NLP - LEZIONE 07

DEL 24/10/2019

In questa lezione vediamo dei semplici metodi per effettuare il PART-OF-SPEECH TAGGING.

POS TAGGING (CONT.)

Date n parole $w_1 w_2 \dots w_n$ vogliamo assegnare ad ogni parola una ed una sola tag $T_1 T_2, \dots, T_n$.

Notiamo che se vedo la parola "la", a questa possiamo associare i TAG: NOUN, PRON, DET, ADV. Se però dopo "la" c'è un NOME, allora sono quasi sicuro che il "la" sarà un ARTICOLO.

Una prima idea è quella di utilizzare un DIZIONARIO V che ci dice per ogni parola quali sono i possibili TAGS che possiamo associare a quella parola.

DIZIONARIO
- Insieme ⇒ tags EQUIPROBABILI
- Liste ⇒ tags ordinati in base alla PROBABILITÀ

Rule Based Pos Tagging ~ (13:00 min)

Cominciamo definendo le coppie di tags validi

- Articolo Nome
- Nome Verbo
- Ausiliare Verbo
- Nome Aggettivo

Continuando con questo approccio però dovremmo costruire l'albero ottenendo una grammatica che non abbiamo.

Quello che vogliamo è un metodo che ci permetta di ridurre il # di tags per ogni parola. Se Pos-Tagging è stato spesso utilizzato per ridurre lo spazio di ricerca per lo step successivo del parsing.

Un primo metodo del genere è stato introdotto da Brill e funziona come segue:

1) Associo ad ogni parola il tag con la maggiore probability.

2) Scorro ogni parola adiacente e se trovo coppie di parole "strane" le TRASFORMO utilizzando delle regole.

L'idea è quindi quella di utilizzare dei CONTROLLORI DI STRANEZZE che trasformano i tags associati alle parole. Dato che i controllori possono INFLUENZARSI in modo indiretto, l'ORDINE in cui applico questi controllori è molto importante. Una volta che ho applicato tutti i controllori ritorno i TAGS assegnati.

Il metodo descritto prima è detto TRANSFORMATION BASED POS-TAGGING. Gli aspetti critici di questo metodo sono i seguenti:

i) Definizione delle REGOLE DI TRASFORMAZIONE.

ii) Definizione dell'ORDINE in cui applichiamo i vari controllori.

In particolare vorremo accoppiare a questo metodo un modulo che APPRENDE l'ordine migliore in cui applicare i controllori.

HIDDEN MARKOV MODEL POS TAGGING ~ (52:00 min)

Date la frase $w_1, \dots, w_n$ vogliamo trovare dei tag che MASSIMIZZANO la probabilità

$$P(t_1, t_2, \dots, t_n \mid w_1, w_2, \dots, w_n)$$

Ovvero vogliamo trovare $\bar{t}_1, \bar{t}_2, \dots, \bar{t}_n$ t.c.

$$(\bar{t}_1, \dots, \bar{t}_n) = \underset{t_1, \dots, t_n}{\text{ARGMAX}} \left\{ P(t_1, \dots, t_n \mid w_1, \dots, w_n) \right\}$$

Un primo approccio potrebbe essere quello di stimare $P(t_1 \dots t_n \mid w_1 \dots w_n)$ contando quante volte la sequenza di tags $(t_1, \dots, t_n)$ è associata alla frase $w_1 \dots w_n$ nel nostro DATASET. Notiamo però che questo è estremamente difficile, in quanto raramente in un qualsiasi CORPUS la sequenza di tags $t_1 \dots t_n$ è associata alla frase $w_1 \dots w_n$. Devo dunque trasformare la formula in un qualcosa che sia STIMABILE. Per fare questo è possibile utilizzare delle ipotesi di INDIPENDENZA tra i vari tags.

Così facendo otteniamo che

$$P(t_1 \dots t_n \mid w_1 \dots w_n) \approx \prod_{i=1}^{n} P(t_i \mid w_1 \dots w_n)$$

$$\approx \prod_{i=1}^{n} P(t_i \mid w_{i-1}\ w_i\ w_{i+1})$$

Qui notiziamo che i TAGS sono ind. tra loro.

Qui notiziamo che il tag associato alla i-esima parola dipende solo dalle parole ADIACENTI alla i-esima parola.

Anche se adesso è più facile stimare la prob., il nuovo può comunque essere molto ONEROSO, in quanto lavorando con un DIZIONARIO di 30K parole dovremmo comunque stimare $(30K)^3$ distribuzioni di prob., ciascuna associata ad una tripletta di parole.

Notando che ora abbiamo solamente 15 tags, possiamo ricominciare applicando BAYES come segue

$$P(t_1 \dots t_n \mid w_1 \dots w_n) = \frac{P(w_1 \dots w_n \mid t_1 \dots t_n) \cdot P(t_1 \dots t_n)}{P(w_1 \dots w_n)}$$

Dato che viene interessati solamente all'ARGMAX, possiamo tranquillamente non considerare il DENOMINATORE. Vogliamo quindi massimizzare la seguente espressione

$$P(w_1 \cdots w_n \mid t_1 \cdots t_n) \cdot P(t_1 \cdots t_n)$$

Se assumiamo noi che le PAROLE SONO IND., troviamo che

$$P(w_1 \cdots w_n \mid t_1 \cdots t_n) \approx \prod_{i=1}^{n} P(w_i \mid t_1 \cdots t_n)$$

$$\approx \prod_{i=1}^{n} P(w_i \mid t_i)$$

A questo punto notiamo che possiamo stimare $P(w_i \mid t_i)$ tranquillamente contando nel CORPUS quante volte il tag t_i è ASSOCIATO alla parola w_i.

Per quanto riguarda $P(t_1 \cdots t_n)$ abbiamo che

$$\begin{aligned} P(t_1 \cdots t_n) &= P(t_1 \mid t_2 \cdots t_n) \cdot P(t_2 \cdots t_n) \\ &= P(t_1 \mid t_2 \cdots t_n) \cdot P(t_2 \mid t_3 \cdots t_n) \cdots P(t_{n-1} \mid t_n) \cdot P(t_n) \\ &= \prod_{i=1}^{n} P(t_i \mid t_{i+1} \cdots t_n) \cdot P(t_n) \end{aligned}$$

Se noi assumiamo che OGNI TAG DIPENDE O DAL PRECEDENTE O DAL SUCCESSIVO (uno tra i due) troviamo la seguente semplificazione

$$\prod_{i=1}^{n} P(t_i \mid t_{i+1} \cdots t_n) \cdot P(t_n) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-1}) \cdot P(t_n)$$

Ci rimangono quindi i seguenti problemi:

i) Come lavoriamo sull'ARGMAX?

Siamo infatti interessati a dover calcolare l'ARGMAX della seguente espressione

$$\prod_{i=1}^{n} P(w_i \mid t_i) \cdot P(t_i \mid t_{i-1}) \cdot P(t_n)$$

ii) Come stimiamo le probabilità?

Vedere la sezione successiva.

MASSIMA VEROSOMIGLIANZA ~ (1:26:00 min)

Perché per stimare le prob. devo contare il # di casi favorevoli rispetto al # di eventi totali?

Per rispondere a tale domanda introduciamo il concetto di LIKELIHOOD.

$$\text{LIKELIHOOD} := P(\text{OSSERVAZIONE} \mid \text{PARAMETRI } \theta)$$

Il metodo di MASSIMA VEROSIMIGLIANZA ci permette di stimare i PARAMETRI di una distribuzione andando a massimizzare la VEROSIMIGLIANZA.

ESEMPIO: Consideriamo n lanci di una moneta

$X_1, \dots, X_n$ i.i.d. con $X_i \sim \text{Bern}(\pi)$

Volendo applicare il metodo di MASSIMA VEROSOMIGLIANZA per stimare π procediamo come segue

$$P(X_1 \dots X_n \mid \pi) = \pi^{x_1 + \dots + x_n} \cdot (1-\pi)^{n - (x_1 + \dots + x_n)}$$

$\log \downarrow$

$$\log P(X_1 \dots X_n \mid \pi) = (x_1 + \dots + x_n) \cdot \log \pi + (n - \sum_i x_i) \log(1-\pi)$$

$\frac{d}{d\pi} \downarrow$

$$\frac{d}{d\pi} \log P(X_1 \dots X_n \mid \pi) = \frac{\sum_i x_i}{\pi} - \frac{n - \sum_i x_i}{1-\pi}$$

Troviamo quindi

$$\frac{\sum_i x_i}{\pi} - \frac{n - \sum_i x_i}{1-\pi} = 0 \iff \frac{\sum_i x_i - \pi \sum_i x_i - n\cdot\pi + \pi \sum_i x_i}{\pi(1-\pi)} = 0$$

$$\iff \sum_i x_i - n\cdot\pi = 0$$

$$\iff \pi = \frac{\sum_i x_i}{n}$$

Troviamo quindi che per stimare π dobbiamo contare casi favorevoli su casi totali.

Nel nostro caso particolare possiamo quindi stimare le prob. come segue

$$P(t_i \mid t_{i-1}) = \frac{c(t_i, t_{i-1})}{c(t_{i-1})}$$

di volte in cui abbiamo esattamente i tags t_i e t_{i-1} nel CORPUS.

$$P(w_i \mid t_i) = \frac{c(w_i, t_i)}{c(t_i)}$$

In quest'ultimo caso può succedere che $c(w_i, t_i) = 0$ in quanto potrei non aver mai visto la parola w_i nel CORPUS. Per risolvere questo problema esistono varie tecniche, tra cui una che prende il nome di GOOD-TURING FREQUENCY ESTIMATION

LAVORARE SULL'ARGMAX ~ (1:48:00 min)

Vogliamo trovare l'ARGMAX della seguente espressione

$$\prod_{i=1}^{n} P(w_i \mid t_i) \cdot P(t_i \mid t_{i-1}) \cdot P(t_n)$$

Per fare questo iniziamo partire scegliendo il TAG t_1 che massimizza la probabilità e continuare mantenendo di volta in volta al più K SEQUENZE DI TAGS DISTINTE e prendendo la sequenza con la prob. maggiore.

Per fare questo devo introdurre il tag $t_0 :=$ START e devo stimare le prob $P(t \mid \text{START})$.

Questo metodo è chiamato LINK SEARCH. Se poi K = 1 allora otteniamo il MONTE CARLO ALGORITHM.

Notiamo che questo metodo non sempre ci permettere di ottenere il MASSIMO ASSOLUTO. Detto questo è FORSE possibile dimostrare che con ALTA PROBABILITÁ otteniamo un buon MASSIMO.

NOTE FINALI ~ (2:02:00 min)

Breve riferimento ai concetti di ACCURACY, PRECISION e RECALL.