POS Tagging e NER

CORSO DI NATURAL LANGUAGE PROCESSING (ELABORAZIONE DEL LINGUAGGIO NATURALE)

a.a. 2025/2026

Prof. Roberto Pirrone

Parti del discorso

- Dalle più antiche tradizioni linguistiche (Yaska e Panini V sec. a.C., Aristotele IV sec. a.C.), nasce l'idea che le parole possano essere classificate in categorie grammaticali.
  - Parti del discorso, classi di parole ovvero POS e POS tags

- 8 parti del discorso sono attribuite a Dionisio Trace di Alessandria (circa I sec. a.C.):
  - noun, verb, pronoun, preposition, adverb, conjunction, participle, article

- Queste categorie sono rilevanti per l'NLP oggi.

Due classi di parole: Aperte vs. Chiuse

- Parole di classe chiusa
- Solitamente sono *function words*: parole brevi, frequenti e con funzione grammaticale
    - *articoli*: **a, an, the**
    - *pronomi*: **she, he, I**
    - *preposizioni*: **on, under, over, near, by, …**

- Parole di classe aperta
- Solitamente sono *content words*: Nomi, Verbi, Aggettivi, Avverbi
- Includono le interiezioni: **oh, ouch, uh-huh, yes, hello**
- Si possono aggiungere nuovi nomi e verbi come like *iPhone* or *to fax*

Parole di classe aperta("content")

Nomi

Propri

Janet

Italy

Comuni

cat, cats

mango

Verbi

Principali

eat

went

Ausiliari

can

had

Aggettivi *old green tasty*

Avverbi *slowly yesterday*

Numeri

122,312

one

Interiezioni *Ow hello*

… more

Parole di classe chiusa

("function")

Articoli *the some*

Congiunzioni *and or*

Pronomi *they its*

Preposizioni *to with*

Particelle *off up*

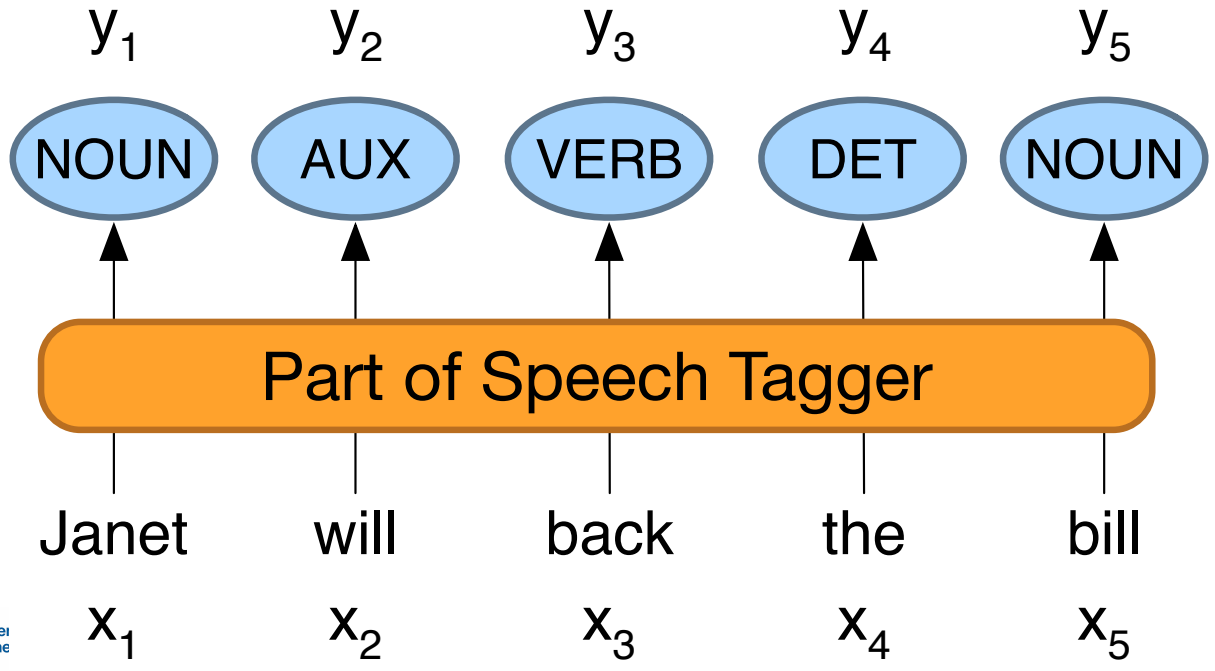
… more

Part-of-Speech Tagging

- Il task consiste nell'assegnare una *part-of-speech* a ogni parola di un testo.
- Spesso le parole possono avere più di un POS.
- ***book***:
  - VERB: (***Book*** *that flight*)
  - NOUN: (*Hand me that* ***book***).

Part-of-Speech Tagging

Si effettua una mappatura da una sequenza di parole x1,...,xn a una sequenza di tag POS y1,...,yn.



"Universal Dependencies" Tagset

Nivre et al. 2016

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

Esempi di frasi inglesi "Taggate"

- There/PRO were/VERB 70/NUM children/NOUN there/ADV ./PUNC
- Preliminary/ADJ findings/NOUN were/AUX reported/VERB in/ADP today/NOUN ’s/PART New/PROPN England/PROPN Journal/PROPN of/ADP Medicine/PROPN

Proviamo on line su https://lindat.mff.cuni.cz/services/udpipe/

Sample "Tagged" Italian sentences

Treebank: termine usato per indicare un corpus in cui la struttura sintattica/semantica delle frasi è annotata, permettendo di ottenere "alberi di parsing".

Esercizio:

Eseguire il parsing **manuale** delle seguenti frasi italiane usando il tagset della *UD Italian Stanford Dependency Treebank* e confrontare il risultato con il *POS tagger* online:

- Ieri l'altro ho visto Giovanni che prendeva un caffè

- M'illumino d'immenso

Perché il Part of Speech Tagging?

- Può essere utile per altri task di NLP:
  - **Parsing:** il *POS tagging* può migliorare il *parsing* sintattico.
  - **Machine Translation:** per il riordino di aggettivi e nomi (es. dallo spagnolo all'inglese).
  - **Sentiment Analysis:** per distinguere aggettivi o altri POS.
  - **Text-to-speech:** per decidere come pronunciare parole come "lead" o "object".

- Utile anche per task computazionali di analisi linguistica:
  - Per monitorare il POS nello studio di cambiamenti linguistici (es. neologismi, slittamenti di significato).
  - Per monitorare il POS nella misurazione della similarità di significato.

Quanto è difficile il POS tagging in inglese?

- Circa il 15% dei tipi di parole (*word types*) è ambiguo.
  - Di conseguenza, l'85% non è ambiguo (es. *Janet* è sempre PROPN, *hesitantly* è sempre ADV).
- Tuttavia, quel 15% di parole ambigue tende a essere molto comune.
- Infatti, circa il 60% dei *token* in un testo è ambiguo.
- Ad esempio, *back*

  earnings growth took a back/ADJ seat
  a small building in the back/NOUN
  a clear majority of senators back/VERB the bill
  enable the country to buy back/PART debt
  I was twenty-one back/ADV then

Performance del POS tagging in inglese

- L'accuratezza dei tag (*tag accuracy*) è di circa il 97% , un valore stabile da oltre 10 anni.
  - Modelli come HMM, CRF e BERT hanno performance simili.
  - L'accuratezza umana è più o meno la stessa.

- Tuttavia, la *baseline* è già del 92%!
  - La *baseline* è la performance del metodo più semplice possibile.
  - Una *baseline* importante è la "most frequent class baseline":
    - Taggare ogni parola con il suo tag più frequente.
    - Taggare le parole sconosciute come nomi.

- Il task è in parte facile perché molte parole non sono ambigue.

Fonti di informazione per il POS tagging

Janet will back the bill

AUX/NOUN/VERB? NOUN/VERB?

- Probabilità a priori della coppia parola/tag:
  - "will" è solitamente un AUX.
- Identità delle parole vicine:
  - "the" suggerisce che la parola successiva non sia un verbo.
- Morfologia e forma delle parole
  - Prefissi — unable: un- → ADJ
  - Suffissi — importantly: -ly → ADJ
  - Maiuscole — Janet: CAP → PROPN

Named Entity Recognition (NER)

- Una *Named Entity* è, nella sua accezione principale, qualsiasi cosa a cui ci si possa riferire con un nome proprio.

- I 4 tag più comuni sono:
  - PER (Persona): “Marie Curie”
  - LOC (Luogo): “New York City”
  - ORG (Organizzazione): “Stanford University”
  - GPE (Entità Geo-politica): "Boulder, Colorado"

- Spesso si tratta di espressioni multi-parola.
- Il termine viene esteso anche a elementi che non sono entità, come date, orari e prezzi.

Named Entity tagging

- Il task del *Named Entity Recognition* (NER) consiste nel:

1. Trovare le porzioni di testo (*span*) che costituiscono nomi propri.
2. Taggare il tipo di entità.

NER output

Citing high fuel prices, [ORG **United Airlines**] said [TIME **Friday**] it has increased fares by [MONEY **$6**] per round trip on flights to some cities also served by lower-cost carriers. [ORG **American Airlines**], a unit of [ORG **AMR Corp.**], immediately matched the move, spokesman [PER **Tim Wagner**] said. [ORG **United**], a unit of [ORG **UAL Corp.**], said the increase took effect [TIME **Thursday**] and applies to most routes where it competes against discount carriers, such as [LOC **Chicago**] to [LOC **Dallas**] and [LOC **Denver**] to [LOC **San Francisco**].

Perché il NER è utile?

- **Sentiment Analysis:** per analizzare il sentiment verso un'azienda o una persona.
- **Question Answering:** per rispondere a domande su una specifica entità.
- **Information Extraction:** per estrarre fatti su entità da un testo.

Perché il NER è difficile?

- **Segmentazione:** a differenza del POS tagging, nel NER bisogna prima identificare e segmentare le entità.
- **Ambiguità del tipo.**

[PER Washington] was born into slavery on the farm of James Burroughs.
[ORG Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [LOC Washington] for what may well be his last state visit.
In June, [GPE Washington] passed a primary seatbelt law.

BIO Tagging

- Come trasformare questo problema in un task di *sequence labeling* (come il POS tagging), con un tag per ogni parola?

- [PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

BIO Tagging

- Si usano i tag **B**, **I**, **O**:
  - **B**: *Begin*, token che inizia uno *span* di entità.
  - **I**: *Inside*, token interno a uno *span*.
  - **O**: *Outside*, token esterno a qualsiasi *span*.

- Numero di tag (con *n* tipi di entità):
  1 tag O + *n* tag B + *n* tag I = 2*n*+1 tag totali.

- *Adesso abbiamo un tag per token!!!*

Words	BIO Label
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

BIO Tagging variants: IO and BIOES

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

Guardiamo un modello funzionante su HuggingFace !!

Standard algorithms for NER/POS tagging

- Algoritmi standard per NER/POS tagging:
- Hidden Markov Models
- *Conditional Random Fields (CRF)*
- Modelli neurali per sequenze (RNNs or Transformers)
- Large Language Models finetuned (come BERT)
- Tutti richiedono un *training set* etichettato a mano e raggiungono performance simili (97% in inglese).
- Sfruttano le fonti di informazione già viste:
- HMMs and CRFs con feature create dall'uomo
- Neural Language Model tramite *representation learning*

HMM

- Si parte dall'*assunzione di Markov* per i bigrammi:

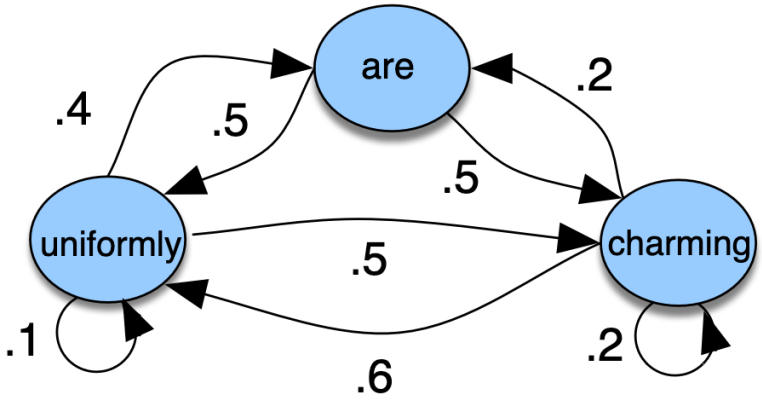
$$P(w_i \mid w_1, w_2, \ldots, w_{i-1}) = P(w_i \mid w_{i-1})$$

- L'assunzione di Markov semplifica un modello che descrive la probabilità che un sistema stocastico si trovi in un certo stato, dopo aver attraversato una serie di stati e cioè una

catena di Markov

HMM

- Catena di Markov



$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{N1} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

HMM

- Un *Hidden Markov Model* usa una catena di Markov per stimare la probabilità di una serie di *eventi nascosti* (gli *stati*, cioè i tag POS) a partire da una serie di *osservazioni* (le *parole*).

HMM

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state q_i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Markov Assumption: $P(q_i|q_1, \ldots, q_{i-1}) = P(q_i|q_{i-1})$

→ *HMM del primo ordine*

Output Independence: $P(o_i|q_1, \ldots q_i, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i|q_i)$

HMM Tagger

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

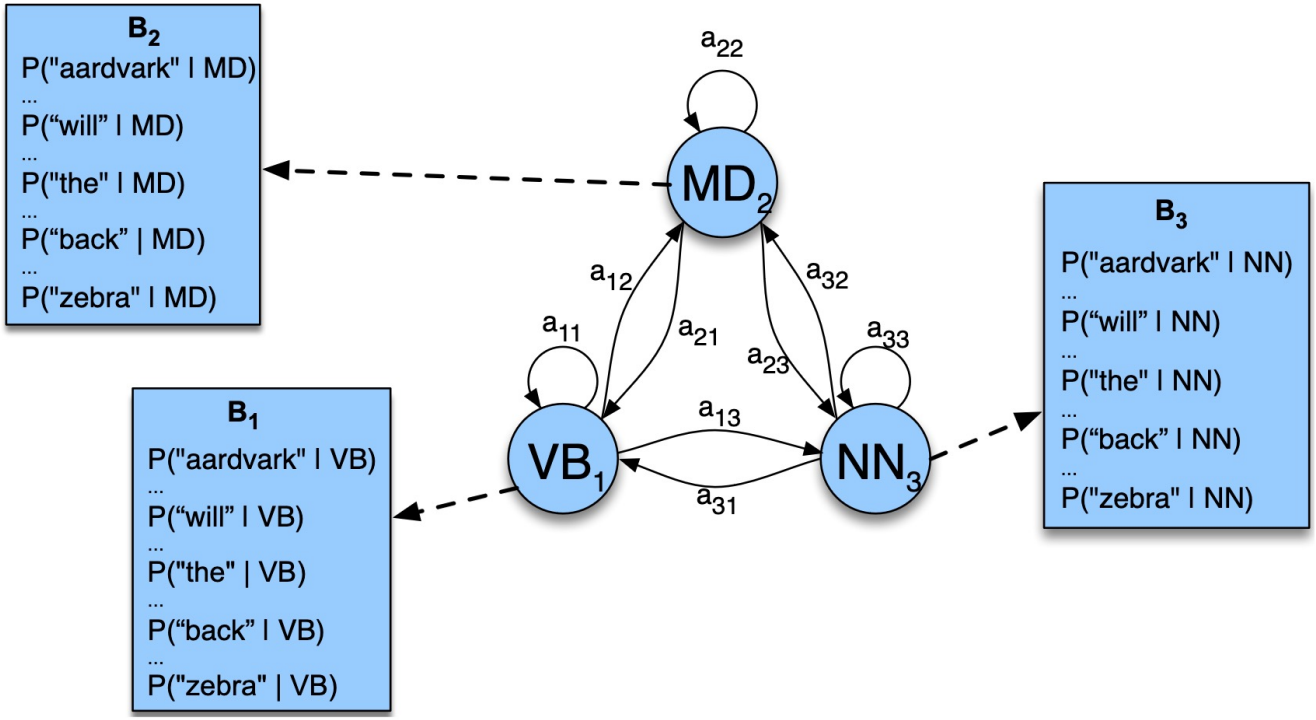
Stima MLE della probabilità di transizione dei Tag (*A*)

$$P(w_i|t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

Stima MLE della probabilità di emissione dei Tag (*B*)

P(*will*|MD): *posto che il tag successivo è MD, quanto è probabile che osserviamo la parola «will»?*

HMM Tagger


B2
P("aardvark" | MD)
...
P("will" | MD)
...
P("the" | MD)
...
P("back" | MD)
...
P("zebra" | MD)
B1
P("aardvark" | VB)
...
P("will" | VB)
...
P("the" | VB)
...
P("back" | VB)
...
P("zebra" | VB)
B3
P("aardvark" | NN)
...
P("will" | NN)
...
P("the" | NN)
...
P("back" | NN)
...
P("zebra" | NN)
MD2
VB1
NN3
a22
a12
a21
a32
a11
a23
a33
a13
a31

HMM Tagger

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n)$$

$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(w_1 \ldots w_n | t_1 \ldots t_n) P(t_1 \ldots t_n)$$

$$P(w_1 \ldots w_n | t_1 \ldots t_n) \approx \prod_{i=1}^{n} P(w_i | t_i)$$

$$P(t_1 \ldots t_n) \approx \prod^{n} P(t_i | t_{i-1})$$

Il problema (una stima MLE)

Applichiamo la re gola di Bayes scartando l'evidenza $P(w_1, \ldots, w_n)$

Indipendenza statistica dell'output

Usiamo i bigrammi

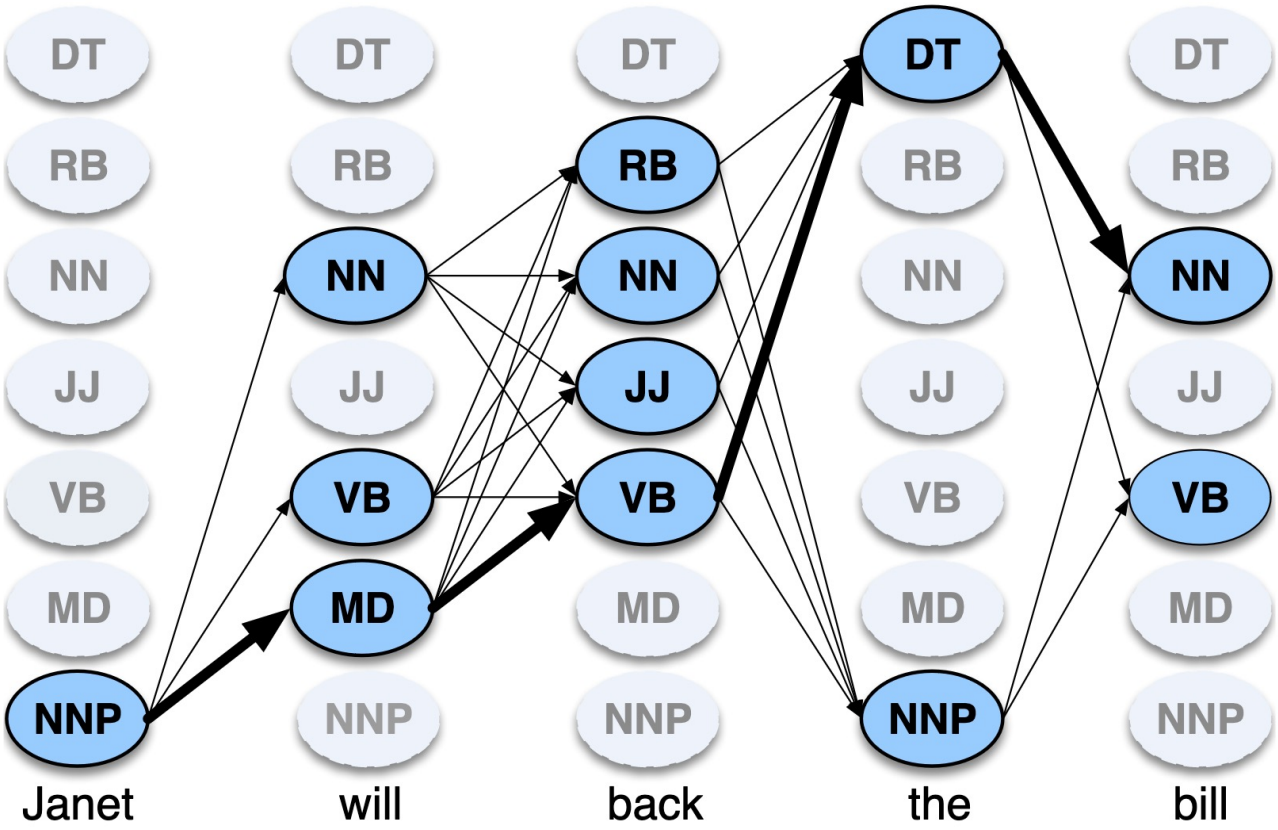
$$\hat{t}_{1:n} = \underset{t_1 \ldots t_n}{\operatorname{argmax}} P(t_1 \ldots t_n | w_1 \ldots w_n) \approx \underset{t_1 \ldots t_n}{\operatorname{argmax}} \prod_{i=1}^{n} \overbrace{P(w_i | t_i)}^{\text{emission}} \overbrace{P(t_i | t_{i-1})}^{\text{transition}}$$

Università degli Studi di Palermo

dipar di ing unipa

CHILAB

HMM Tagger



Algoritmo di Viterbi:

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) P(t_j \mid t_i) P(w_t \mid t_j) \quad 1 \le j \le N, 1 < t \le T$$

Università degli Studi di Palermo

dipartimento di ingegneria unipa

CHILAB

CRF (Conditional Random Fields)

- Sarebbe utile poter considerare *feature* arbitrarie, cosa che gli HMM non gestiscono bene.
    - Parole sconosciute nel POS tagging
    - Nuovi verbi e nomi propri o comuni
    - Regole di morfologia (i.e. *–ed* → VBD oppure VBN)
    - …

CRF

- Sarebbe utile poter considerare *feature* arbitrarie, cosa che gli HMM non gestiscono bene.
  - Gli HMM sono modelli generativi e richiedono il pre-calcolo di molte probabilità.
  - Bisogna ricalcolare le probabilità per l'aggiunta di una sola feature
- I ***modelli discriminativi***, come i CRF, gestiscono meglio vettori di *feature* lunghi.

CRF

- Un CRF impara a predire globalmente la sequenza di tag più probabile $\hat{Y}$ tra tutte le possibili sequenze di tag $\mathcal{Y}$ data una sequenza di input X

$$\hat{Y} = \underset{Y \in \mathcal{Y}}{\operatorname{argmax}} P(Y|X)$$

- Fa uso di una regressione logistica multinomiale (cioè su molte classi)

(Linear chain) CRF

Softmax

Dipende solo da X: non influisce su argmax

$$p(Y|X) = \frac{\exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)}{\sum_{Y' \in \mathcal{Y}} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y')\right)} = \frac{1}{Z(X)} \exp\left(\sum_{k=1}^{K} w_k F_k(X,Y)\right)$$

Feature globale

$$F_k(X,Y) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, X, i)$$

Università degli Studi di Palermo

dipartimento di ingegneria unipa

CHILAB

CRF

- Le *<u>feature locali</u>* dipendono solo dalla coppia di tag (y_i, y_{i-1}) dalla posizione *i* e dalla frase *X*

$$f_k(y_{i-1}, y_i, X, i)$$

$$\mathbb{1}\{x_i = the,\ y_i = \text{DET}\}$$
$$\mathbb{1}\{y_i = \text{PROPN},\ x_{i+1} = Street,\ y_{i-1} = \text{NUM}\}$$
$$\mathbb{1}\{y_i = \text{VERB},\ y_{i-1} = \text{AUX}\}$$

Regole binarie:
1 se la regola vale
0 altrimenti

CRF

- Le *feature* possono essere definite tramite *feature templates*, ovvero specifiche astratte.

$$\langle y_i, x_i \rangle, \langle y_i, y_{i-1} \rangle, \langle y_i, x_{i-1}, x_{i+2} \rangle$$

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

⟨VB, *back*⟩, ⟨VB, MD⟩, ⟨VB, *will*, *bill*⟩

CRF

- Si possono usare anche *feature* morfologiche e di *word-shape*:
  - Prefissi, suffissi.
  - Forma della parola

well-dressed

$$\text{prefix}(x_i) = \texttt{w}$$
$$\text{prefix}(x_i) = \texttt{we}$$
$$\text{suffix}(x_i) = \texttt{ed}$$
$$\text{suffix}(x_i) = \texttt{d}$$
$$\text{word-shape}(x_i) = \texttt{xxxx-xxxxxxx}$$
$$\text{short-word-shape}(x_i) = \texttt{x-x}$$

CRF

- Feature tipiche in un NER
-

identity of w_i, identity of neighboring words
embeddings for w_i, embeddings for neighboring words
part of speech of w_i, part of speech of neighboring words
presence of w_i in a **gazetteer**
w_i contains a particular prefix (from all prefixes of length ≤ 4)
w_i contains a particular suffix (from all suffixes of length ≤ 4)
word shape of w_i, word shape of neighboring words
short word shape of w_i, short word shape of neighboring words
gazetteer features

Una lista di nomi geografici

CRF

Words	POS	Short shape	Gazetteer	BIO Label
Jane	NNP	Xx	0	B-PER
Villanueva	NNP	Xx	1	I-PER
of	IN	x	0	O
United	NNP	Xx	0	B-ORG
Airlines	NNP	Xx	0	I-ORG
Holding	NNP	Xx	0	I-ORG
discussed	VBD	x	0	O
the	DT	x	0	O
Chicago	NNP	Xx	1	B-LOC
route	NN	x	0	O
.	.	.	0	O

Tutte le feature sono binarie

CRF

- **Training:** si usa lo *Stochastic Gradient Descent* con *Cross-Entropy Loss*.
  - Si usa la regolarizzazione

- **Inferenza:** si usa l'Algoritmo di Viterbi, modificato per includere il peso delle *feature* del CRF che vengono *sommate* ad ogni passo

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) + \sum_{k=1}^{K} w_k f_k(y_{t-1}, y_t, X, t) \quad 1 \leq j \leq N, 1 < t \leq T$$