MAADB

Obiettivi

1. Costruire una word cloud delle parole con la frequenza maggiore per ognuna delle emozioni
2. Fare statistiche su hashtag ed emoticon
3. Arricchire il tesauro con le parole più frequenti nei tweet ma non presenti in esso ed eliminare da quest’ultimo quelle con la frequenza bassa (approfondire la scelta del valore soglia).
4. Comparare i risultati tra il nuovo tesauro creato e il vecchio
5. Comparare le performance tra dbms relazionale e nosql
6. Proporre una soluzione di machine learning su sentiment analisys

Linguaggio: Python

DBMS: MongoDB e Oracle (Express Edition).

Per MongoDB 4 nodi da mandare in esecuzione 2 su un pc e 2 sull’altro.

Pipeline

1. Eliminare dalle risorse le parole composte (facilmente riconoscibili dalla presenza del carattere speciale ‘\_’)
2. Conteggiare la presenza delle singole parole in più risorse (più file) relative allo stesso sentimento.
3. Calcolare la percentuale di presenza di ogni parola nelle diverse risorse (i diversi file relativi allo stesso sentimento).
4. Caricare le risorse su database relazionale (parole e frequenze (array con frequenze, una per ogni emozione)).
5. Riconoscere gli hashtag e memorizzarli in qualche modo (con struttura simile a quella delle parole)
6. Riconoscere gli emoji e rendicontarne la presenza nei vari sentimenti/tweet
7. Riconoscere le forme di slang
8. Trattare la punteggiatura.
9. Pulire i tweet dalla presenza delle parole tipiche anonimizzate USERNAME e URL.
10. Eliminare le stop words.
11. Lemmizzare le parole per permettere il match con le risorse lessicali
12. Conteggiare la presenza nei vari tweet delle parole associate a un determinato sentimento.
13. Memorizzare le ‘nuove’ parole trovate (se alla fine del conteggio saranno altamente presenti avremo trovato una nuova parola da aggiungere alle risorse!)