Document Semantic Similarity

TIS Project

Alberto Pirovano Francesco Picciotti

Politecnico di Milano

2nd May 2017

Outline

Introduzione

Le tecniche adottate attualmente per trovare la **similitudine semantica tra testi** si basano su tre approcci:

1. NLP Tradizionale
2. Vector Space Model
3. Deep Learning based

Outline

NLP Tradizionale

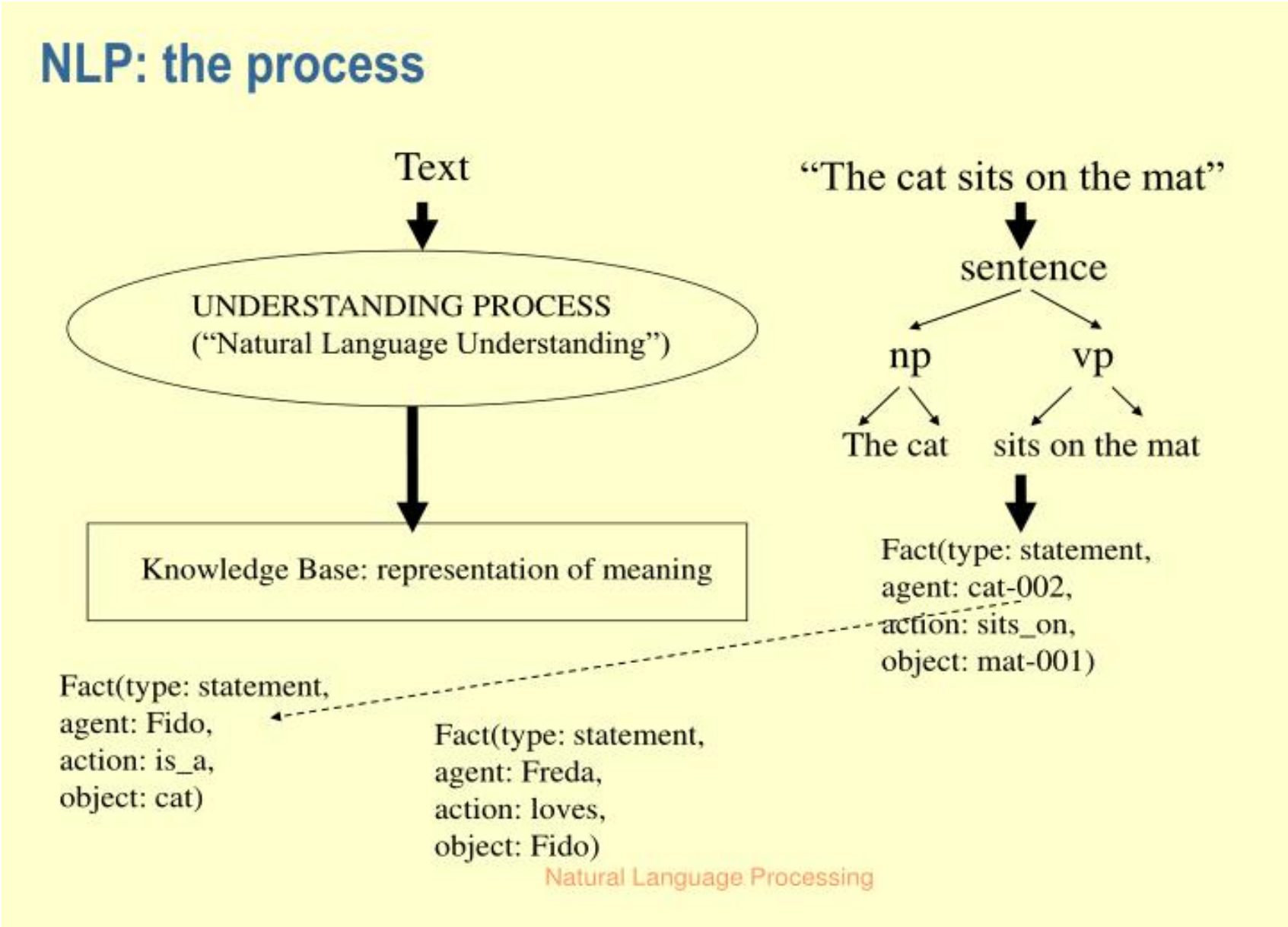
Questo approccio si basa sull'utilizzo delle tradizionali tecniche di **Natural Language Processing** e si costituisce dei seguenti step:

- Cleaning dei dati
- Pos-Tagging
- Stemming o Lemmatisation
- Parsing
- Ontologia

Tuttavia, dato che il nostro lavoro è molto **sensibile** e **dipendente** dalla qualità dei tool utilizzati, abbiamo trovato alcune consistenti **criticità** riguardanti:

- **L'affidabilità** del Pos-Tagger italiano di TreeTagger
- **Reperire** una Ontologia e un parsing toll per la lingua italiana

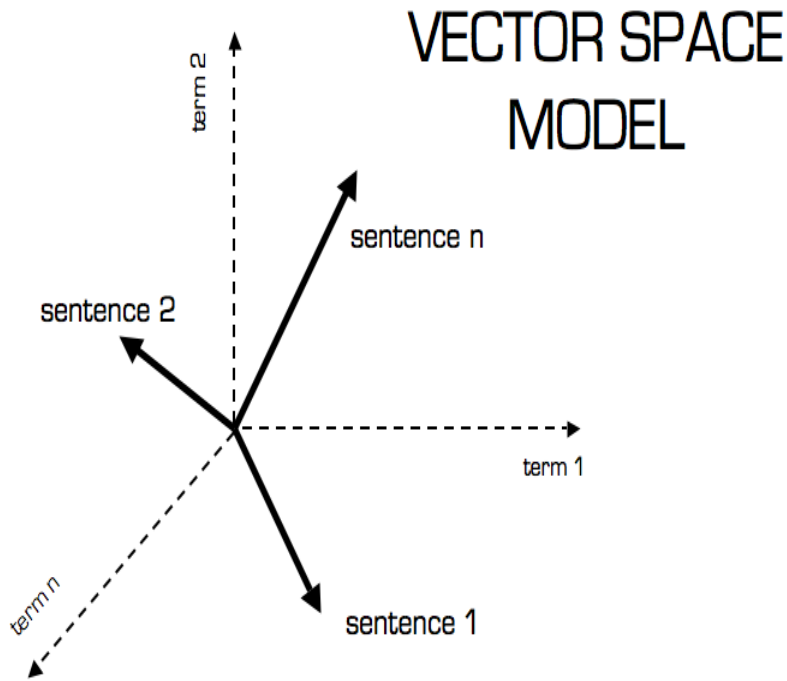
NLP Tradizionale



Outline

Vector Space Model

Differentemente dal precedente, questo approccio ha le sue basi nello sviluppo di una **rappresentazione geometrica e vettoriale** delle parole o dei documenti. Nel primo caso si parla di analisi **word-level** e nel secondo di **document-level**. Gli elementi testuali che si vogliono analizzare sono rappresentati in uno **spazio vettoriale**, le quali **dimensioni** sono gli elementi di un dizionario. Se l'obbiettivo è generare un **modello fine grained word level**, le dimensioni sono le parole del **Bag of Word** ottenuto da tutti i documenti.

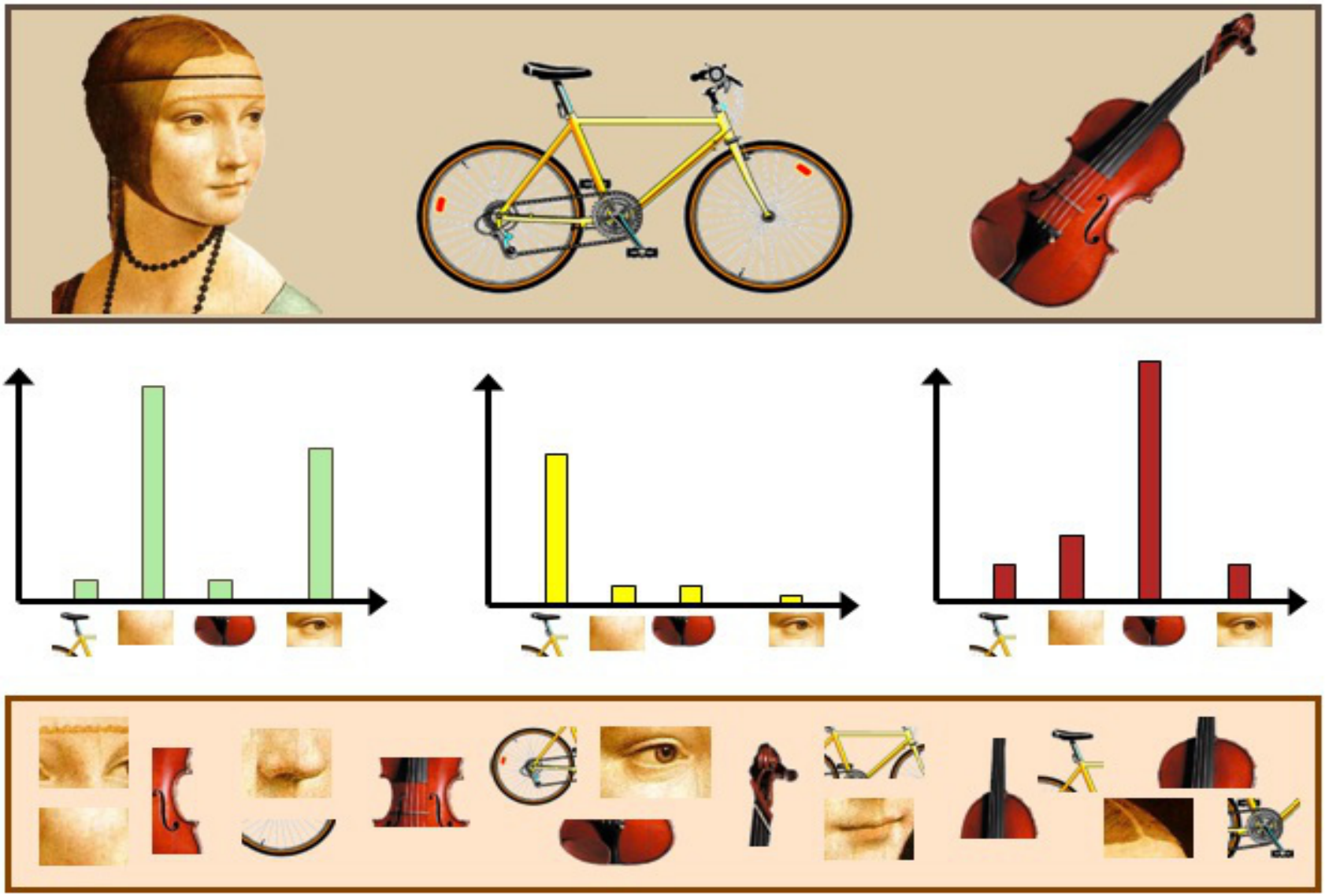


Vector Space Model

Questo approccio si articola nei seguenti **passaggi**:

- Cleaning dei dati
- Stemming o Lemmatisation
- Document/word encoding
- TF/IDF + LSA (Latent Semantic Analysis)
- Similarity (Cosine, Pearson, ...)

...limiti?



Geometrico: limiti

Per riassumere:

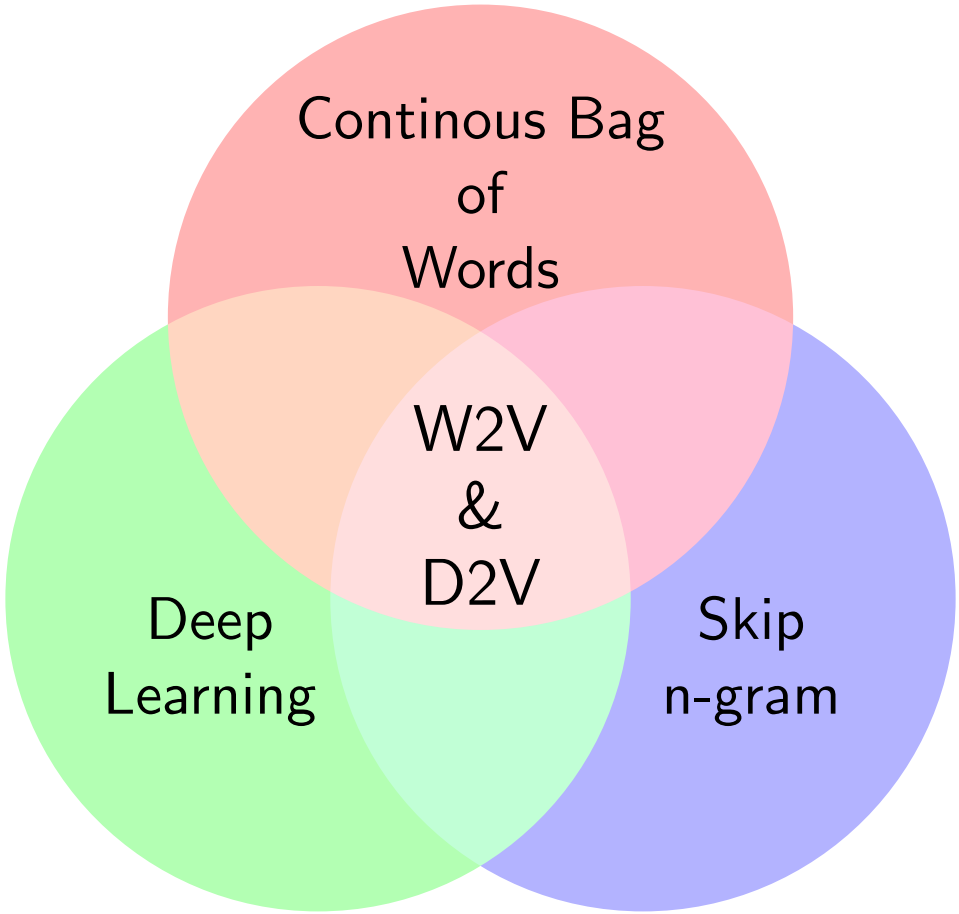
- **Molto dipendente dal preprocessing del corpus**
- Grande Bag Of World $\rightarrow$ bisogno di LSA, ma non è così banale. Perché?

  "LSA assumes that words that are close in meaning will occur in similar pieces of text"

- Semantica e ambiguità
- Un documento è un insieme non ordinato di parole

Outline

E non solo



Deep Learning: W2V & D2V

Negli ultimi anni il **Deep Learning** è stato usato in numerosi ambiti con ottimi risultati.
In particolare **Google**, con la release di **Word2Vec**, ha offerto alla community una tecnica per determinare **la similarità semantica** che un particolare **corpus di testi** assegna ad un **Bag Of Words** di parole.
Doc2Vec invece, rilasciato anche esso da **Google**, è una tecnica che si configura come una **estensione di Word2Vec** che, preso in ingresso un set di documenti (corpora), genera un **grado di similarità** reciproco.

Pros and Cons

Dopo una ampia **discussione**, seguita da una approfondita **analisi critica** di questi due approcci, siamo giunti alle seguenti **conclusioni**, che in termini di pro e contro si possono riassumere nel seguente modo:

Pros:

- Molto meno dipente da un preprocessing
- **Context-aware**
- Combina il metodo *Geometrico* con quello *NLP Tradizionale*
- **Non sfrutta una ontologia, ma la crea**

Cons:

- Tecnica **unsupervised**
- Necessità di un esperto per **validare** la similarità
- Può risultare in **GIGO** system (Garbage In Garbage Out)

Outline

Dataset

"Preprocessing is 80% of NLP work"

Lev Konstantinovskiy

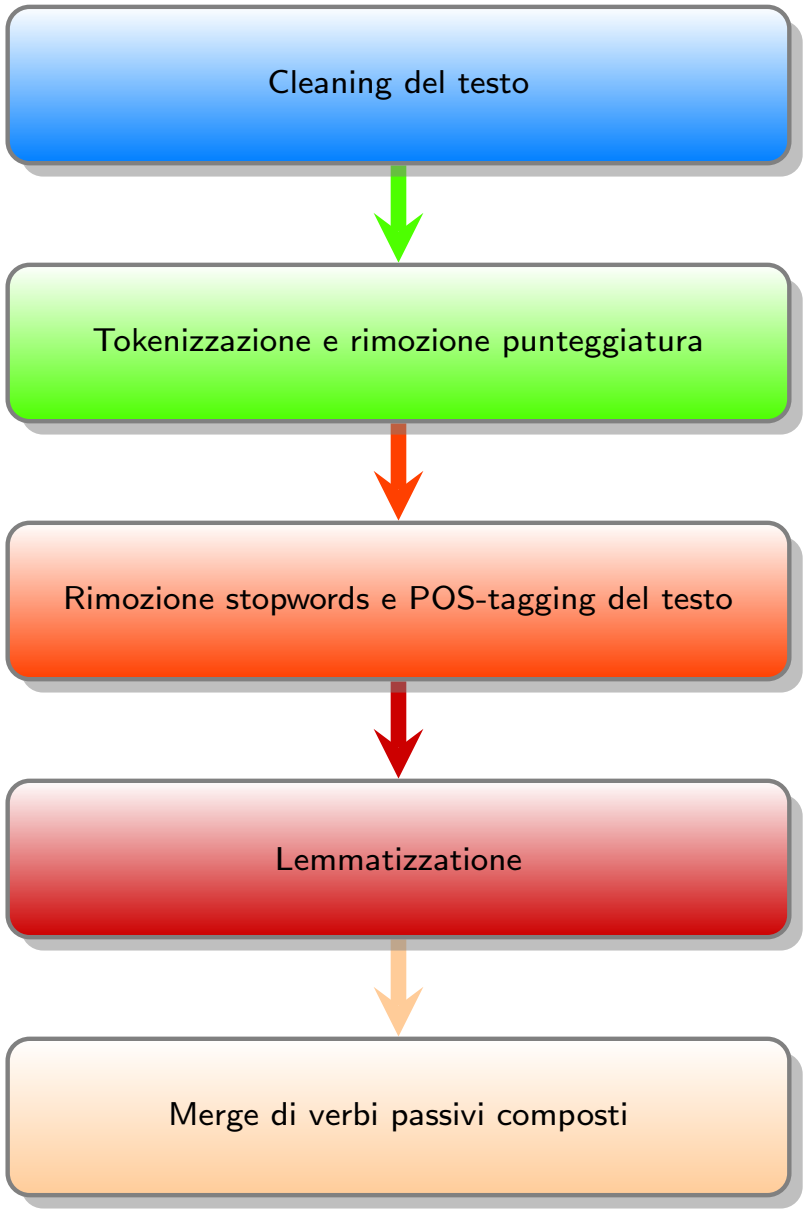
Il dataset fornitoci è composto da due corpora:

- il corpus del Sole 24 Ore con 3265 articoli, di cui 31 non hanno body
- il corpus di Radiocor con 6916 articoli

Il corpus prima del preprocessing contiene quindi 10150 articoli.
Togliendo i duplicati otteniamo 9283 articoli, cioé ci sono 867 articoli duplicati.

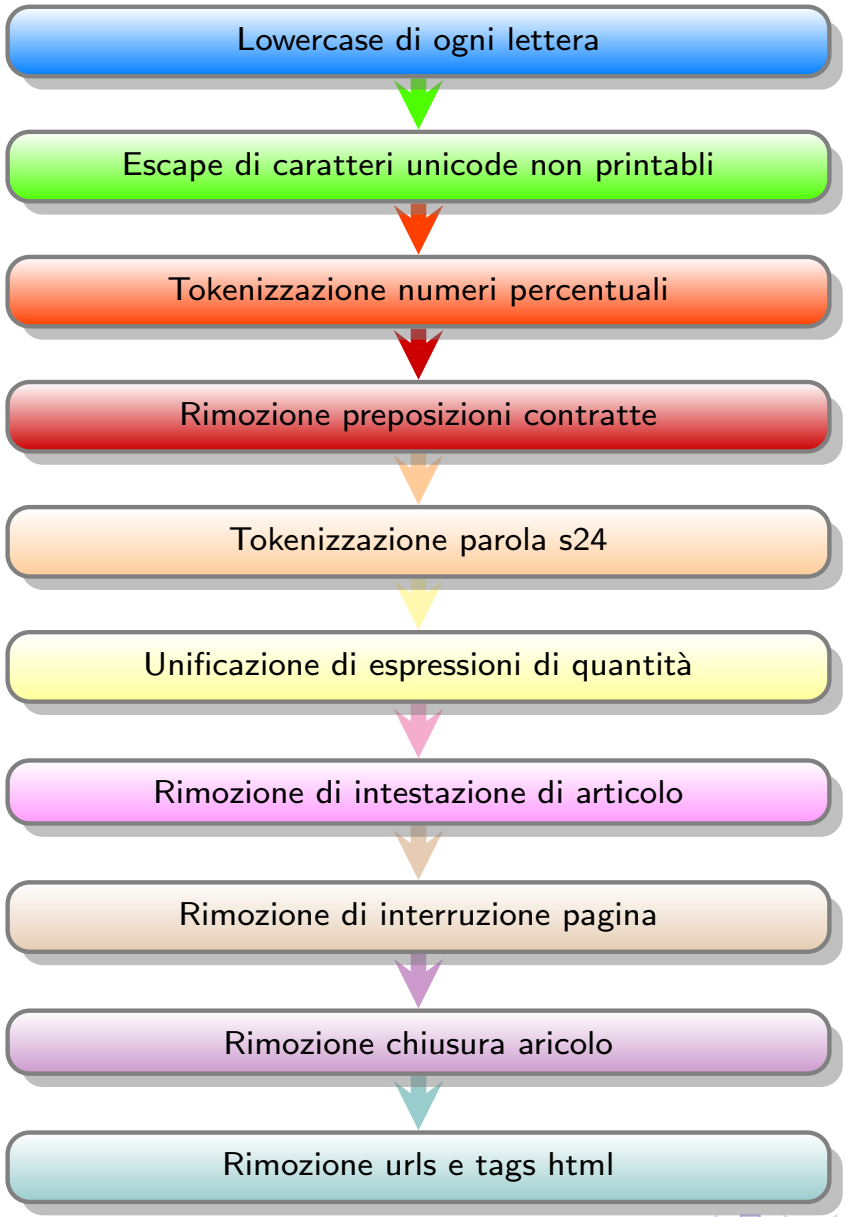
Outline

Pipeline completa



Outline

Cleaning pipeline



Outline

Hello

Qui speghiamo per bene come funziona word2vec

Outline

Hello

Qui speghiamo per bene come funziona word2vec