Analisi di Documenti Testuali

Federico Dassereto

Chi Sono

- Federico Dassereto
  - per voi Federico
- Studente di dottorato in Informatica
  - Superiori, Laurea Triennale, Laurea Magistrale, Dottorato
- Mi occupo di basi di dati e ricerca su tabelle
- Liceo Scientifico Tecnologico, Laurea Triennale e Magistrale a Genova

Chi Siete

- Nome e Cognome
- Provenienza
- Cosa vi aspettate dallo stage
  - Conoscente l’Inverted Index?
  - Conoscete tf-idf?
- Esperienza di programmazione

Cosa mi aspetto

- Curiosità!
  - Io scelsi di iscrivermi ad informatica dopo aver fatto lo stage
  - Non diamo voti, nessuno vi giudica
- Collaborazione
  - Ve l’avranno già detto, ma *lavorare in gruppo aumenta esponenzialmente* la qualità del lavoro
- Last but not least, che vi resti qualcosa dell’analisi di documenti!

Documenti

- Sequenza di parole organizzate in frasi
- Questa presentazione è un documento
  - Un libro, una lettera, una email

Intuitivi per gli essere umani, ma per le macchine?

Documenti

Problemi con i Documenti

- “Oggi è una giornata afosa e soleggiata”
- “Sole ma non si respira!”

Nessuna parola in comune, come fa la macchina a cogliere la somiglianza nel significato?

Semantics!

Se sarete abbastanza curiosi ne parleremo.. :)

Stemming e Lemmatization

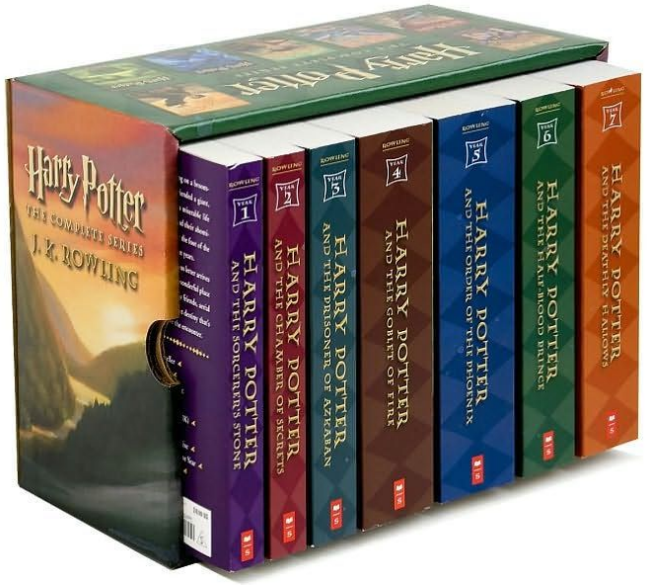
```
run --> run

runner --> runner

running --> run

ran --> ran

runs --> run

easily --> easili

fairly --> fairli
```

```
I             PRON      -PRON-

saw           VERB      see

eighteen      NUM       eighteen

mice          NOUN      mouse

today         NOUN      today

!             PUNCT     !
```

Collezioni di Documenti

“Il signore e la signora Dursley, di Privet Drive numero 4, erano orgogliosi di affermare di essere perfettamente normali, e grazie tante.”

Harry Potter e la Pietra Filosofale



Cercare un Documento in una Collezione

Lasciamo fare alle Macchine..

Quali metodi di ricerca vi vengono in mente?

Cercare Documenti

Immaginiamo di avere moltissimi documenti, diciamo 100.000. In questi documenti ci sono libri, post tratti da social network e articoli di giornale.

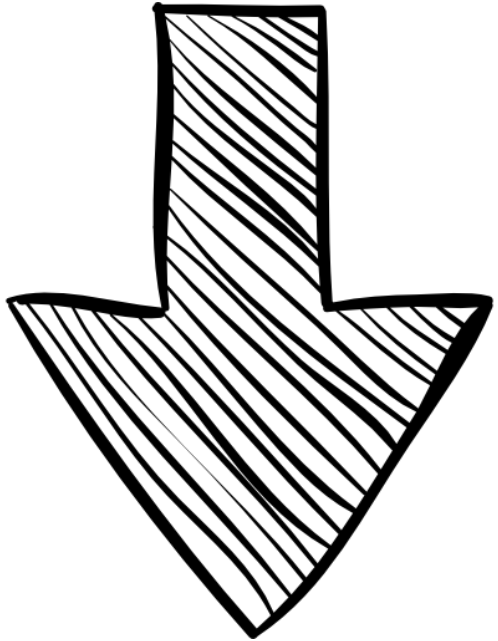
Vorremmo cercare tutti i documenti in cui esiste la sequenza di parole “Harry Potter”. Come fare?

Cercare Documenti

Meta pseudo-codice

1. Per ogni documento
2. Per ogni frase
3. Se “Harry Potter” è nella frase
4. Il documento va restituito

Quanto tempo ci vuole?



Quanto Tempo

Dipende da molte cose

1. Linguaggio utilizzato
2. Numero di Documenti
3. Lunghezza dei Documenti
4. etc

Ma in generale....

Cercare Documenti

Abbiamo raccolti libri, articoli e post nel periodo in cui è uscito al cinema l’ultimo film di Harry Potter

Quasi tutti i post parlano del film, per cui quasi tutti soddisfano i requisiti di ricerca.

Ma hanno tutti la stessa importanza?


Well yes, but actually no

Che cosa sappiamo di Python

Eseguire codice

Implementare funzioni

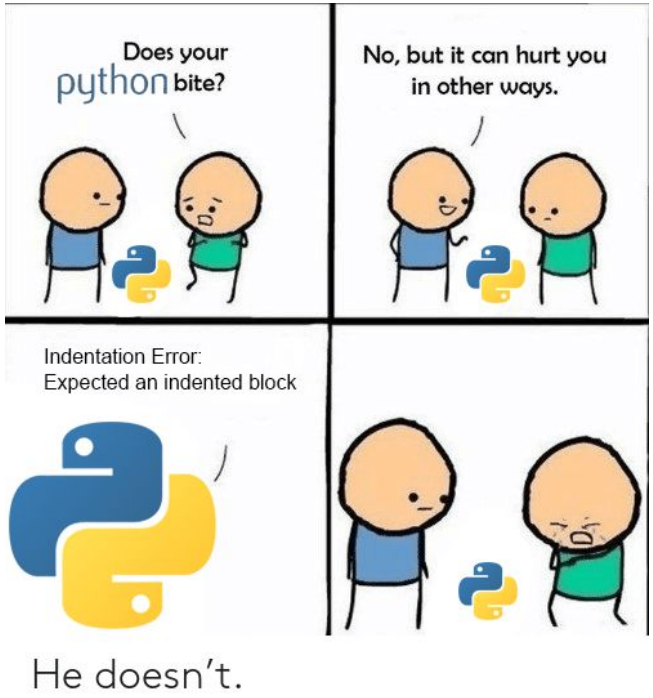
Disegnare grafici

Semplificarci la vita

Prima Intuizione

La frequenza con cui una parola appare all'interno di un documento è rilevante

Let's code a bit!



Tf (Term Frequency)

Misura relativa ad ogni parola rispetto ad un documento

Ma se abbiamo una collezione?

Formalizziamo un po'

t è un termine (parola)

d è un documento

$tf(t, d) = |\{x_i \in d \mid x_i = t\}|$

Normalizzando

$tf(t, d) = |\{x_i \in d \mid x_i = t\}| \,/\, |d|$

Seconda Intuizione

Parole che appaiono più raramente nella collezione sono più significative

Se una parola appare spesso in tutti i documenti non ci da molta informazione sulla collezione

Ri-Formalizziamo

t è un termine (parola)

D è una collezione di documenti d_i

N = |D|

$$idf(t, D) = \log(N/|\{d_i \in D \mid t \in d_i\}|)$$

Perché la funzione logaritmo? Let’s step back on Python!

TF-IDF

Term Frequency - Inverse Document Frequency

- non è una sottrazione!

TF x IDF

Quante volte una parola appare in un documento x quantità di informazione che da' la parola

> "[...] TF-IDF was the most frequently applied weighting scheme. In addition to simple terms, n-grams, topics, and citations were utilized to model users' information needs [...]"

Perchè TF-IDF è un prodotto?

Term Frequency - Intra Document

Inverse Document Frequency - Inter Document

Perchè Tf-Idf è utile [Wikipedia]

A survey conducted in 2015 showed that 83% of text-based recommender systems in digital libraries use TF-IDF.

Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

TF-IDF can be successfully used for stop-words filtering in various subject fields, including text summarization and classification.

No, Google non usa tf-idf

Non fornisce pesi ai collegamenti tra le varie pagine (documenti)

Varianti molto più sofisticate, ormai user-dependent

https://blog.seoprofiler.com/google-dont-use-tf-idf-to-optimize-your-web-pages/

Cosa Dovrete Fare

Implementare questo “indice”!

Nello specifico, vi chiedo di implementare un Inverted Index

Token	Document Id
Harry	1, 2
Potter	1, 2
And	1, 2
The	1, 2
Half	1
Blood	1
Prince	1
Deathly	2
Hallows	2

Inverted index

Step 1 - Inverted Index

Data una collezione di documenti `[[“ciao”, “sono”, “io”], [“io”, “non”, “lo”, “sono”]]` crea una rappresentazione del tipo:

```
ciao [1]
sono [1,2]
io   [1,2]
non  [2]
lo   [2]
```

Step 2 - TF-IDF

Say we have 3 documents

Document 1: Machine learning teaches machine how to learn

Document 2: Machine translation is my favorite subject

Document 3: Term frequency and inverse document frequency is important

Calcolare le frequenze delle parole rispetto ad ogni documento (normalizzate)

Calcolare IDF per ogni parola

Calcolare TF * IDF

Salvare in un indice, che dimensioni avrà?

Cosa Dovrete Fare

L'obiettivo è creare una struttura alla quale si possano sottomettere delle interrogazioni (query), inizialmente composte da una parola.

1) leggere la query (una parola)
2) leggere ogni documento in un vettore **D = [doc0, doc1, etc]**
3) per ogni parola **p** della query
    a) calcolare idf(**p**, **D**)
    b) per ogni documento $\mathbf{d_i}$ in **D**
        i) calcolare tf(**p**, $\mathbf{d_i}$)

Una volta calcolate queste cose, dove le scriverete? **To the notebook!**

Ma andiamo con Ordine

Prima di implementare per davvero, è opportuno imparare alcune cose a livello di linguaggio

1) Come elenco, ed eventualmente apro, tutti i file “.txt” in una cartella?
    a) python packages `os` & `csv`
2) Come posso contare quante volte ogni parola appare in un documento?
    a) python package `collections`
3) Come scrivere una funzione che prenda in input dei valori e ne restituisca altri?
    a) keyword `def`