Relazione
Esercitazione 7

False Friends

Introduzione

La seguente esercitazione ha visto la definizione di un algoritmo di individuazione di parole *false friends* ovvero coppie di parole del lessico di lingue diverse che presentano una veste lessicale molto simile, se non identica in alcuni casi, ma che individuano significati completamente differenti.

Per questa esercitazione si è deciso di lavorare su un corpus usato già nelle esercitazioni precedenti ovvero `wikisent2.txt` che contiene circa 8 milioni di frasi prese da un dump di Wikipedia.

Una volta eseguite attività di pre-processing sul corpus, per comprendere se due parole sono o meno false friends si è deciso di utilizzare il modulo `fuzz` dalla libreria `fuzzywuzzy` che utilizza la distanza di Levenshtein per calcolare le differenze tra le sequenze di stringhe.

In seguito, sono state valutate le similarità, tra le coppie di parole individuate precedentemente, attraverso la metrica di `Wu & Palmer` di modo che termini con valore di distanza di Levenshtein alto e similarità `Wu & Palmer` bassa saranno dei buoni candidati per essere dei false friends.

NOTE:

- In questa esercitazione viene trattata una sola lingua (inglese), e la definizione di *false friends* si restringe semplicemente a due termini quasi omonimi che condividono molti caratteri in comune ma che differiscono di molto nel significato.

Struttura del codice ed implementazione

Per lo svolgimento di questa esercitazione l'intero codice è presente nel file **`main.py`** che implementa i seguenti metodi

- `bag_of_words(sentence)`

  metodo si occupa di restituire, data una stringa in input, le *bag of word* mantenendo solo le parole che contengono lettere e rimuovendo le stop words.

- `parse_corpus()`

  metodo che si occupa di parsificare il corpus dell'esercitazione.
  Per ogni riga nel corpus vengono definite le BoW (tramite il metodo prima descritto), successivamente ogni lista del corpus viene accorpata in un'unica lista. Infine, viene ritornata la lista che contiene solo quelle parole che presentano almeno un synset di Wordnet.

- `lex_similar_words(word_list)`

  metodo che prende in input una lista di parole, e si occupa cliclare su quest'ultima identificando in ogni possibile coppia di parole, quelle che presentano una distanza di Levenshtein maggiore di `LEX_SIMILARITY_THRESHOLD` (90)

- `false_friends(pairs_list)`

  metodo che, data una lista di coppie di parole, restituisce una nuova lista contenente solo le coppie di parole che presentano un valore di similarità `Wu & Palmer` minore o uguale a `SEM_SIMILARITY_THRESHOLD` (0.10).

Risultati

Di seguito i risultati ottenuti.

Sono mostrate le prime 15 coppie più rilevanti

word pairs	lex sim	sem sim
blighty - blight	92	0.095
scrum - sacrum	91	0.1
finches - inches	92	0.1
faucet - facet	91	0.09
couch - crouch	91	0.1
cheque - chequer	92	0.1
swinge - swine	91	0.09
burial - urial	91	0.07
doves - droves	91	0.1
begging - beging	92	0.1
spawns - pawns	91	0.09
sirens - sires	91	0.09
ducking - duckling	93	0.09
ormers - formers	92	0.1
healths - heaths	92	0.09