Linguistica computazionale

(A/A 2024-25)

Prima prova in itinere del 13 novembre 2024

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. Spiegate cosa è la *maximum likelihood estimation* della probabilità e quali sono i suoi problemi (PUNTI: 4) (**la risposta deve essere lunga min. 5, max 10 righe**)

2. Illustrare le dinamiche principali del vocabolario di un testo (PUNTI: 4) (**la risposta deve essere lunga min. 5, max 10 righe**)

3. Spiegate la differenza tra codifica di livello 0 e di alto livello (PUNTI: 4) (**la risposta deve essere lunga min. 5, max 10 righe**)

4. In un corpus lungo 15.000 token, il bigramma <*andare*, *forte*> ricorre 200 volte. a). Costruite la tabella di contingenza del bigramma; b.) calcolate la frequenza attesa del bigramma e la sua Mutual Information, sapendo che P(*andare*) = 0.04, e F(*forte*) = 380 (NB: F = frequenza; P = probabilità) (PUNTI: 4).

	forte	!forte	
andare	200	400	600
!andare	180	14220	14400
	380	14620	15000

O<andare,forte> = 200
E<andare,forte> = (f(andare)*f(forte))/N = (380*600)/15000 = 15.2

MI = log2(O<andare,forte>/E<andare,forte>) = 3,718

5. Assumete di avere un corpus di addestramento così strutturato:

|C| = 2000 |V| = 420

F(il) = 60, F(la) = 52, F(gatto) = 25, F(palla) = 15, F(rincorre) = 10, F(rossa) = 5, F(bianco) = 7

F(<la, palla>) = 13, F(<rincorre, la>) = 7, F(<bianco, rincorre>) = 3, F(<palla, rossa>) = 4, F(<gatto, bianco>) = 2

F(<rincorre, il, gatto>) = 4, F(<bianco, rincorre, il >) = 2, F(<gatto, bianco, rincorre>) = 1, F(<rincorre, la, palla>) = 3, F(<la, palla, rossa>) = 2, F(<bianco, rincorre, la>) = 1

Usando un modello markoviano del II ordine e lo add-one smoothing, calcolate la probabilità della frase *Il gatto bianco rincorre la palla rossa*. (PUNTI: 5).

P(il, gatto, bianco, rincorre, la, palla, rossa) = P(il) * P(gatto|il) * P(bianco|il, gatto) * P(rincorre|gatto, bianco) * P(la|bianco, rincorre) * P(palla|rincorre, la) * P(rossa|la, palla) =

(60+1)/(2000+420) * (0+1)/(60+420) * (0+1)/(0+420) * (1+1)/(2+420) * (1+1)/(3+420) * (3+1)/(7+420) * (2+1)/(13+420) =

61/2420 * 1/480 * 1/420 * 2/422 * 2/423 * 4/427 * 3/433 =

0.025 * 0.002 * 0.002 * 0.005 * 0.005 * 0.009 * 0.007 = 1.575e-16

6.

Ogni individuo ha diritto alla vita, alla libertà ed alla sicurezza della propria persona. Nessun individuo potrà essere tenuto in stato di schiavitù o di servitù; la schiavitù e la tratta degli schiavi saranno proibite sotto qualsiasi forma.

a.) Tokenizzate questo testo (includete la punteggiatura e normalizzate le maiuscole) mettendo un token per riga; b.) calcolate la frequenza cumulata relativa di parole token, la frequenza media di parola, e costruite lo spettro di frequenza (PUNTI: 5).

ogni
individuo
ha
diritto
alla
vita
,
alla
libertà
ed
alla
sicurezza
della
propria
persona
.
nessun
individuo
potrà
essere
tenuto
in
stato
di
schiavitù
o
di
servitù
;
la
schiavitù
e

la
tratta
degli
schiavi
saranno
proibite
sotto
qualsiasi
forma
.

Tokens = 42
Types = 35

Per calcolare la frequenza cumulata relativa di parole token devo prima calcolare la distribuzione di frequenza:

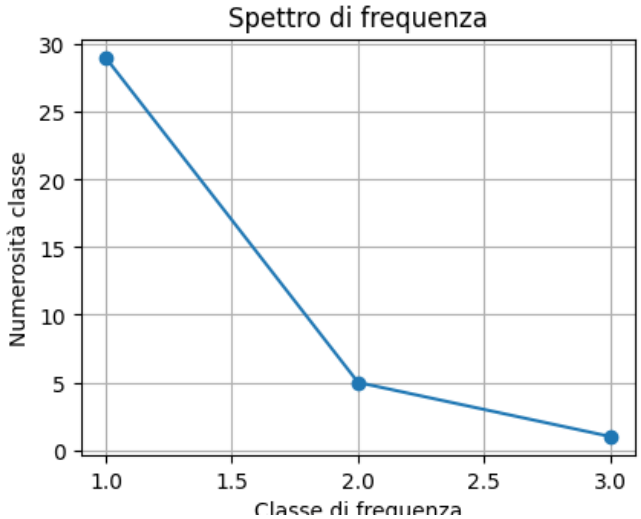
Distribuzione di Frequenza:

Classe	Numerosità
1	29
2	5
3	1

Frequenza cumulata relativa di parole token:
V1 = (1*29)/42 = 0.690
V2 = V1 + (2*5)/42 = 0.929
V3 = V2 + (3*1)/42 = 1

La frequenza media di parola si calcola come il numero di token fratto il numero di type.
$f(|C|) = |C|/|V_C| = 42/35 = 1.2$

Utilizzo la distribuzione di frequenza per rappresentare lo spettro di frequenza.



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

1) togliete la sillaba *la* quando questa appare in fondo a parole che iniziano per *ato* o *ire* e sono lunghe non più di 10 caratteri;

```
s/\b(ato|ire)(\w{0,5})la\b/\1\2/
```

2) trovate le parole che iniziano e terminano per la stessa coppia consonante-vocale (es. *la, pa,* ecc.) e si trovano all'inizio di una riga di testo;

```
/^([qwrtypsdfghjklzxcvbnm][aeiou])\w*\1\b/
```

3) trovate tutti i numeri formati da sequenze identiche di un numero dispari e un numero pari (es. 3434, 787878, ecc.) e che si trovano alla fine di una riga di testo;

```
/\b([13579][02468])\1*$/
```