NATURAL LANGUAGE PROCESSING

Firenze

November, 25th 2017

Cesare Taronna

REPLY
DATA

AGENDA

- Introduction
- Real Use Case
- Data Preprocessing
- TF – IDF weighting
- Document similarity metrics

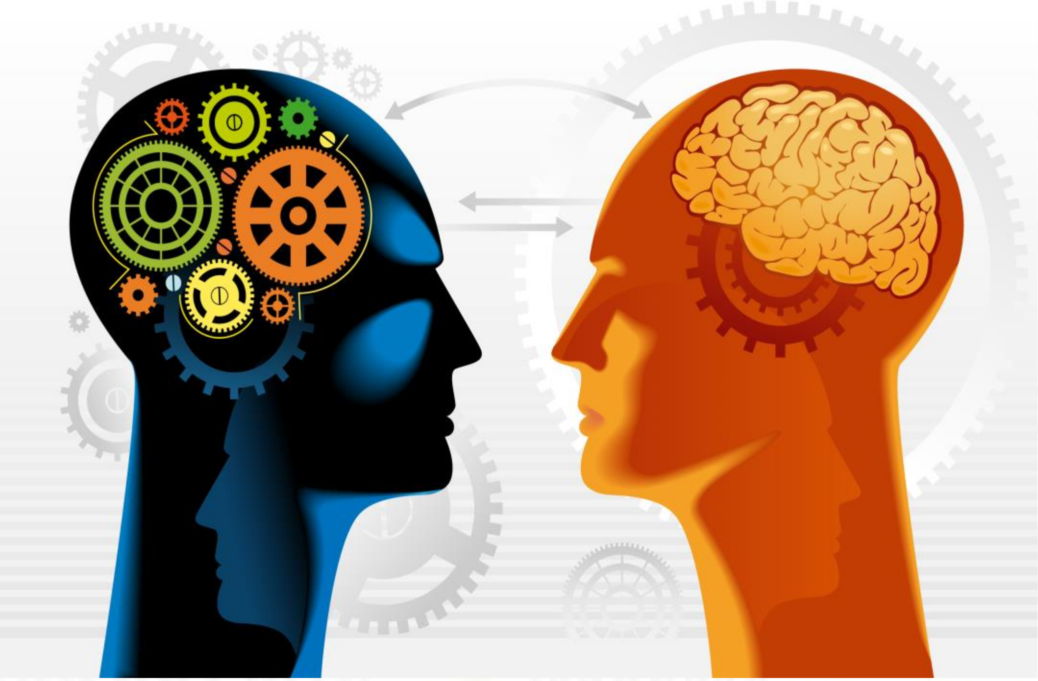
INTRODUCTION: NATURAL LANGUAGE PROCESSING

What is Natural Language?

It refers to the language spoken by people, e.g. English, Italian, Spanish. As opposed to artificial languages like Python, Java, etc.

What is Natural Language Processing?

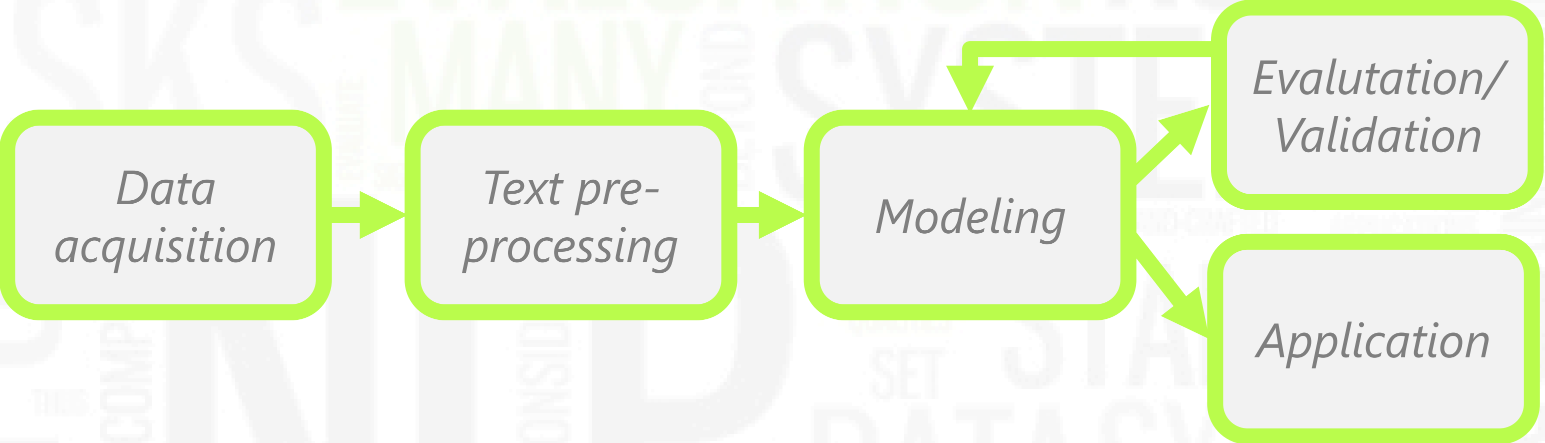
It is the field of study that focuses on the interactions between human language and computers.



INTRODUCTION: TEXT MINING

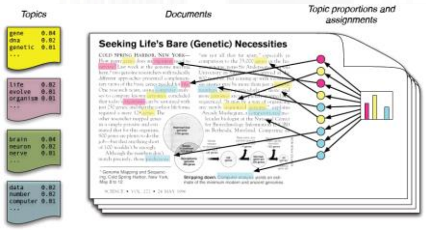
What is Text Mining?

The **goal of text mining is to discover relevant information in** text **by transforming the text into data** that can be used for further analysis.



Text mining accomplishes this through the use of a variety of analysis methodologies; natural language processing (NLP) is one of them.

INTRODUCTION: NATURAL LANGUAGE PROCESSING



NLP

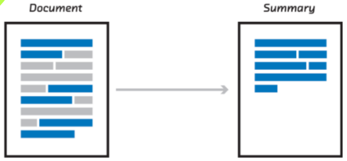
Sentiment analysis

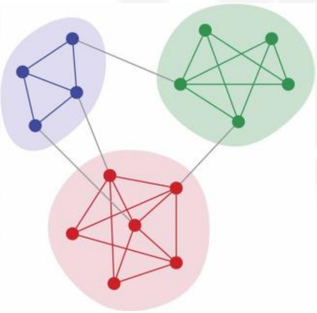
Automatic text summarization

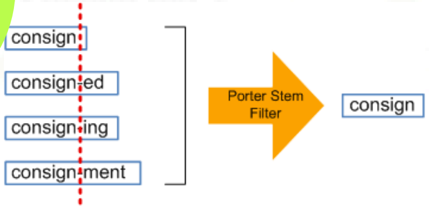
Stemming

Relathionship extraction

Topic extraction







INTRODUCTION: ACTUAL USE CASE

TV advertising & audience analysis:

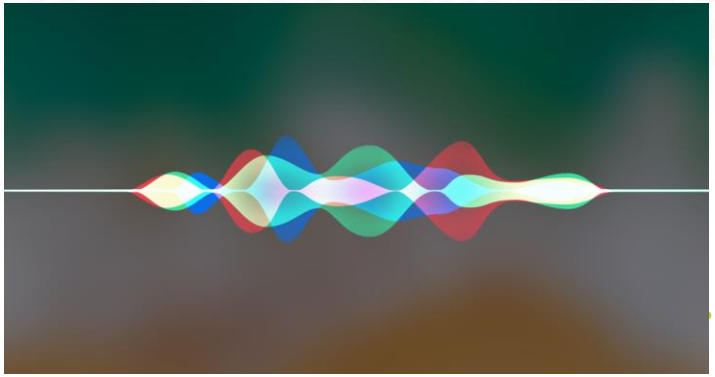
TV shows or live televised events are some of the most talked-about topics on Twitter. Marketers and TV producers can benefit from using Text Analytics **getting an understanding of how their audience feels** about certain characters, settings, storylines, featured music.

Spam detection:

To decide if an email is a spam or not, **several hundred rules are applied to each email** that passes Google's data centers. Each rule describes some attributes of a spam and has some numerical value associated with it, based on the likelihood that the attribute is a spam.

Chatting with machines:

SIRI and Cortana involve a number of technologies, including natural language processing, question analysis, data mashups, and machine learning.



REAL USE CASE

REAL USE CASE: DATA OVERVIEW

Data description:

Over 20k recipes listed by recipe rating, nutritional information and assigned category (sparse) with information about ingredients and description of how to make it.

Data source: Kaggle

Our goal:

- Retrieve most used ingredients from all recipes
- Find similarity between recipes based on the key ingredients or procedures

Techniques:

- Data pre-processing
- Stemming
- Topic extraction
- Relationship evaluation

DATA PREPROCESSING

DATA PREPROCESSING

Cleaning:

- Clean empty data
- Remove duplicate

Tokenization:

In this phase we divide text into a sequence of tokens, which roughly correspond to "words“

A token is:

- Linguistically significant
- Methodologically useful

The dog is on the table →

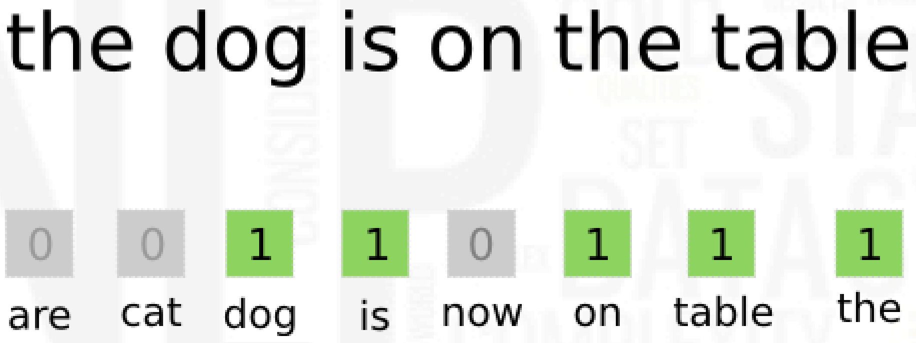
- The
- dog
- is
- on
- the
- table

DATA PREPROCESSING

Text Vectorization – Bag of words:

Turn a text into a vector of number, treating text as a Bag of words:

- Count how many times each word occurs
- Do a binary Yes-No for whether each word is conteined



the dog is on the table

are	cat	dog	is	now	on	table	the
0	0	1	1	0	1	1	1

DATA PREPROCESSING

Removing Stop words:

- Omit certain common words when doing the analysis

N-Gram:

- Look at n-grams of length n-word instead of looking at just single words

The | dog | is | on | the | table — ***Tokens***

The dog | dog is | is on | on the | the table — ***Bi-Gram***

The dog is | dog is on | is on the | on the table — ***Tri-Gram***

DATA PREPROCESSING

Stemming:

- Replace by a common root, or stem, the entire word.

Lemming:

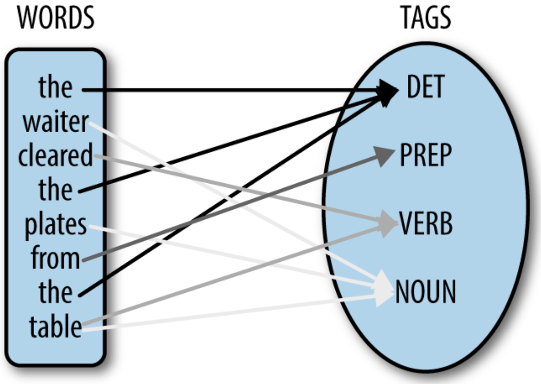
- The analog of the stem here is an actual word

affect
- affect
- affectation
- affected
- affecting
- affection
- affections
- affects

amus
- amuse
- amused
- amusement
- amusements
- amusing

close
- close
- closed
- closely
- closing

grate
- grate
- grateful
- gratefully



Part of Speech Tagging:

- Process a sequence of words and attaches a part of speech tag to each word

TERM FREQUENCY
–
INVERSE DOCUMENT FREQUENCY
(TF-IDF)

TF – IDF WEIGHTING

Definition: tf-idf is a numerical statistic that is intended to reflect how important a word is to a document in a collection of documents.

Goal: Word that appears in most of the documents should not have a big impact on the relevance and a word that appears in very few documents make them very relevant when it appears in the query.

Tf:

$$\text{tf(t,d)} = \frac{\text{\#occurences of word in document}}{\text{\#words in document}}$$

$$tf(t,d) = tf^{L^2}(t,d) = \frac{tf^{raw}(t,d)}{\sqrt{\sum_t tf^{raw}(t,d)^2}}$$

tf measures how relevant is a word for a specified document

Idf:

$$\text{idf(t,d)} = \log \frac{\text{\#documents}}{\text{\#documents containing word}}$$

$$idf^{naive}(t,D) = log \frac{\#D}{\#\{d \in D : t \in d\}}$$

IDF measures how relevant is word according the full corpus of documents

TF – IDF WEIGHTING

Now we would like to find **words that are common in one document, and not common in all of them.**

Tfidf:

$$tfidf(t,d,D) = tf(t,d)idf(t,D)$$

The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general

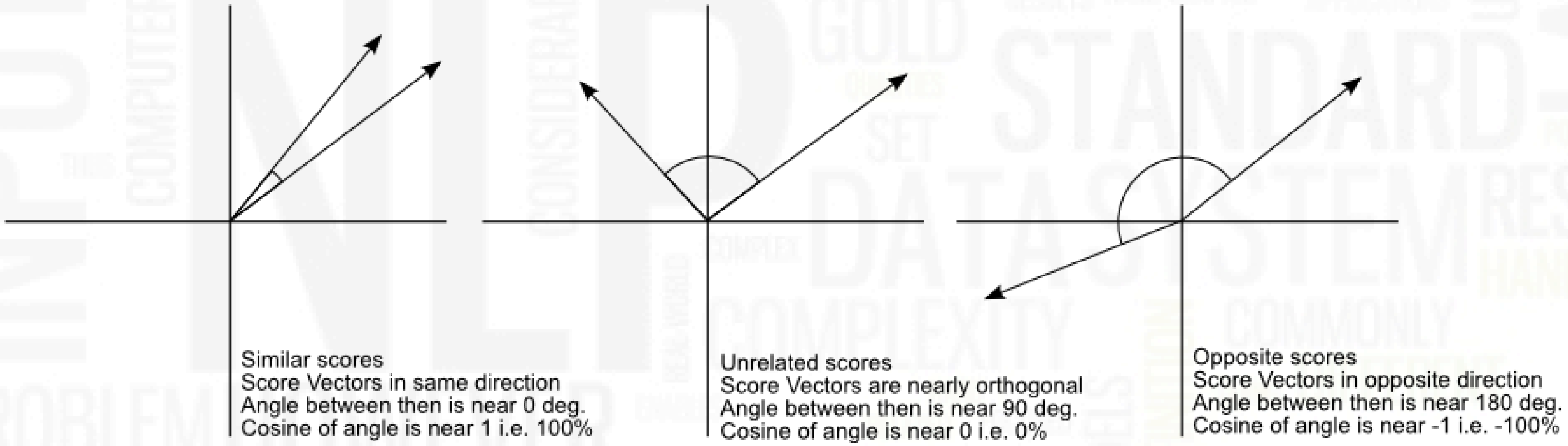
DOCUMENT SIMILARITY METRICS

COSINE SIMILARITY

Goal: find most similar documents to a given one

Issue: A common problem is looking up a document similar to a given snippet, or relatedly comparing two documents for similarity.

Cosine similarity provides a simple method for this.



Similar scores
Score Vectors in same direction
Angle between then is near 0 deg.
Cosine of angle is near 1 i.e. 100%

Unrelated scores
Score Vectors are nearly orthogonal
Angle between then is near 90 deg.
Cosine of angle is near 0 i.e. 0%

Opposite scores
Score Vectors in opposite direction
Angle between then is near 180 deg.
Cosine of angle is near -1 i.e. -100%

COSINE SIMILARITY

Cosine similarity using tf-idf weighting

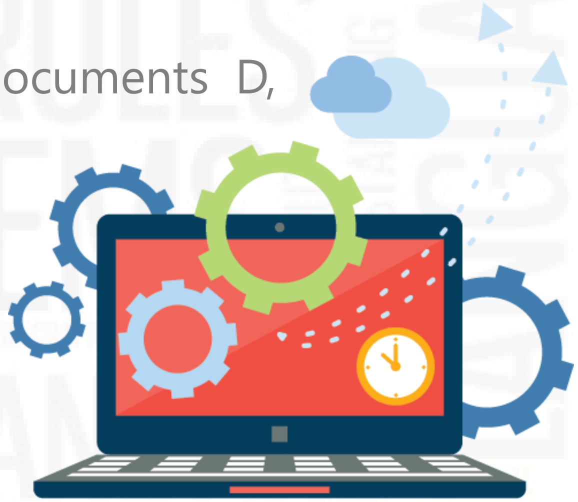
Suppose we want to find the similarity between d1 and d2:

- To each of the two documents d1,d2 in a corpus of documents D, assign its tf or tf-idf vector for each possible words

$$(v_i)_j = \mathrm{tfidf}(t_j, d_i, D)$$

where i= indices for documents

j= indices for terms in the vocabulary.



- To compare the two documents, simply find the cosine of the angle between the vectors i and i′ associated respectively to d1 and d2:

$$\frac{v_i \cdot v_{i'}}{|v_i||v_{i'}|}$$

HANDS-ON

THANK YOU

Cesare Taronna

Data Scientist

Data Reply
Via Robert Koch, 1/4
20152 - Milano - ITALY
phone: +39 02 535761
c.taronna@reply.it