# Linguistica computazionale

(A/A 2022-23)

*Prova in itinere del 8 novembre 2022*

**Testo A**
**(studenti la cui PENULTIMA cifra del numero di matricola è compresa tra 0 e 4)**

Cognome e nome:

Matricola:

Istruzioni per la consegna:
- ridenominare il file del compito con *CognomeNomeMatricola* (es. RossiMario001293.doc)
- inviare il file a: alessandro.lenci@unipi.it

---

1. Illustrate la Legge di Zipf (riportando anche la formula matematica) e le sue conseguenze per l'analisi computazionale del linguaggio (PUNTI: 4) (**la risposta deve essere lunga min. 5, max 10 righe**)

2. Spiegate la differenza tra ASCII, Unicode e Iso-Latin1 (PUNTI: 4) (**la risposta deve essere lunga min. 5, max 10 righe**)

3. Spiegate cosa sono le concordanze di una parola (PUNTI: 4) (**la risposta deve essere lunga min. 5, max 10 righe**)

4. In un corpus lungo 15.000 token, il bigramma <*centro, sociale* > ricorre 80 volte. a). Costruite la tabella di contingenza del bigramma e calcolate la sua frequenza attesa; b.) calcolate la mutua informazione del bigramma sapendo che P(*sociale*) = 0.02, e F(*centro*) = 120 (NB: F = frequenza; P = probabilità) (PUNTI: 4).

| | Y=sociale | Y!=sociale |
|---|---|---|
| X=centro | 80 | 40 |
| X!=centro | 220 | 14660 |

Frequenza attesa= (120*300) / 15000 = 2.4

MI(centro, sociale) = log2 ((80 * 15000)/(300*120)) = 5.058

5. |C| = 3000 |V| = 375

F(lo) = 50, F(un) = 35, F(di) = 65, F(studente) = 30, F(linguistica) = 22, F(legge) = 10, F(libro) = 14, F(interessante) = 6

F(<lo, studente>) = 15, F(<studente, di>) = 25, F(<linguistica, legge>) = 5, F(<legge, un >) = 9, F(<un, libro>) = 10

F(<lo, studente, di>) = 7, F(<linguistica, legge, un>) = 4, F(<legge, un, libro>) = 8

Usando un modello markoviano del I ordine e lo add-one smoothing, calcolate la probabilità della frase *Lo studente di linguistica legge un libro interessante*. (PUNTI: 5).

P(*lo, studente, di, linguistica, legge, un, libro, interessante)* = P(lo)*P(studente|lo)*P(di|studente)*P(linguistica|di)*P(legge|linguistica)*P(un|legge)*P(libro|un) * P(interessante|libro) =

(51/(3000+375)) * (16/(50+375)) * (26/(30+375)) * (1/(65+375)) * (6/22+375) * (10/(10+375)) * (11/(35+375)) * (1/(14+375)) =

(51/3375) * (16/425) * (26/405) * (1/440) * (6/397) * (10/385) * (11/410) * (1/389) =

0.015 * 0.038 * 0.064* 0.002 * 0.015 * 0.025 * 0.027 * 0.002 = 1.47744e-15

6.

In una caverna sotto terra viveva uno hobbit. Non era una caverna brutta, sporca, umida, piena di resti di vermi e di trasudo di fetido, e neanche una caverna arida, spoglia, sabbiosa, con dentro niente per sedersi o da mangiare: era una caverna hobbit, cioè comodissima.

Tokenizzate questo testo (normalizzando le maiuscole e includendo la punteggiatura) mettendo un token per riga; calcolate la frequenza cumulata relativa di parole token, calcolate la frequenza relativa della parola *di*, l'indice di ricchezza lessicale e costruite lo spettro di frequenza (PUNTI: 5).

in
una
caverna
sotto
terra
viveva
uno
hobbit
.
Non
era
una
caverna
brutta
,
sporca
,
umida
,
piena
di
resti
di
vermi
e
di
trasudo
di
fetido
,
e
neanche

una
caverna
arida
,
spoglia
,
sabbiosa
,
con
dentro
niente
per
sedersi
o
da
mangiare
:
era
una
caverna
hobbit
,
cioè
comodissima
.

TTR = 37/57 = 0.649

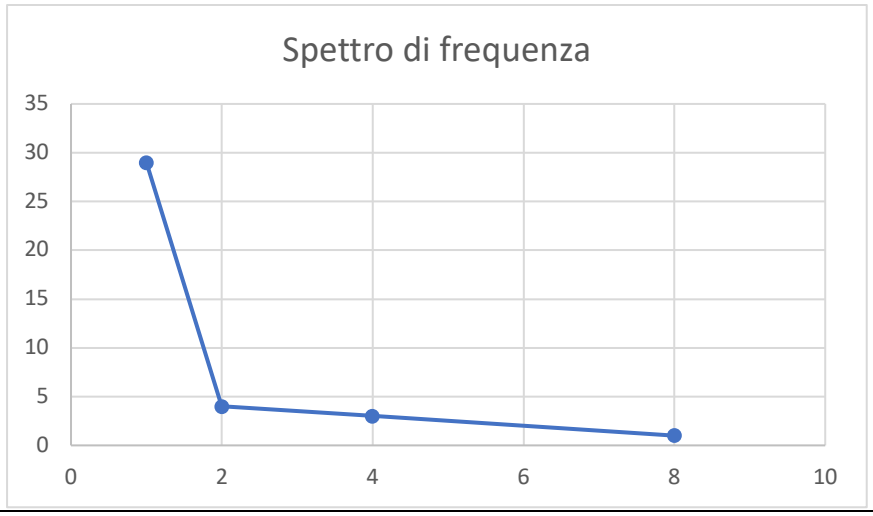
Frequenza relativa di *di*: 4/57 = 0.07

Classi di Frequenza:

| | |
|---|---|
| V1 | 29 |
| V2 | 4 |
| V4 | 3 |
| V8 | 1 |

Frequenza cumulata relativa di parole token:

| | |
|---|---|
| (V1*1)/57 | = 0.50877193 |
| ((V1*1)+(V2*2))/57 | = 0.649122807 |
| ((V1*1)+(V2*2)+(V4*3))/57 | = 0.859649123 |
| ((V1*1)+(V2*2)+(V4*3)+(V8*8))/57 | = 1 |




7. Scrivete le espressioni regolari corrispondenti ai seguenti pattern (PUNTI: 4):

1) sostituite la consonante iniziale di una parola con *a*, soltanto se la parola non termina per *e*, ed è lunga almeno 15 caratteri;

```
s/\b([cdfghklmpqrstvz])(\w{13,}[^e])\b/a\2/
```

2) inserite la stringa *ro* dopo la prima lettera delle parole che compaiono all'inizio di una riga di testo e terminano per *da* o *li*

```
s/^(\w)(\w*)(da|li)\b/\1ro\2\3/
```

3) trovate le parole formate da un numero dispari di caratteri, che terminano per *r* oppure *l* e compaiono alla fine di una riga di testo;

```
/\b(\w\w)*[rl]$/
```