Information Retrieval

Giuseppe Magazzù

2021 - 2022

Contents

Chapter 1

Definitions

1.1 Document

Un **documento** è solitamente formato da un testo, una struttura, altri media (immagini, suoni, ...) e da dei metadata.

Per **testo** si intende una sequenza di stringhe di caratteri di un alfabeto.
E.g. le sequenze del genoma, formule chimiche e parole del linguaggio naturale.

Un documento può essere composto da

- structured data (tabelle, database, ...)
- semi-structured data (html, xml, ...)

I **metadati** sono dati esterni riguardo al documento. Possono essere classificati in due categorie:

- **metadati descrittivi**: riguardano la creazione del documento (e.g. titolo, autore, data, ...)
- **metadati semantici**: descrivono informazioni contestualmente rilevanti o specifiche del dominio (e.g. ontologie)

1.2 Terms

I termini sono dei descrittori che vengono associati al testo.

1.3 Stop Words

I termini che non sono significativi per la rappresentazione del testo (particelle, articoli, ...).

Chapter 2

Text Processing

Il text processing è una fase necessaria per preparare e pulire il testo.

2.1 Tokenization

La tokenization consiste nell'identificare e separare all'interno di un testo delle unità chiamate token. I token possono essere parole, frasi, simboli o n-grammi. Ogni token è un candidato a essere un termine significativo (index).

> e.g. "Text mining is to identify useful information"
>
> **Tokens**: "Text", "mining", "is", "to", "identify", "useful", "information"

Problemi:

- parole composte ("Hewlett-Packard" $\rightarrow$ "Hewlett", "Packard")
- numeri, date ("Mar. 12, 1991", "12/3/1991", "(800) 234-2333")
- problemi linguistici (parole composte, assenza di spazi, ...)

I token possono essere raggruppati in sequenze contigue di N elementi chiamate N-grammi.

> e.g. "Corpus is the collection of text documents."
>
> **Bigrammi**: "Corpus is", "is the", "the collection", "collection of", "of text", "text documents", "documents ."

La tokenization si può effettuare tramite espressioni regolari o metodi statistici.

2.2 Normalization

Ad un parola possono essere associati diversi token. La normalizzazione consiste nell'ottenere le classi di equivalenza dei token rimuovendo punti, trattini, accenti.

U.S.A. ⇔ USA
anti-aliasing ⇔ antialiasing
résumé ⇔ resume
15/10/2021 ⇔ 15 Ott 2021

Lemmatization

Le parole vengono ridotte alla loro forma base (lemma) tenendo in considerazione l'intero vocabolario della lingua e analizzando la parte del discorso.

e.g. "ladies" ⇒ "lady", "forgotten" ⇒ "forgot"

Stemming

Le parole vengono ridotte a una radice (stem) rimuovendo le flessioni tramite l'eliminazione dei caratteri non necessari.

e.g. "automate(s)", "automation", "automatic" ⇒ "automat"

Case folding

Tutte le parole vengono convertite in lowercase a parte alcune eccezioni.

Thesaurus and Soundex

Un thesaurus (tesauro) è una risorsa linguistica generata manualmente da essere umani in cui è possibile esprimere relazioni tra parole (e.g. gerarchie, sinonimi, ...).

Soundex è un algoritmo fonetico che permette di rappresentare correttamente diverse parole omofone nonostante differenze di ortografia usando delle euristiche fonetiche.

2.3 Stop Words Removal

Le **stop words** sono le parole più frequenti all'interno di un testo che possono essere rimosse senza perdere il significato. Queste parole essendo presenti in più documenti non portano informazioni utili per distinguerli.

Esistono delle liste di **stop words** definite in base alla lingua che possono essere usate per la rimozione.

I web search engine non effettuano la rimozione delle **stop words** perché sono necessarie per alcune ricerche.

Chapter 3

Text Representation

In un sistema di information retrieval i documenti devono essere rappresentati in un formato interno e ordinati per essere indicizzati.

3.1 Bag Of Words

Un modo semplice per rappresentare un testo è una matrice in cui sulle righe ci sono termini estratti dal corpus (vocabolario) e sulle colonne i documenti.

La **Bag Of Words (BOW)** è una rappresentazione del testo che descrive le occorrenze di parole in un documento.

Incidence Matrix: specifica la presenza di un termine in un ogni documento.

Ogni documento può essere rappresentato da un insieme di termini o da un vettore binario.

	Doc1	Doc2	Doc3	Doc4
Term1	1	1	1	0
Term2	0	1	1	1
Term3	0	0	1	0

Rappresentazione di Doc1

$R1 = \{\text{Term1}, \text{Term2}, \text{Term3}\}$
$R1 = \langle 1, 0, 0 \rangle$

Count Matrix: specifica la numero di occorrenze di un termine in ogni documento.

Un documento viene rappresentato da un vettore di occorrenze.

	Doc1	Doc2	Doc3	Doc4
Term1	57	57	71	133
Term2	4	34	17	92
Term3	232	2	10	293

Rappresentazione di Doc1

$R1 = \langle 157, 4, 232 \rangle$

Le rappresentazioni vettoriali non considerano l'ordine delle parole nel testo.

Bag Of Words con N-grammi
Pro: cattura le dipendenze locali e l'ordine
Contro: incrementa la frequenza delle parole

3.2 Zipf's Law

Descrive la frequenza di un evento (parola) in un insieme in base al suo rank.

rank: posizione di un termine nell'ordine decrescente di frequenza dei termini in tutta la collezione.

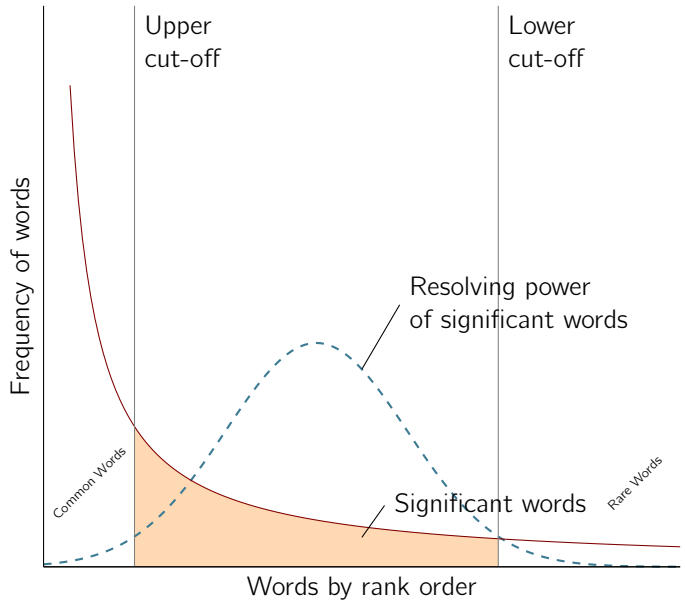
La frequenza di una parola w, f(w) è proporzionale a 1/r(w).

$$f \propto \frac{1}{r} \quad \Rightarrow \quad f \cdot r = k \text{ (costante)}$$

$$P_r = \frac{f}{N} = \frac{A}{r} \qquad \text{probabilità del termine di rank r,} \qquad A = \frac{k}{N} \approx 0.1$$

3.3 Luhn's Analysis

Generalmente termini con frequenza molto alta e molto bassa sono inutili per discriminare i documenti.



L'abilità delle parole di discriminare il contenuto di un documento è massimo nella posizione tra i due livelli di cut-off.

Vogliamo assegnare dei pesi ai termini tenendo conto di questi due fattori:

- corpus-wide: alcuni termini portano più informazione riguardo al documento
- document-wide: non tutti i termini sono ugualmente importanti

Andiamo a definire due frequenze:

- Term Frequency (TF): frequenza di un termine all'interno di un documento
- Inverse Document Frequency (IDF): frequenza di un termine in tutta la collezione

Il peso di un termine deve essere proporzionale a TF e inversamente proporzionale a IDF.

La **term frequency** $tf_{t,d}$ è il numero di occorrenze del termine t nel documento d diviso il numero totale di termini nel documento.

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t_i} f_{t_i,d}}$$

Questa misura può essere normalizzata dividendo per la frequenza massima nel documento d per essere confrontabile tra documenti diversi.

$$ntf_{t,d} = \frac{f_{t,d}}{max_{t_i} f_{t_i,d}}$$

La **inverse document frequency** idf_t è la frazione inversa della frequenza di un termine in un documento in scala logaritmica.

$$idf_t = log(N/df_t), \quad df_t \leq N$$

dove df_t è la **document frequency**, ovvero il numero di documenti che contengono il termine t, e N il numero totale di documenti.

Infine possiamo calcolare la funzione TF-IDF come prodotto di TF e IDF.

$$tf\text{-}idf_{t,d} = (tf_{t,d}/max_{ti} tf_{ti,d}) * log(N/df_t)$$

Questa funzione rappresenta il peso del termine t all'interno di un documento d.

- termine comune in un documento $\rightarrow$ high tf $\rightarrow$ high weight
- termine raro nella collezione $\rightarrow$ high idf $\rightarrow$ high weight

Chapter 4

Text Enrichment

riconoscere una frase: - n-grams - pos tagger - store words position in a index

POS, NER approaches: - rules based - supervised learning

4.1 Part-of-Speech (POS) tagging

Il POS tagging è il processo che marca ogni termine nel documento con un tag che corrisponde a una part-of-speech.

EXAMPLE...

Word Classes

words that somehow behave alike:

- similar transformations
- similar functions in the phrase
- similar contexts

9 traditional word classes of POS (noun, verb, adjective, adverb, preposition, article, interjection, conjunction)

Applicazioni: - Machine Translation - Parsing - Speech Recognition

Tag Ambiguity

Spesso una parola può essere associata a più di un POS, quindi è necessario considerare il contesto.

- The **back** door (adjective)
- Promised to **back** the bill (verb)
-

Rule Based Tagging

- assegno ogni possibile tag a una parole usando un dizionario - scrivo delle regole a mano per rimuove dei tag - lascio un solo tag per parola

consideriamo le frasi con n-grammi

4.2 Named Entity Recognition (NER)

Trovare e classificare nomi in un testo (persone, date, luoghi, organizzazioni).

Chapter 5

Statistical Language Models

Un Statistical LM specifica una distribuzione di probabilità su sequenze di parole.

Applicazioni:

- Machine Translation, P(**high** winds tonite) > P(**large** winds tonite)
- Spell Correction, P(about 15 **minutes**) > P(about 15 **minuets**)
- Speech Recognition, P(I saw a van) > P(eyes awe of an)

5.1 Language Model

L'obiettivo di un Language Model è quello di calcolare la probabilità di una sequenza di parole $P(W) = P(w_1, w_2, ..., w_n)$.

Con una Language Model è possibile calcolare anche la probabilità di una parola data una sequenza $P(w_5|w_1, w_2, w_3, w_4)$.

Possiamo calcolare la probabilità della sequenza W con la chain rule:

$$P(w_1, w_2, ..., w_n) \approx \prod_{i=1}^{n} P(w_i|w_1, ..., w_{i-1})$$

Con la Markov Assumption possiamo ridurre il numero di parole da condizionare

$$P(w_1, w_2, ..., w_n) \approx \prod_{i=1}^{n} P(w_i|w_{i-k}, ..., w_{i-1})$$

Un Language Model è ben formato su un alfabeto Ω se $\sum_{s \in \Omega} P(s) = 1$.

N-grams Language Model

La probabilità di una parola di una sequenza dipende dalle N parole precedenti. Nel caso di uni-grammi la probabilità non dipende da nessuna altra parola.

Sparsity Problems

1. La parola di cui vogliamo calcolare la probabilità non è presente nel corpus. Soluzione: aggiungere una δ alla frequenza di ogni parola (smoothing).
2. La sequenza di cui vogliamo calcolare la probabilità non è presente nel corpus. Soluzione: ridurre la sequenza da condizionare.

I modelli con uni-grammi sono i più usati

- Spesso sono sufficienti per valutare l'argomento
- Con N più grandi ci sono più problemi di sparsity
- Implementazione semplice ed efficiente