Corpora, parole e token

CORSO DI NATURAL LANGUAGE PROCESSING (ELABORAZIONE DEL LINGUAGGIO NATURALE)

a.a. 2025/2026

Prof. Roberto Pirrone

Corpora

Le parole usate nei modelli di NLP non spuntano dal nulla!

Un testo è prodotto da

- uno o più scrittori specifici,
- in un momento specifico,
- in una varietà specifica,
- di una lingua specifica,
- per una funzione specifica.

I corpora variano lungo le dimensioni come

Lingua: 7097 lingue nel mondo

È importante testare gli algoritmi su più lingue

Ciò che può funzionare per uno potrebbe non funzionare per un altro

I corpora variano lungo le dimensioni come

Varietà, come le varietà inglesi afroamericane

- I post di Twitter di AAE potrebbero includere forme come "*iont" (non lo faccio)*

Genere: newswire, narrativa, articoli scientifici, Wikipedia

Demografia dell'autore: età, sesso, etnia, status socio-economico dello scrittore

Commutazione di codice

I parlanti usano più lingue nella stessa espressione

Questo è molto comune in tutto il mondo

Soprattutto nella lingua parlata e nei generi correlati come gli SMS e i social media

Code switching: spagnolo/inglese

Por primera vez veo a @username actually being hateful! It was beautiful:)

Per la prima volta vedo che @username è davvero odioso! è stato bellissimo:)

Schede tecniche del corpus

Gebru et al (2020), Bender e Friedman (2018)

Motivazione:

- Perché il corpus è stato raccolto?
- Da chi?
- Chi l'ha finanziata?

Situazione: In quale situazione è stato scritto il testo?

Schede tecniche del corpus

Gebru et al (2020), Bender e Friedman (2018)

Processo di raccolta: se si tratta di un sottocampione, come è stato campionato? C'è stato consenso? C’è stato pre-processing?

- *Inoltre processo di annotazione, varietà linguistica, dati demografici, ecc.*

Quante parole in una frase?

```
Fecero un picnic a bordo piscina, poi si
sdraiarono sull'erba e guardarono le stelle.
```

- 15 parole
  - se non contiamo i segni di punteggiatura come parole
- 17 se contiamo la punteggiatura

Quante parole in un enunciato?

- "Mi occupo uh prin- principalmente dell'elaborazione dei dati aziendali"

- Disfluenze
  - Frammenti *prin-*
  - *Pause piene: uh* e *um*

  - Dovremmo considerare queste come parole?

Quante parole in una frase?

```
Fecero un picnic a bordo piscina,
poi si sdraiarono sull'erba e
guardarono le stelle.
```

- **Tipo**: un elemento del vocabolario *V*
  - Il numero di tipi è la dimensione del vocabolario |*V*|
- **Istanza**: un'istanza di quel tipo nel testo corrente.
  - 14 tipi e 15 istanze (se ignoriamo la punteggiatura).

Quante parole in una frase?

- Non tutte le lingue scritte usano gli spazi!!

  Cinesi, giapponesi e tailandesi no!

Come scegliere i token in cinese

Le parole cinesi sono composte da caratteri chiamati "**hanzi" (汉字)**
(o a volte semplicemente "**zi**")

Ognuno di essi rappresenta un'unità di significato chiamata morfema.

Ogni parola ne ha in media 2,4.

Ma decidere cosa conta come parola è complesso e non è concordato.

Come scegliere i token in cinese?

•姚明进入总决赛 "Yao Ming raggiunge la finale"
•Yáo míng jìn rù zǒng jué sai

•3 parole?
•姚明 进入 总决赛
•YaoMing raggiunge le finali

•5 parole?
•姚 明 进入 总 决赛
•Yao Ming raggiunge generali le finali

•7 parole?
•姚 明 进 入 总 决 赛
•Yao Ming entra entra nelgenerale decisione gioco

Chinese Treebank

Peking University

Usiamo i caratteri

Università degli Studi di Palermo

dipartimento di ingegneria unipa

CHILAB

Tokenizzazione tra lingue

- Quindi in cinese usiamo i caratteri (zi) come token
  - Ma questo non funziona, ad esempio, per i thailandesi e i giapponesi
  - Queste differenze rendono difficile l'uso delle parole come token
- E c'è un altro motivo per cui non usiamo le parole come token!

Ci sono semplicemente troppe parole!

- Si noti che (all'incirca) più grandi sono i corpora, più parole troviamo!

	Tipi = |V|	Istanze = N
Shakespeare	31 mila	884,000
Corpus Brown	38 mila	1 milione
Conversazioni in centralino	20 mila	2,4 milioni
COCA	2 milioni	440 milioni
Google N-grammi	13+ milioni	1 trilione

Ci sono semplicemente troppe parole!

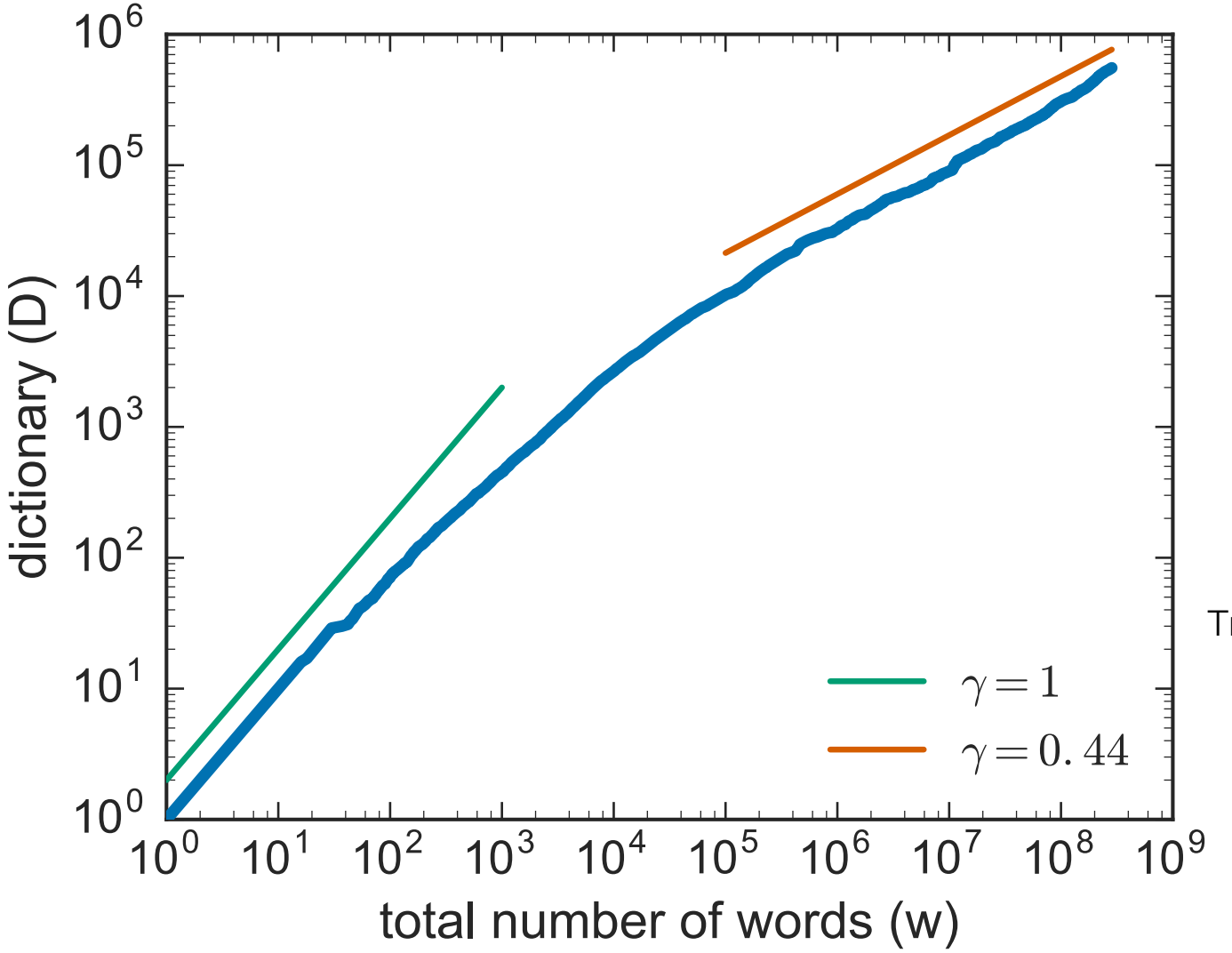
N = numero di istanze

$|V|$ = numero di tipi nel vocabolario V

Legge degli heap = Legge di Herdan

$$|V| = kN^{\beta}$$ ← Circa 0,5

La dimensione del vocabolario di un testo aumenta con la radice quadrata della sua dimensione in parole



Tria, Loreto, Servedio, 2018

Perché troppe parole sono un problema?

- Non importa quanto sia grande il nostro vocabolario
- Ci saranno sempre parole che ci siamo persi!
- Avremo sempre parole sconosciute!

Parole e sottoparole

A causa di questi problemi:

- Molte lingue non hanno parole ortografiche
- Definire le parole post-hoc è impegnativo
- Il numero di parole cresce senza limiti

I sistemi di NLP in genere non utilizzano parole, ma unità più piccole chiamate **sottoparole o subword**

Le parole hanno parti

Morfema: un'unità minima portatrice di significato in una lingua.

`Volpe`: un morfema

`gatti`: due morfemi `gatto` e `-i`

Morfologia: lo studio dei morfemi

Tipi di morfemi

radice: morfema centrale della parola
- fornire il significato principale
affisso: aggiunta di ulteriori significati

lavorato
    radice *lavorare*
    affisso *-ato*
temporalmente
    radice *temporale*
    affisso *-mente*

Tipi di affissi

- **Morfemi flessionali**
  - Morfemi grammaticali
  - spesso ruolo sintattico come l'accordo
  - passato sui verbi
  - plurale dove sostantivi
- **Morfemi derivazionali**
  - più idiosincratico nel significato
  - cambia spesso classe grammaticale

Clitici

- Un morfema che agisce sintatticamente come una parola ma:
  - è ridotto nella forma
  - ed è attaccato a un'altra parola

- Ti voglio
- Ci capiamo

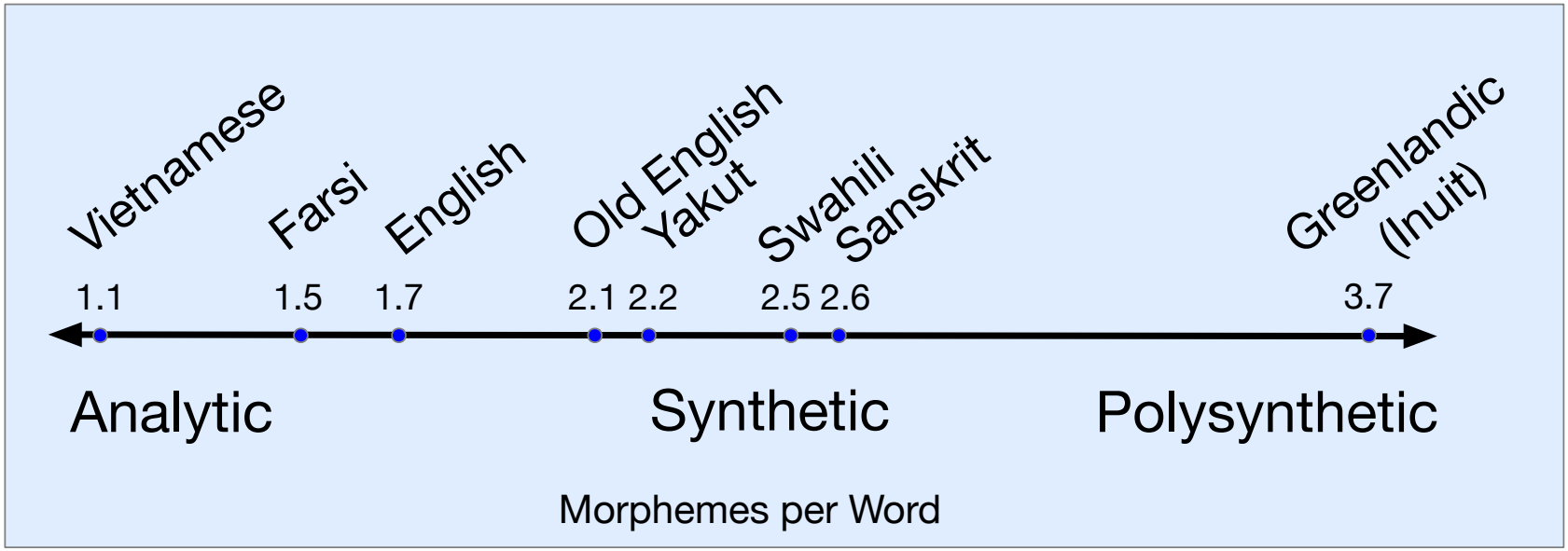
Tipologia morfologica

- Dimensioni lungo le quali variano le lingue

- Due sono salienti per la tokenizzazione:

  1. Numero di morfemi per parola
  2. Quanto è facile segmentare i morfemi

Scala di Joseph Greenberg (1960)



Approccio standard alla tokenizzazione in NLP

Invece di

- spazi bianchi / parole ortografiche
  - Molte lingue non li hanno
  - Il numero di parole cresce senza limiti
- Caratteri Unicode
  - Troppo piccoli come token per molti scopi
- Morfemi
  - Molto difficile da definire

Usiamo i dati per capire come tokenizzare.

Perché tokenizzare?

- L'utilizzo di una serie deterministica di token significa che i sistemi possono essere confrontati equamente

- Elimina il problema delle parole sconosciute

Tokenizzazione delle subword

- Due algoritmi più comuni:
  - *Byte-Pair Encoding (BPE)* (Sennrich et al., 2016)
  - *Unigram language modeling tokenization* (Kudo, 2018) (a volte chiamata in modo confuso "SentencePiece" dalla libreria in cui si trova)

Tokenizzazione delle subword

- Tutti hanno 2 parti:
  - Un *token learner* che prende un corpus di formazione grezzo e induce un vocabolario (un insieme di token).
  - *Un segmentatore* di token che prende una frase di test non elaborata e la tokenizza in base a quel vocabolario

Byte Pair Encoding (BPE) token learner

Unisce in modo iterativo i token adiacenti frequenti per creare token più lunghi.

Ripetere:

1. Scegli la coppia vicina più frequente (ad es. 'A', 'B')
2. Aggiungi un nuovo simbolo fuso ('AB') al vocabolario
3. Sostituisci ogni 'A' 'B' nel corpus con 'AB'.

- Fino a *k* fusioni

Vocabolario

[A, B, C, D, E]

[A, B, C, D, E, AB]

[A, B, C, D, E, AB, CAB]

Corpus

A B D C A B E C A B

AB D C AB E C AB

DA D CAB E CAB

Algoritmo BPE

function BYTE-PAIR ENCODING(strings C, number of merges k) **returns** vocab V

$V \leftarrow$ all unique characters in C # initial set of tokens is characters
for $i = 1$ **to** k **do** # merge tokens til k times
  $t_L, t_R \leftarrow$ Most frequent pair of adjacent tokens in C
  $t_{NEW} \leftarrow t_L + t_R$ # make new token by concatenating
  $V \leftarrow V + t_{NEW}$ # update the vocabulary
  Replace each occurrence of t_L, t_R in C with t_{NEW} # and update the corpus
return V

Algoritmo BPE

- Generalmente viene eseguito *all'interno delle* parole
- Non esegue fusioni attraverso i confini delle parole
  - Dapprima si separa il corpus con gli spazi bianchi
  - Questo fornisce una serie di stringhe iniziali, con gli spazi bianchi attaccati davanti
  - I conteggi provengono dal corpus, ma possono essere fusi solo all'interno di stringhe

Algoritmo BPE

- La maggior parte degli algoritmi di tipo subword viene eseguita all'interno di token separati da spazi.
- Comunemente si aggiunge prima uno speciale simbolo di fine parola '__' prima dello spazio nel corpus di addestramento
- Quindi, separa in lettere.

Esempio di BPE

Corpus originale:

`set␣new␣new␣renew␣reset␣renew`

Metti il token spazio all'inizio delle parole

corpus

2	␣ n e w
2	␣ r e n e w
1	s e t
1	␣ r e s e t

vocabulary

␣, e, n, r, s, t, w

Esempio di BPE

corpus

2 ␣ n e w
2 ␣ r e n e w
1 s e t
1 ␣ r e s e t

vocabulary

␣, e, n, r, s, t, w

Unisci n e → ne (conteggio 4 = 2 new + 2 renew)

corpus

2 ␣ ne w
2 ␣ r e ne w
1 s e t
1 ␣ r e s e t

vocabulary

␣, e, n, r, s, t, w, ne

Esempio di BPE

corpus	**vocabulary**
2 ␣ ne w	␣, e, n, r, s, t, w, ne
2 ␣ r e ne w	
1 s e t	
1 ␣ r e s e t	

Unisci ne w → new (conteggio 4)

corpus	**vocabulary**
2 ␣ new	␣, e, n, r, s, t, w, ne, new
2 ␣ r e new	
1 s e t	
1 ␣ r e s e t	

Esempio di BPE

corpus	**vocabulary**
2 ␣ new	␣, e, n, r, s, t, w, ne, new
2 ␣ r e new	
1 s e t	
1 ␣ r e s e t	

Unisci ␣ r → ␣r (conteggio 4) e ␣r e → ␣re (conteggio 3)

corpus	**vocabulary**
2 ␣ new	␣, e, n, r, s, t, w, ne, new, ␣r, ␣re
2 ␣re new	
1 s e t	
1 ␣re s e t	

Il sistema ha appreso il prefisso re- !

Esempio di BPE

Le prossime fusioni sono:

merge	current vocabulary
(␣, new)	␣, e, n, r, s, t, w, ne, new, ␣r, ␣re, ␣new
(␣re, new)	␣, e, n, r, s, t, w, ne, new, ␣r, ␣re, ␣new, ␣renew
(s, e)	␣, e, n, r, s, t, w, ne, new, ␣r, ␣re, ␣new, ␣renew, se
(se, t)	␣, e, n, r, s, t, w, ne, new, ␣r, ␣re, ␣new, ␣renew, se, set

Algoritmo **di codifica** BPE

Tokenizzare una frase di test significa eseguire ogni fusione appresa dai dati di addestramento:

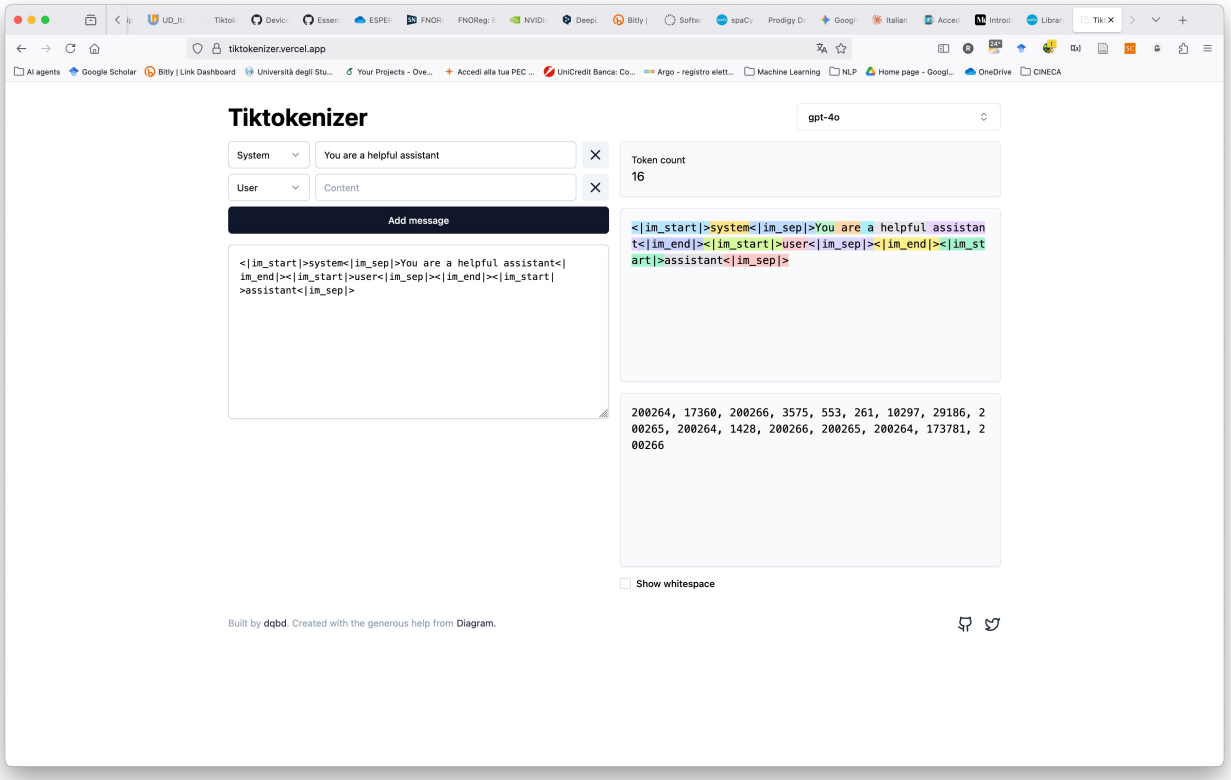
- Con approccio greedy, nell'ordine in cui le abbiamo apprese
- Le frequenze di test non hanno alcun ruolo

1. Segmenta ogni parola di prova in caratteri
2. Esegui le regole apprese: (1) unisci ogni n e → ne, (2) unisci ne w → new, (3) ␣r, (4) ␣re ecc.

Risultato:

- Ricrea le parole del training set
- Ma impara anche sottoparole come ␣re che potrebbero apparire in nuove parole come rearrange

BPE all’opera nei LLM



Università degli Studi di Palermo

dipartimento di ingegneria unipa

CHILAB