livelli Linguistici

1.2.1 Da Frase a Significato

Problema: Convertire una frase o un testo in una forma che permetta l'applicazione di meccanismi di ragionamento automatico.

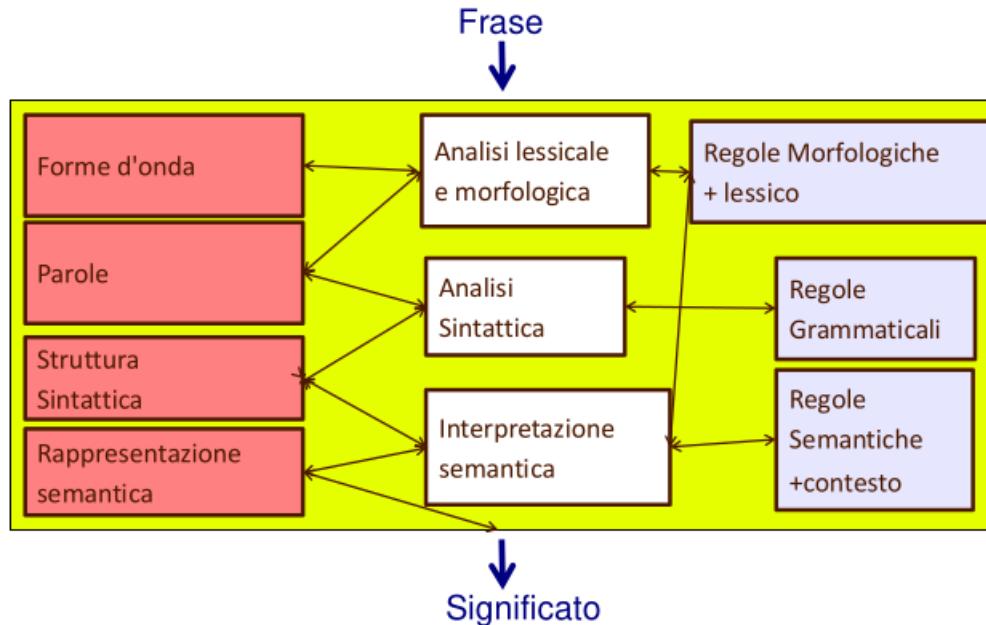


Figure 1.7: Passaggio da frase a significato.

Note:-

Però la situazione non è così semplice. Bisogna capire come funzionano i moduli e come comunicano

Nella linguistica computazionale c'è una divisione tra **regole** e **statistica**:

- Rules-driven.
- Data-driven.

Note:-

Steedman sostiene che i due aspetti dovrebbero convivere tra loro (2008). Evitare il regionamento tribale.

Domanda 1.4

Quando finisce una frase?

Definizione 1.2.1: Sentence splitting

Task in cui si deve capire quando una frase finisce.

- "!", "?" → Okay, pongono fine alla frase.
- ".":
 - Fine frase.
 - Abbreviazione (Doc., Mx.).
 - Numeri (0.2).

Domanda 1.5

Quindi come si costruisce un classificatore binario che decida EoS (End of String) o not EoS?

- Si possono scrivere regole a mano:
 - Espressioni regolari.
 - Tokenizer (FA) e regole.
- Addestrare un sistema di machine learning.

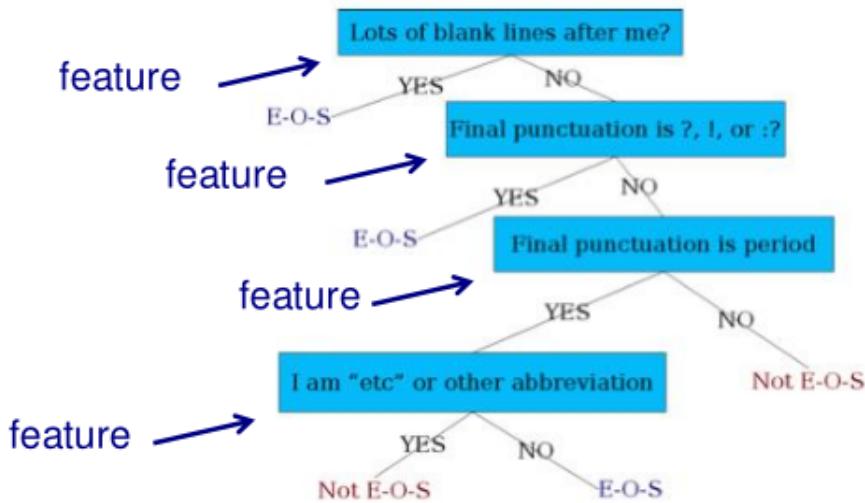


Figure 1.8: Albero di decisione.

Features più complesse:

- Caso di parole con ".".
- Caso di parole dopo ".".
- Features numeriche:
 - Lunghezza di parole con ".".
 - Probabilità che una parola con "." avvenga alla fine della frase.
 - Probabilità che una parola dopo "." avvenga all'inizio di una frase.

Domanda 1.6

Cos'è davvero un albero di decisione?

- Una serie di IF-THEN-ELSE encapsulati.
- Due possibilità per costruirlo:
 - *By-hand*: solo in contesti semplici.
 - *Machine learning*: su un training corpus.
- Il punto cruciale è la scelta delle features.

Osservazioni 1.2.1

- In questo corso ci concentreremo sullo studio delle feature linguistiche.
- In alcuni casi l'approccio by-hand verrà privilegiato poiché è didatticamente più chiaro/semplice e poiché è più semplice verificarne la fondatezza cognitiva mediante introspezione.

Domanda 1.7

Nei sistemi end-to-end cosa sono le feature linguistiche?

Definizione 1.2.2: Features Linguistiche Neurali

L'architettura neurale, ovvero il numero e il tipo di connessioni, codifica in maniera *implicita* le features linguistiche.

Note:-

La ricerca, in questo caso, si focalizza su quale scelta architettonica è più adatta alla modellazione implicita del fenomeno linguistico e alla creazione del corpus di training.

1.2.2 Il Livello Morfologico e l'Analisi Lessicale

Il lessico è fondato sul concetto di *parola*.

Domanda 1.8

Che cos'è una parola?

- Intuitivamente è una sequenza di caratteri delimitata da spazi o punteggiatura.
- Sequenze di più parole, Es. passammela = passa a me essa.
- Le parole hanno un significato unitario (semantica lessicale), ma volte sequenze di parole hanno un significato unitario. Es. di corsa, by the way.
- In altre lingue il problema è più grave.
 - In tedesco: Lebenversicherungsgesellschaftangestellter = impiegato di una società di assicurazione sulla vita.
 - In inglese: Wouldn't? = Would not.

Presenza di suffissi:

- CAPITANO (forma non declinabile).
- CAPITAN + O (nome o aggettivo o forma del verbo capitare).
- CAPIT + ANO (forma del verbo capitare).

Note:-

Non c'è una forma giusta a priori, ma c'è una forma giusta in base al contesto.

Definizione 1.2.3: Forme Composte

Generalmente una parola contenuto più una (o più) parole funziona.

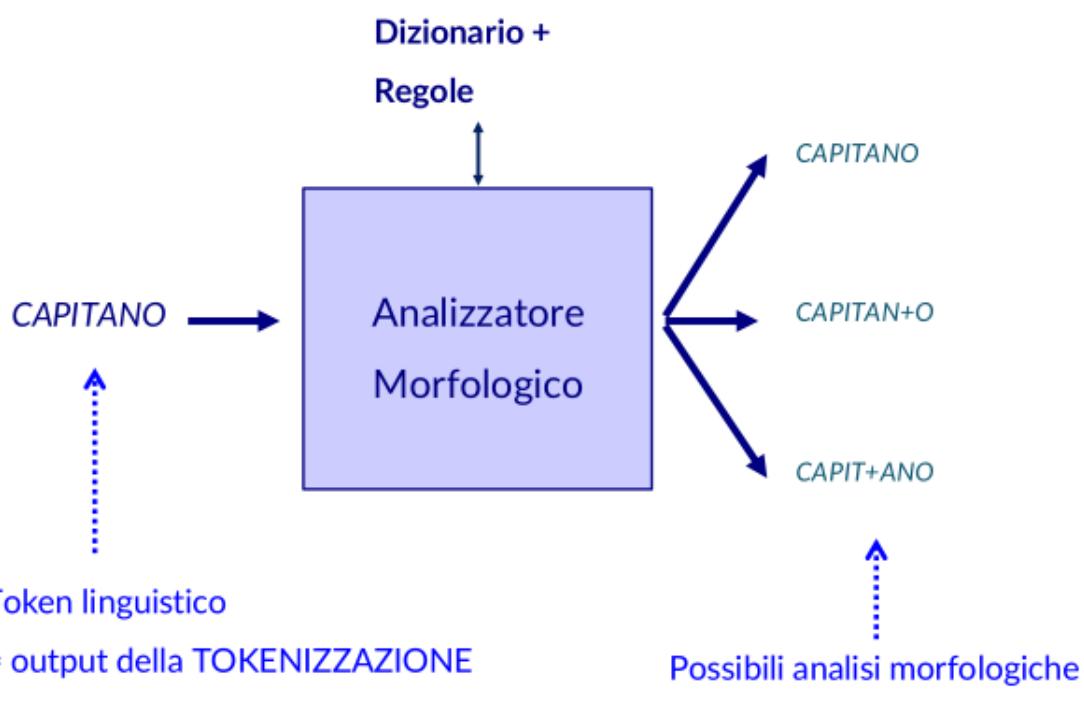


Figure 1.9: Analizzatore morfologico.

Esempio 1.2.1 (Forme composte)

STAMPAMELO:

- STAMP è una radice verbale.
- A è un suffisso verbale.
- ME e LO sono forme pronominali^a.

^aPer triggerare gli Alt-Right.

Definizione 1.2.4: Forme Multiple

Le diversi componenti sono nel dizionario ma la semantica non è composituale.

Esempio 1.2.2 (Forme multiple)

- Più o meno: puntatore tra le parole per recuperare la giusta semantica.
- Prendere un abbaglio: rimandare all'interprete semantico.

Definizione 1.2.5: Lemmatizzazione

Trasformare un lemma in forma normale.

Note:-

La forma normale non è stabile nel tempo.

Definizione 1.2.6: Stemming

Estrarre la forma radice (detta tema) da una parola.

Definizione 1.2.7: Paradigmatico

Si cambia una parte della parola con una equivalente si ha una frase morfologicamente corretta.

Definizione 1.2.8: Sintagmatico

I rapporti che intercorrono tra gli elementi che si succedono nella frase i rapporti che intercorrono tra gli elementi che si succedono nella frase.

Nome:

- Persone, oggetti, luoghi.
- Proprietà sintagmatiche:
 - Comparire dopo gli articoli.
 - Avere un possessivo.
 - Avere un singolare o un plurale.
- Comuni, propri, di massa, contabili.

Verbo:

- Eventi, azioni, processi.
- Molte forme morfologiche.
 - Tempo.
 - Modo.
 - Numero.
- Tante categorie (ausiliari, modali, copula, etc.).

Aggettivi:

- Proprietà.

Avverbi:

- Modificano qualcosa, spesso verbi, ma anche altri avverbi o intere frasi.

Note:-

Nomi, verbi, aggettivi e avverbi sono *di contenuto*, che puntano a oggetti reali.

Definizione 1.2.9: Classi Aperte

Classi che aumentano o scompaiono nel tempo costantemente (nomi, verbi, aggettivi, avverbi).

Definizione 1.2.10: Classi Chiuse

Classi che aumentano o scompaiono con tempi lunghissimi.

Note:-

Un esempio di classi chiuse sono i pronomi: una volta, in inglese, la seconda persona singolare era "thou", attualmente "you" ha assunto sia il ruolo di seconda persona singolare che plurale.

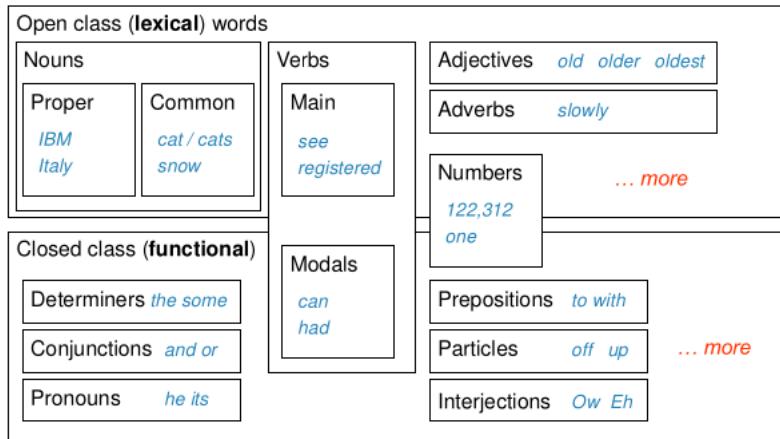


Figure 1.10: Parti aperte e parti chiuse.

Google Unviversal PoS: 12 PoS: NOUN (nouns), VERB (verbs), ADJ (adjectives), ADV (adverbs), PRON (pronouns), DET (determiners and particles), ADP (prepositions and postpositions), NUM (numerals), CONJ (conjunctions), PRT (particles), ‘.’ (punctuation marks) and X (a catch-all, e.g. abbreviations and foreign words).

1.2.3 Il Livello Sintattico

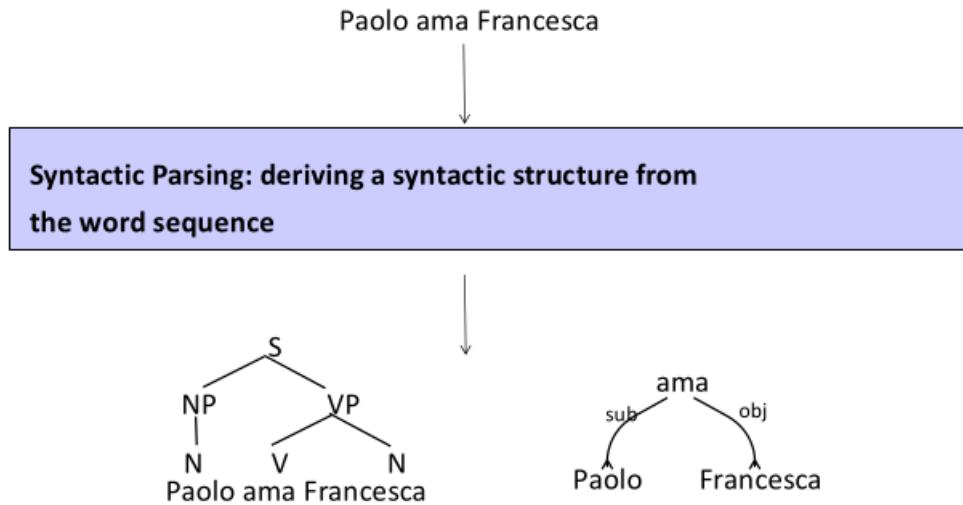


Figure 1.11: Parsing sintattico.

Note:-

Le due alternative derivano da prospettive diverse:

- Quella a sinistra è la struttura sintagmatica (o a costituenti).
- Quella a destra è la struttura a dipendenze (scuola di praga).

Entrambe le alternative sono equivalenti.

Definizione 1.2.11: Costituenza

La struttura della frase organizza le parole in costituenti annidati.

Domanda 1.9

Come si fa a sapere cos'è un costituente?

- Distribuzione: un costituente si comporta come un'unità che compare in differenti parti della frase.
- Sostituzione: test per verificare un costituent

Note:-

La cosa interessante è automatizzare il processo della costruzione di alberi.

Osservazioni 1.2.2

- NP: la parola più importante sintatticamente è un nome.
- VP: la parola più importante sintatticamente è un verbo.
- PP-LOC: la parola più importante sintatticamente è una preposizione.

- VP -> ... VB* ...
- NP -> ... NN* ...
- ADJP -> ... JJ* ...
- ADVP -> ... RB* ...
- SBAR(Q) -> S|SINV|SQ -> ... NP VP ...
- Plus minor phrase types:
 - QP (quantifier phrase in NP), CONJP (multi word constructions: as well as), INTJ (interjections), etc.

Figure 1.12: Parti della sintassi.

Note:-

I costituenti si comportano come un'unità:

- Esperimento di Fodor-Bever.
- Esperimento di Bock-Loebell.

Definizione 1.2.12: Context Free Grammar

I CFG mettono in relazione i *simboli non terminali* e i *constituenti* (Chomsky).

Definizione 1.2.13: X-barra

La teoria X-barra sostiene che se si costruisce un albero a costituenti con una determinata proprietà l'oggetto sarà presente internamente e il soggetto sarà presente esternamente.

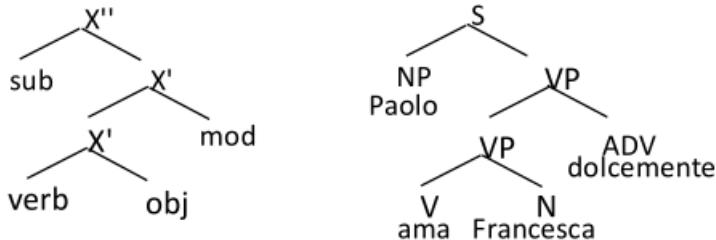


Figure 1.13: X-barra.

Definizione 1.2.14: Dipendenza

Relazione tra due parole:

- *Head*: parola dominante.
- *Dipendenza*: parola dominata.

Note:-

La testa seleziona le sue dipendenze e determina le loro proprietà.

Corollario 1.2.1 Argomenti

Modificano in maniera sostanziale un evento (obbligatori).

Corollario 1.2.2 Modificatori

Modificano parzialmente un evento (facoltativi).

1.2.4 Il Livello Semantico

Esistono 2 approcci alla semantica lessicale:

- Classico.
- Distribuzionale (anni '60):
 - Statistico.
 - Neurale.

Definizione 1.2.15: Semantica Lessicale Classica

Le connessioni sono legate al significato dei vari lessemi. La struttura interna dei lessemi è legata al significato.

Corollario 1.2.3 Lessema

Una coppia *forma-significato*, elemento del lessico.

Note:-

Il problema è che sono possibili definizioni ricorsive "infinite".

Relazioni tra lessemi:

- Omonimia: 2 lessemi con la stessa forma ortografica hanno due sensi diversi.
 - A bank can hold the investments.
 - We can go on the right bank of the river.
- Polisemia: lo stesso lessema ha due sensi diversi:
 - A bank can hold the investments.
 - He got the blood from the bank.
- Sinonimia: due lessemi con forma diversa hanno lo stesso senso (sostituibilità).
 - How big is that plane?
 - How large is that plane?
- Iponimia: due lessemi di cui uno denota una sottoclasse dell'altro:
 - Automobile è un iponimo di veicolo.
 - Veicolo è un iperonimo di automobile.

Esempio 1.2.3 (Iponimia)

- Quella è un automobile → quello è un veicolo.
- (?) Quello è un veicolo → quella è un automobile.

Corollario 1.2.4 Syn-set

Insieme di relazioni tra lessemi, usato per costruire le mappe in worldnet.

Definizione 1.2.16: Semantica Distribuzionale (vettoriale)

Il significato di una parola è collegato alla distribuzione delle parole attorno a sé.

Esempio 1.2.4 (Semantica Distribuzionale)

- A bottle of tesguino is on the table.
- Everybody likes tesguino.
- Tesguino makes you drunk.

Si può inferire che *tesguino* sia un super alcolico.

I vettori:

- Lunghi (lunghezza 20.000-50.000).
- Sparsi (molti elementi sono zero).

I lean vectors:

- Piccoli (lunghezza 200-1000).
- Densi (molti elementi sono non-zero).
- Vettori più corti sono più facili da usare come features nel ML.

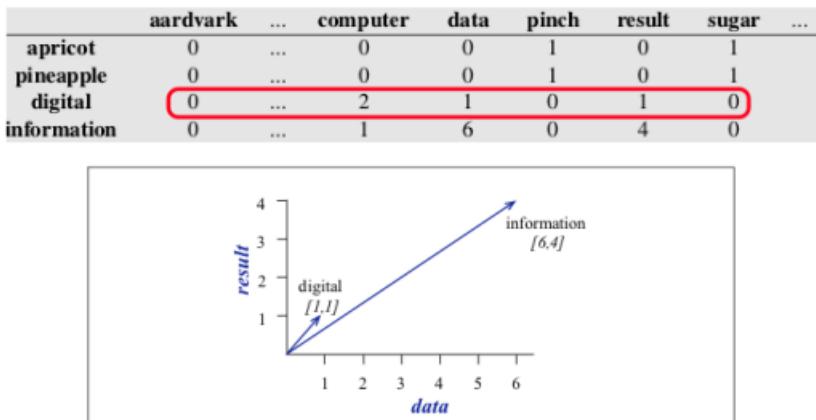


Figure 15.5 A spatial visualization of word vectors for *digital* and *information*, showing just two of the dimensions, corresponding to the words *data* and *result*.

Figure 1.14: Si può andare a determinare la "vicinanza" di parole con la semantica vettoriale.

Osservazioni 1.2.3

- Con l'avvento delle reti neurali si ha un miglioramento.
- $\text{vector}(\text{'king'}) - \text{vector}(\text{'man'}) + \text{vector}(\text{'woman'}) = \text{vector}(\text{'queen'})$.
- $\text{vector}(\text{'Paris'}) - \text{vector}(\text{'France'}) + \text{vector}(\text{'Italy'}) = \text{vector}(\text{'Rome'})$.

Definizione 1.2.17: Parole Contestualizzate

Costruire un vettore per ogni parola, condizionandolo al suo contesto. La rappresentazione per ogni token è una funzione dell'intera sequenza di input.

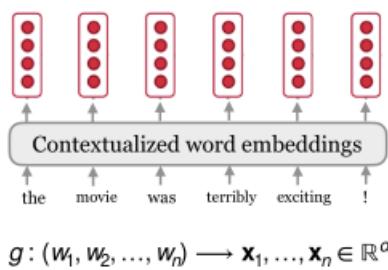


Figure 1.15: Contestualizzazione.

Corollario 1.2.5 Semantica Composizionale

La semantica di un sintagma è funzione della semantica dei sintagmi componenti; non dipende da altri

sintagmi esterni al sintagma stesso.

Note:-

Conoscendo il significato di X, Y, e +, possiamo comporre il significato “X+Y”.

Reasoning:

- Deduzione: conseguenza logica.
- Induzione: basata su molti casi, si assume una regola generale.
- Abduzione: regionamento per indizi.

Definizione 1.2.18: Metasemantica

L'insieme di semantica compositiva e semantica lessicale distribuzionale. Serve per dare senso a parole sconosciute.

1.2.5 Il Livello Pragmatico e del Discorso

Definizione 1.2.19: Pragmatica

L'interpretazione di “io” (sottinteso) e “oggi” dipende da chi enuncia la frase e quando, rispettivamente.

Note:-

Il vero significato deve essere integrato da oggetti metalinguistici.

Corollario 1.2.6 Anafora

Sintagmi che si riferiscono a oggetti precedentemente menzionati.

Esempio 1.2.5 (Anafora)

- “La torta era sul tavolo. Giorgio la divorò”.
- “In giardino c'erano il cane e il gatto che giocavano con un pezzo di stoffa. Il felino lacero' la stoffa”.
- “Dopo essersi fidanzati, Giorgio e Maria trovarono un prete e si sposarono. Per la luna di miele, essi andarono ai Caraibi”.

Le **strutture dati** dei livelli:

- Livello morfologico e l'analisi lessicale: Lista.
- Livello sintattico: Alberi.
- Livello Semantico:
 - Semantica lessicale: Insiemi, Vettori.
 - Semantica formale: Logica, Alberi/Grafi.
- Livello paradigmatico e del discorso: Frame, Ontologie.

2

Sequence Tagging

2.1 Part of Speech (PoS) Tagging

Definizione 2.1.1: PoS Tagging

Assegnare dei tag per distinguere le varie parti di una frase.

2.1.1 Perché studiare PoS?

Domanda 2.1

Perché studiare PoS?

- Text-to-Speech: la pronuncia di alcune parole cambia in base alla loro parte nel discorso.
- Scrivere regexps: per cercare le frasi principali.
- Input per un parser completo.
- MT (Machine Translation): riordinare aggettivi e nomi nelle traduzioni.
- Si potrebbe volere distinguere tra aggettivi o altre parti del discorso.
- Si potrebbe voler studiare cambiamenti linguistici come la ceazione di nuove parole o shifting del significato.

Domanda 2.2

Quanto è difficile il PoS Tagging?

- 85% delle parole non sono ambigue.
- 15% delle parole sono ambigue e molto frequenti (il 60% delle parole che si ascoltano sono ambigue).

Domanda 2.3

Quanti tag sono corretti?

- Attualmente 97%.

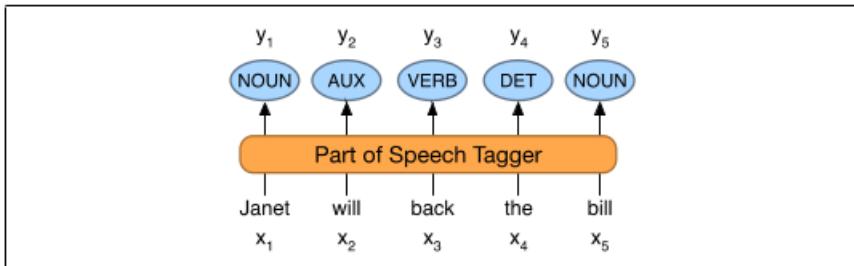


Figure 17.3 The task of part-of-speech tagging: mapping from input words x_1, x_2, \dots, x_n to output POS tags y_1, y_2, \dots, y_n .

Figure 2.1: Part of Speech Tagging.

- Una *baseline* del 92% è possibile con il metodo più banale:
 - Si dà un tag a ogni parola con il suo significato più frequente.
 - Si dà un tag nome alle parole sconosciute.

Tag	Description	Example
Open Class	ADJ Adjective: noun modifiers describing properties	<i>red, young, awesome</i>
	ADV Adverb: verb modifiers of time, place, manner	<i>very, slowly, home, yesterday</i>
	NOUN words for persons, places, things, etc.	<i>algorithm, cat, mango, beauty</i>
	VERB words for actions and processes	<i>draw, provide, go</i>
	PROPN Proper noun: name of a person, organization, place, etc..	<i>Regina, IBM, Colorado</i>
Closed Class Words	INTJ Interjection: exclamation, greeting, yes/no response, etc.	<i>oh, um, yes, hello</i>
	ADP Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	<i>in, on, by under</i>
	AUX Auxiliary: helping verb marking tense, aspect, mood, etc.,	<i>can, may, should, are</i>
	CCONJ Coordinating Conjunction: joins two phrases/clauses	<i>and, or, but</i>
	DET Determiner: marks noun phrase properties	<i>a, an, the, this</i>
	NUM Numeral	<i>one, two, first, second</i>
	PART Particle: a preposition-like form used together with a verb	<i>up, down, on, off, in, out, at, by</i>
	PRON Pronoun: a shorthand for referring to an entity or event	<i>she, who, I, others</i>
	SCONJ Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	<i>that, which</i>
Other	PUNCT Punctuation	<i>:, ;, ., !, ?, ,</i>
	SYM Symbols like \$ or emoji	<i>\$, %</i>
	X Other	<i>asdf, qwfg</i>

Nivre et al. 2016

Figure 2.2: Tagset.

Esempio 2.1.1 (*Janet will back the bill*)

- Probabilità a parole: "will" è di solito un ausiliario.
- Identità delle parole adiacenti: "the" implica che la prossima parola probabilmente non è un verbo.
- Morfologia e forma delle parole:
 - Prefissi.
 - Suffissi.
 - Capitalizzazione.

Algoritmi di supervised learning:

- Hidden Markov Models (programmazione dinamica).
- Conditional Random Fields (CRF) / Maximum Entropy Markov Models (MEMM).
- Natural Sequence Models (RNNs o transformers).
- Large Language Models.

2.1.2 Analisi Basata su Regole

Idee di Base:

1. Assegnare tutti i possibili tags alle parole (analisi morfologica).
2. Rimuovere i tags in base a un *insieme di regole*.
3. Solitamente più di 1000 regole scritte a mano (ma possono essere apprese automaticamente).

2.1.3 ENGTWOL Lexicon

Word	POS	Additional POS features
smaller	ADJ	COMPARATIVE
entire	ADJ	ABSOLUTE ATTRIBUTIVE
fast	ADV	SUPERLATIVE
that	DET	CENTRAL DEMONSTRATIVE SG
all	DET	PREDETERMINER SG/PL QUANTIFIER
dog's	N	GENITIVE SG
furniture	N	NOMINATIVE SG NOINDEFDETERMINER
one-third	NUM	SG
she	PRON	PERSONAL FEMININE NOMINATIVE SG3
show	V	IMPERATIVE VFIN
show	V	PRESENT -SG3 VFIN
show	N	NOMINATIVE SG
shown	PCP2	SVOO SVO SV
occurred	PCP2	SV
occurred	V	PAST VFIN SV

Figure 2.3: Esempio di ENGTWOL.

- Utilizzare un analizzatore morfologico per ottenere tutte le parti del discorso.

Example: *Pavlov had shown that salivation ...*

Pavlov	PAVLOV N NOM SG PROPER
had	HAVE V PAST VFIN SVO
	HAVE PCP2 SVO
shown	SHOW PCP2 SVOO SVO SV
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	N NOM SG

Figure 2.4: Primo passaggio.

- Si applicano i limiti.

Example: Adverbial "that" rule -> Given input: "that"

```

IF (and
    (+1 A/ADV/QUANT)          /* if next word is adj, adverb, or quantifier */
    (+2 SENT-LIM)             /* and following which is a sentence boundary, */
    (NOT -1 SVOC/A) )         /* and the previous word is not a verb like
                               'consider' which allows adjs as object
                               complements */
THEN
    eliminate non-ADV tags
ELSE
    eliminate ADV
  
```

Figure 2.5: Secondo passaggio.

2.2 PoS Tagger Statistici

Domanda 2.4

Viene fornita una frase (un'osservazione o una sequenza di informazioni). Qual è la migliore sequenza di tags che corrisponde a questa sequenza di osservazioni?

Vista probabilistica:

- Considera tutte le possibili sequenze di tags.
- Di questo universo di sequenze viene scelta la sequenza più probabile.

Terminologia:

- *Modelling*: fornire un modello formale.
- *Learning*: un algoritmo per impostare i parametri del modello.
- *Decoding*: un algoritmo per applicare il modello per calcolare risultati.

2.2.1 HMM

Si vuole modellare, di tutte le sequenze di n tags $t_1 \dots t_n$, la sequenza di tags tale che $P(t_1 \dots t_n | w_1 \dots w_n)$ è la maggiore.

$$\hat{t}_1^n = \operatorname{argmax} P(t_1^n | w_1^n)$$

Note:-

" $\hat{\cdot}$ " significa "la nostra stima del migliore".
 $\operatorname{argmax}_x f(x)$ significa "la x che massimizza $f(x)$ ".

Domanda 2.5

Ma come si utilizza questa equazione?

Inferenza Bayesiana: si usa la regola di Bayes per trasformare quest'equazione in un insieme di probabilità che sono facilmente calcolabili.

Note:-

Nell'ipotesi di Markov sono presenti approssimazioni, per rendere il tutto più facile da calcolare.

$$\begin{aligned}
 \hat{t}_1^n &= \underset{t_1^n}{\operatorname{argmax}} \overbrace{P(w_1^n | t_1^n)}^{\text{likelihood}} \overbrace{P(t_1^n)}^{\text{prior}} \\
 P(w_1^n | t_1^n) &\approx \prod_{i=1}^n P(w_i | t_i) \\
 P(t_1^n) &\approx \prod_{i=1}^n P(t_i | t_{i-1}) \\
 \hat{t}_1^n &= \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \approx \boxed{\underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})}
 \end{aligned}$$

Figure 2.6: Ipotesi di Markov, utilizzo dell'inferenza Bayesiana.

Learning:

• PoS → PoS:

- Si calcola la probabilità che una determinata parte di una frase ne preceda un'altra.
- Esempio: $P(NN|DT) = \frac{C(DT, NN)}{C(DT)}$ è la probabilità che un articolo (*DT*) preceda un nome (*NN*).
- In generale $P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$, dove C conta le occorrenze.

• PoS → World:

- Calcola la probabilità che una certa parola assuma una certa valenza (likelihood, probabilità di verosimiglianza).
- Esempio: $P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)}$, la probabilità che un verbo alla terza persona presente (*VBZ*) sia *is*.
- $P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$.

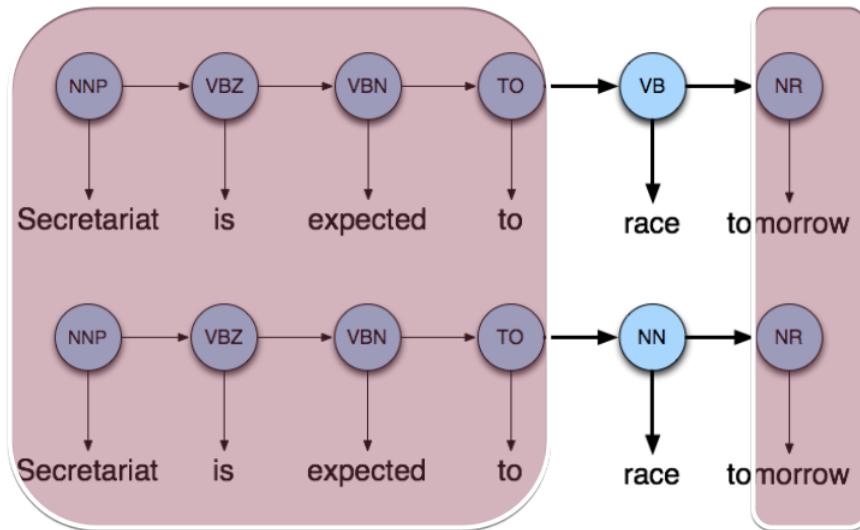


Figure 2.7: Disambiguare la parola "race".

Note:-

Si va a calcolare la probabilità della parola diversa.

Domanda 2.6

Come si fa a fare il decoding?

- L'algoritmo banale (che non funziona) porta a considerare tutte le possibili sequenze e andare a prendere la probabilità maggiore.
- Però con 30 tags e una frase di lunghezza media (20 parole) si hanno 30^{20} casi.

Janet/NNP will/MD back/VB the/DT bill/NN

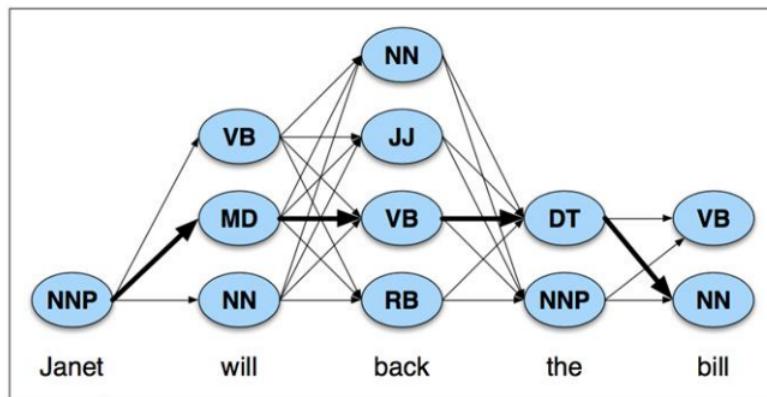


Figure 2.8: Tentativo di decoding.

Idee della programmazione dinamica:

- Si consideri una sequenza di stati (tag sequence) che termini con uno stato j con un particolare tag T .
- La probabilità che la tag sequence possa essere rotta in due parti:
 - La probabilità della migliore tag sequence attraverso $j - 1$.
 - Moltiplicata con la probabilità di transizione del tag alla fine della sequenza ($j - 1$) rispetto a T

Sommario di Viterbi:

- Crea una matrice:
 - Con colonne corrispondenti agli input.
 - Con righe corrispondenti ai possibili stati ($\text{PoS_Tag} + S_{ini} + S_{fin}$).
- Si attraversa la matrice riempendo le colonne a destra con i valori immediatamente a sinistra.
- Ogni cella della matrice, $v_t(j)$, rappresenta la probabilità che HMM sia nello stato j dopo aver visto le prime t parole e attraversato la sequenza di stati più probabile q_1, \dots, q_{t-1} .
- $v_t(j) = \max_{i=1}^N v_{t-1}(i)a_{i,j}b_j(o_t)$.

Con i simboli:

- $v_{t-1}(i)$: la probabilità della precedente iterazione dell'algoritmo di Viterbi.
- $a_{i,j}$: la probabilità di transizione da un precedente stato q_i allo stato corrente q_j .



Figure 2.9: Transition probability or something, idk.

- $b_j(o_t)$: la likelihood dello stato di osservazione del simbolo o_t dato lo stato corrente j .

function VITERBI(*observations* of len T ,*state-graph* of len N) **returns** *best-path*

```

create a path probability matrix viterbi[ $N+2,T$ ]
for each state  $s$  from 1 to  $N$  do ; initialization step
    viterbi[ $s,1$ ]  $\leftarrow a_{0,s} * b_s(o_1)$ 
    backpointer[ $s,1$ ]  $\leftarrow 0$ 
for each time step  $t$  from 2 to  $T$  do ; recursion step
    for each state  $s$  from 1 to  $N$  do
        viterbi[ $s,t$ ]  $\leftarrow \max_{s'=1}^N viterbi[s',t-1] * a_{s',s} * b_s(o_t)$ 
        backpointer[ $s,t$ ]  $\leftarrow \operatorname{argmax}_{s'=1}^N viterbi[s',t-1] * a_{s',s}$ 
    viterbi[ $q_F, T$ ]  $\leftarrow \max_{s=1}^N viterbi[s, T] * a_{s,q_F}$  ; termination step
    backpointer[ $q_F, T$ ]  $\leftarrow \operatorname{argmax}_{s=1}^N viterbi[s, T] * a_{s,q_F}$  ; termination step
return the backtrace path by following backpointers to states back in time from
backpointer[ $q_F, T$ ]
```



Figure 2.10: Algoritmo di Viterbi.

Note:-

La chiave di programmazione dinamica è che abbiamo bisogno solo memorizzare il percorso prob MAX in ogni cella e non in tutti percorsi.

2.2.2 MEMM e CRF

