



**HAL**  
open science

## Data Analysis - Lecture 10: Introduction to Text Mining

Jérémie Sublime

► **To cite this version:**

Jérémie Sublime. Data Analysis - Lecture 10: Introduction to Text Mining. Engineering school. France. 2022. hal-03845336

**HAL Id: hal-03845336**

**<https://hal.science/hal-03845336>**

Submitted on 9 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Data Analysis - Lecture 10

## Introduction to Text Mining

Dr. Jérémie Sublime

LISITE Laboratory - DaSSIP Team - ISEP  
LIPN - CNRS UMR 7030

[jeremie.sublime@isep.fr](mailto:jeremie.sublime@isep.fr)

# Plan

- 1 Introduction
- 2 Text mining tasks and processes
- 3 Analyzing similarities between and within texts
- 4 Word2vec
- 5 Conclusion

# Outline

- 1 Introduction
- 2 Text mining tasks and processes
- 3 Analyzing similarities between and within texts
- 4 Word2vec
- 5 Conclusion

# Working with texts is important and hard !

## Text Mining: Definition

Text Mining or Text data Mining or approximately **Text Analytics**, is the process of deriving high-quality information from texts.

### What texts ?

- Books
- Web pages
- Social networks
- Science articles
- ...

Very different contents, with different formats and potentially large amount of information to be extracted.

# Data analysis so far

In our previous classes, we worked only with numerical data:

- Numerical vectors
- Numerical time series

We did a bit of clustering, visualization and classification:

- All of them are based on distance or similarity functions

# Data analysis so far

In our previous classes, we worked only with numerical data:

- Numerical vectors
- Numerical time series

We did a bit of clustering, visualization and classification:

- All of them are based on distance or similarity functions

**Problem: text data represent the largest amount of data !**

How do we do all of these with text data ?

- Similarity between words, sentences, or texts ?
- Classification or clustering of texts ?
- Text time series analysis ?

# Stylometry: an early attempt at text analysis

- Stylometry is a science used in case of disputed authorship of a text: it aims at analysis writing patterns.
- Thomas C. Mendenhall pioneered stylometry in 1887 by proposing to analyze and compare the "**word spectrum**" of texts.
- He was a physicist and a meteorologist at The Ohio State University from 1873 and later became President of the Worcester Polytechnic Institute.



SCIENCE.-

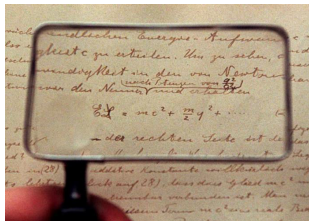
FRIDAY, MARCH 11, 1887.

*THE CHARACTERISTIC CURVES OF COM-  
POSITION.*



# The age of information overload

- Mendenhall's techniques and the idea of "word spectrum" are still used in modern text mining algorithms. But we added a few tools since:
  - Word semantic analysis
  - Sentiment analysis of words
  - Similarity links and measures between words
- The main difference between 1887 and now, is that nowadays there is way too much information coming in real time for a manual analysis to be possible !



# Examples of text analysis tasks

The most common uses of text mining include but are not limited to:

- Machine translation
- Part-of-speech tagging
- Information extraction
- Sentiment analysis from text
- Automatic quizz answering
- Text categorization
- Stylometry and plagiarism analysis

## Examples of text analysis tasks

The most common uses of text mining include but are not limited to:

- Machine translation
- Part-of-speech tagging
- Information extraction
- Sentiment analysis from text
- Automatic quizz answering
- Text categorization
- Stylometry and plagiarism analysis

Some of these applications are very important for world leaders.



# Text Mining VS Natural Language Processing

## Text Mining

It deals with the text itself

- Frequency count of words
- Sentence length
- Word spectrum
- Presence/absence of key words
- etc.

## Natural Language Processing

It deals with the underlying metadata

- Linguistics
- Computational linguistics
- Content analysis
- Stylistics
- etc.

### **Roughly summed up:**

- Text Mining is statistics oriented.
- NLP is analysis oriented.

Mining the texts: what are the **relative** texts to understand

- The text themselves ?
- The authors of the texts ?
  - The author as a writer ?
  - The author as a person in the world ?
- Things in the world ?
  - Described by the text ?
  - Linked to the text ?

## LET'S SAY YOU'VE GONE BACK IN TIME.

USE ONE, MAY BE GOING TO ASSURE THAT YOU'RE ON EARTH AND YOU CAN READ ENGLISH. SO FAR SO GOOD. BUT HOW CAN YOU BUILD ALL THE AMENITIES OF TOMORROW WHEN YOU'RE STRUCK IN THE PAST? FIRST, WE NEED SOME UNITS, BUT NO BIGGIE: THE EXACT SPEED OF LIGHT IN A VACUUM IS **DON'T WORRY, THE SAME AMOUNT OF TIME TO SWING REGARDLESS OF HOW HIGH YOU START THEM OFF. SO ZERO. THE ASB HEREIN PER SECOND. EGGED TO KNOW, A METER IS DEFINED IN TERMS OF LIGHT, BUT IF YOU CAN'T MEASURE IT ACCURATELY, THE LENGTH OF A PENDULUM THAT TAKES ONE SECOND TO SWING FROM END-TO-END WILL DO THE TRICK. PENDULUMS **COVERED**, YOU TO SAY 'ONE MISSISSIPPI', AND A STRAIN IS PRETTY MUCH THE WEIGHT OF ONE LUTHERAN CLEDGE OF AWAY.**

## FLIGHT:



**TECHNOLOGY:** HEAT BREAK TO JUST BELOW ITS BOILING POINT: IT WILL BE HELD. CONGRATULATIONS! YOU JUST INVENTED PASTEURIZATION. A MOVING ELECTRIC FIELD PRODUCES MAGNETISM, AND VICE VERSA. YOU COOPER WIRE AROUND AN IRON CORE, AND RUN ELECTRICITY THROUGH IT, AND YOU'VE GOT AN ELECTROMAGNET. DON'T HAVE ANY ELECTRICITY? PUT A MAGNET ON A WATER WHEEL, AND PUT YOUR COPPER WRAPPED IRON BEHIND THE WHEEL, AND WEY PIVOTS. YOU'RE CONVERTING MECHANICAL ENERGY INTO ELECTRICITY. DON'T HAVE ANY MAGNETS? HEAT IRON TO GET A VERY HEAVY ONE. LOOK FOR BLACK GRASSHOPPER IN BEACH SANDS THAT STICK TOGETHER, OR EXAMINE THE SERIES OF LIGHTNING STRIKES FOR MAGNETIC LODESTONES. RUN ELECTRICITY THROUGH TENDERON TO GET THE LIGHTBULB. RUN ELECTRIC CHARGE AND FORMS ALONG A WIRE WITH ENOUGH POWER AND YOU'VE INVENTED RADIO. WHEN A RADIO WAVE INTERACTS WITH A RECEIVING ANTENNA, THE EXACT SAME ELECTRICAL CHARGE CAN BE RECOVERED. INFORMATION CAN BE ENCODED IN TWO WAYS: IN THE CHARGE OF THE ELECTRICAL CURRENT, OR IN THE FREQUENCY OF THE ELECTRICAL CURRENT. SEND OUT HIGH-FREQUENCY WAVES AND ELECTRICITY WILL TRAVEL. INVENTED RADAR. THE SAME WAY SOUND AND ON MAN. SOUND WAVES A MAGNET SO IF CAN MOVE FREELY AND YOU'VE INVENTED THE COMPASS. SET THE WHEEL ON THE GROUND AND YOU'VE INVENTED THE WHEEL.

THE BEE-DIFFER. MEASURE NEW IRON IT APPEARS TO BE, AND THAT ANALYZE YOUR LATITUDE. QUANTITY AS YOU OBSERVE THEM. LEAD IN YOUR LONGEST FREQUENCY. YOU CAN DETERMINE LENGTH. JUST GET YOUR CLOCK IN LONDON TIME, TAKE THE DIFFERENCE BETWEEN LONDON HOUR AND LOCAL HOUR, AND THAT'S HOW FAR AWAY FROM THE BEE. REMEMBER YOU ARE THAT'S LENGTH. HEARD THAT, THAT WAS EASY. THAT DESCRIBES HOME CREDIT. IF YOU'RE IN STAY, THE BRITISH HAVE A 25,000 POUND PRICE OF FOR A SIMPLE WAY TO CALCULATE LENGTH.

**HEALTH:** GERMS SPREAD DISEASE, AND YOU CAN LIVE GENERALLY AVOIDING, FIGHT THEM WITH HYGIENE AND ANTIBIOTICS. WASH TO KEEP YOURSELF AND THOSE AROUND YOU CLEAN. MEDICAL INSTRUMENTS SHOULD BE CLEANED BY BOILING. PREVENTING OTHER BACTERIA FROM HAVING NEW CELLS WALLS AND REPRODUCING. IT'S LOOSER IN A NEW AGE OF ANTI-BIOTICS. TAKE THE CREDIT! IF YOU DON'T HAVE A MARK, A SECOND IS ABOUT HOW LONG IT TAKES THAT WORN SERVED UNDER A MICROSCOPE. LOOK FOR CRACKS WOUND ON LONG STRAIN. THERE'S YOUR FENCIBLE! UNWASHED SPREAD DISEASE, BUT UNCLE GERMS. CAN'T REPRODUCE ON THEIR OWN, THEY RELY ON HOST CELLS TO REPRODUCE. FIGHT THEM WITH VACCINATION. WHEN THE HEAT WILL KILL THE VIRUS, AND THEIR DEAD SHELLS ARE ENOUGH TO TEACH THE BODY TO FIGHT. HEAVIER STRAINS ALSO INCREASE, YOU'RE PREBABY GOOD FOR MOST DISEASES, BUT OUR GENERATION WERE THE FIRST TO GROW UP WITHOUT SMALLPOX. CONSUME BE SPOTS FOUND ON COX VIRUS, AND ITS CLOSE RELATIVE TO VICTIMIZE YOU WHILE YOU WEAR ENOUGH NOT TO HALL. YOUR SPREADING OF NOT BEING BUILT. DISEASE IS CHARACTERIZED BY UNDESIRABLE SYMPTOMS, EXTREME TIRISY OF WINDSOR, AND YOUR EXTREME WASHING LOG. WITHOUT TREATMENT IT'S FATAL. BUT INHALER CAN BE EXTRACTED FROM THE PANCREAS OF GOATS AND PUGS BY TIVING, STRONG AROUND THE PANCREATIC STUMP. INJECT THIS EXTRACT AND IT WILL ACT AS A MIRACULOUS TREATMENT. FORGET SWINE AND BEE. TAKE THE CREDIT!

ALSO GOOD SYNTHETIC SUBSTITUTES. ALUMINUM USED TO BE MORE VALUABLE THAN GOLD, DEPENDING ON WHEN YOU ARE AT STILL. EXTRACT IT FROM BOHRIE. HEAVY METALS CAN BE EXTRACTED. HEAVILY POLLUTED CURRENT THROUGH IT YOU'LL FIND CHLORINE IN GREENLAND, LATTITUDE ALL 2. LENGTH. 48.16. LONG FOR GLASSY WHITE CRYSTALS. ATOMS CAN BE SPLIT, WHICH RELEASES A TENDRIOUS AMOUNT OF ENERGY AND RADIATION. ENERGY CANNOT BE CREATED OR DESTROYED. BE CREATED OR DESTROYED. REUSABLE ENERGY IN MASS. INTO THE SPEED OF LIGHT SQUARED. TAKE THE CREDIT! OTHER FUN INVENTIONS: GEAR, CATALPITS, GLASSES, KITES, THERMOMETERS, SCISSORS, CIRCUITRY, RINGS, BUTTER, FISH, PLANT HOLLERS, WRITTEN AND SPOKEN LANGUAGE.

THE WHEEL, RUDDERS, CUP, UMBRELLAS, SCISSORS, LOCOMOTIVES, DELICIOUS CARBOHYDRATE WATER.

ANYWAY, HANG THIS UP IN YOUR TIME MACHINE.

# Mining the texts: what are the relative to understand

- The text themselves ?
- The authors of the texts ?
  - The author as a writer ?
  - The author as a person in the world ?
- Things in the world ?
  - Described by the text ?
  - Linked to the text ?

## LET'S SAY YOU'VE GONE BACK IN TIME.

USE ONE MAY BE GOING TO ASKING THAT YOU'RE ON EARTH AND YOU CAN READ ENGLISH SO FAR SO GOOD BUT HOW CAN YOU BUILD ALL THE AMENITIES OF TOMORROW WHEN YOU START THERE IN THE PAST? FIRST, WE NEED SOME UNITS, BUT NO BIGGER. THE EXACT SPEED OF LIGHT IN A VACUUM IS **300,000,000 METERS PER SECOND**. EASY TO KNOW. A METER IS DEFINED IN TERMS OF LIGHT, BUT IF YOU CAN'T MEASURE IT ACCURATELY, THE LENGTH OF A PENDULUM THAT TAKES ONE SECOND TO SWING FROM END-TO-END WILL DO THE TRICK. **PENDELUMS DON'T WORRY. THE SAME AMOUNT OF TIME TO SWING REGARDLESS OF HOW HIGH YOU START THEM OFF. SO DON'T WORRY ABOUT THAT EITHER. EASY TO KNOW. A SECOND ABOUT THAT PRETTY HARD TO MEASURE, BUT NOT IMPOSSIBLE.** YOUR PASTOR HAS THIS ONE TAKE THE CREDIT IF YOU DON'T HAVE A WATCH, A SECOND IS ABOUT HOW LONG IT TAKES TO SAY "ONE SECOND TO SWING FROM END-TO-END WILL DO THE TRICK. **PENDELUMS COVERED. YOU TO SAY "ONE MISSISSIPPI, AND A SECOND IS PRETTY MUCH THE WEIGHT OF ONE LUTHERAN CUBED OF WATER.**

### FLIGHT:



### TECHNOLOGY:

HEAT BREAK TO JUST BELOW ITS BOILING POINT: IT WILL BE HELD. CONGRATULATIONS! YOU JUST INVENTED PASTEURIZATION. A MOVING ELECTRIC FIELD PRODUCES MAGNETISM, AND VICE VERSA. WHEN COPPER WIRE AROUND AN IRON CORE, AND RUN ELECTRICITY THROUGH IT, AND YOU'VE GOT AN ELECTROMAGNET. DON'T HAVE ANY ELECTRICITY? PUT A MAGNET ON A WATER WHEEL, AND PUT YOUR COPPER WRAPPED IRON BEHIND THE WHEEL, AND WHAY presto, YOU'RE CONVERTING MECHANICAL ENERGY INTO ELECTRICITY. DON'T HAVE ANY MAGNETS? HEAT IRON TO GET A VERY HEAVY ONE. LOOK FOR BLACK GRAVING IN BEACH SANDS THAT STICK TOGETHER, OR EXAMINE THE ENERGY OF LIGHTNING STRIKES FOR MAGNETIC LOCKED STONES. RUN ELECTRICITY THROUGH TENDERON TO GET THE LIGHTBULB. RUN ELECTRICITY BACK AND FORTH ALONG A WIRE WITH ENOUGH POWER AND YOU'VE INVENTED RADIO. WHEN A RADIO WAVE INTERACTS WITH A RECEIVING ANTENNA, THE EXACT SAME ELECTRICAL CHARGE CAN BE RECOVERED INFORMATION CAN BE ENCODED IN TWO WAYS: IN THE CHARGE OF THE ELECTROLYTIC MAGNET CURRENT PRODUCES A HIGHER AMPLITUDE ONE ON THE FREQUENCY OF THE ELECTRICAL REFORMS. FIG IS MORE RESISTANT TO AMPITRICE ORGY IN TRANSIT. SEND OUT LOW-FREQUENCY WAVES AND REBOUND HOW AND WHEN THEY BOUNCE BACK. AND YOU'VE INVENTED RADAR. TOO THE SAME WITH SOUND, AND ON MAN. SONAR. MOUNT A MAGNET SOLE IF CAN MOVE FREELY, AND YOU'VE INVENTED THE COMPASS. EASY TO DO. INVENT THE CRYSTAL RADIO BY USING A PIECE OF QUARTZ ONE IN THE AIR AND PUT NEAR THE BEE DROPS. MEASURE HOW HIGH IT APPEARS TO BE, AND THAT ANGLE IS YOUR LATITUDE. QUANTUM MECHANICS CAN RUN SOME CIPHERS THAT YOU CAN'T RUN CLASSICALLY. YOU CAN TRY THEM AND THEY'LL REVEAL THEIR SHAPE, AND THEY'LL GENERATE ELECTRICITY WITH A VERY PRECISE FREQUENCY AS TRY TO DO. INVENT THE TUNING FORK BY USING A PIECE OF METAL THAT VIBRATES AT A CERTAIN LENGTH. JUST SET YOUR CLOCK TO LONDON TIME. TAKE THE DIFFERENCE BETWEEN LONDON HOUR AND LOCAL HOUR, AND THAT'S HOW MANY HOURS YOU ARE. TRY THAT LONGITUDE HEART THERE, THAT HAD SAID THAT **DISCOVERED HOME CREDIT** IF YOU'RE IN ST. PAUL, THE BRITISH HAVE A 20,000 POUND PRIZE FOR A SIMPLE WAY TO CALCULATE LONGITUDE.

THE WHEEL, RUDDERS, CPU, UMBRELLAS, SCISSORS, CONCHOS, DELICIOUS CARBOHYDRATE WATER.

### HEALTH:

WINGS SPREAD DISEASE, AND YOU CAN LIVE RADIALLY ANYWAY. FIGHT THEM WITH HYGIENE AND ANTIBIOTICS. WASH TO KEEP YOURSELF AND THOSE AROUND YOU CLEAN. MEDICAL INSTRUMENTS SHOULD BE CLEANED IN BOILING WATER. OTHER BACTERIA FROM HANDS CAN'T BE KILLED. THE BEST ANTIBIOTIC IS PENICILLIN. THE MOLD PENICILLIN NATURALY FOUND ON FOOD. IT'S INCREDIBLY EFFECTIVE AT STOPPING INFECTION, AND WORKS BY PREVENTING OTHER BACTERIA FROM MAKING NEW CELL WALLS AND REPRODUCING. IT'S LIQUID IN A NEW AGE OF ANTIBIOTICS. TAKE THE CREDIT IF YOU DON'T HAVE A WATCH, A SECOND IS ABOUT HOW LONG IT TAKES TO SAY "ONE MISSISSIPPI, AND A SECOND IS PRETTY MUCH THE WEIGHT OF ONE LUTHERAN CUBED OF WATER. YOUR PENICILLIN UNDOES SPREAD DISEASE, BUT UNLAME CERN. CAN'T REPRODUCE ON THEIR OWN. THEY RELY ON HOST CELLS TO REPRODUCE. FIGHT THEM WITH VACCINATION. WASH-UP THE HEAT WILL KILL THE VIRUS, AND THEIR DEAD SHELLS ARE ENOUGH TO TEACH THE BODY TO FIGHT. HEARDER STRAINS ALSO INCREASE. YOU'VE PROBABLY GOOD FOR MOST DISEASES, BUT OUR GENERATION WERE THE FIRST TO GROW UP WITHOUT SMALLPOX. COUSINS ARE SPOTS FIGHTING OUR OWN VIRUSES, AND IT'S CLOSE ENOUGH TO VACCINATE YOU YOURSELF. WEAR ENOUGH NOT TO KILL YOU. SPREADING OF HOT BEING KILLED. DISEASE IS CHARACTERIZED BY UNUSUAL SITUATION, EXTREME THIRST OF HUNGER, AND/OR EXTREME WEAKNESS. WHO? WITHOUT TREATMENT IT'S FATAL. BUT INSALIN CAN BE EXTRACTED FROM THE PANCREAS OF DOGS AND PIGS BY TIVING A STRING AROUND THE PANCREAS. DUES. INJECT THIS EXTRACT AND IT WILL LET AS A MIRACULOUS TREATMENT. FORGET SWINDLERS AND BEST. TAKE THE CREDIT.

### CHEMISTRY:

DON'T FORGET THE BASIC BUILDING BLOCKS OF MATTER: ATOMS. ATOMS CONSIST OF POSITIVELY CHARGED PROTONS, SUPERIMPOSED BY NEUTRALLY CHARGED ELECTRONS. AN ATOM WITH THE SAME NUMBER OF ELECTRONS AND PROTONS IS ELECTRICALLY NEUTRAL. OTHERWISE IT WILL HAVE A POSITIVE OR NEGATIVE CHARGE. THE MOST USEFUL OF THESE ARE NEUTRONS. NEUTRONS ARE NEUTRAL. IF IT IS. FOR OUR PURPOSES THE MOST USEFUL ONE ARE NEUTRONS. IN WITH 1 PROTON, CARBON IS WITH A NEUTRON IN WITH 2. AND OXYGEN IN WITH 8 AND 8 IN MAKE OF HYDROGEN AND OXYGEN. H<sub>2</sub>O. RADIANT LIGHT IS ALPHA, H<sub>2</sub>O. YOU'RE A MURDER AND YOU SLOWLY LOSE WEIGHT. BEING HEAVY IS BEING VALUABLE THAT BOLD. DEPENDING ON WHEN YOU ARE IT WILL EXTRACT IT FROM ROCKS. THERE'S A LOT OF URANIUM IN THE EARTH'S CRUST. CURRENT THROUGH IT YOU'LL FIND URANIUM IN GREENLAND, LANTHANUM 48-LB. LONG FOR GLASSY WHITE CRYSTALS. ATOMS CAN BE SPLIT, WHICH RELEASES A TENDRIOUS AMOUNT OF ENERGY AND IRRADIATION. ENERGY CANNOT BE CREATED OR DESTROYED. IT IS CONVERTED. REHEAT IT WITH A MIRROR IT WOULD FIND THE SPEED OF LIGHT SQUARES. TAKE THE CREDIT.

## ANYWAY. HANG THIS UP IN YOUR TIME MACHINE.

### Remark

Remember text analysis in high school ? We are trying to teach computers how to do it !

# Which interpretation ?

Finding the correct interpretation of a text is anything but obvious !

- Lexical ambiguity (Reilly 1991, Walton 1996)
  - She bagged two **silver** medals !
  - She made a **silver** speech !
- Syntactical ambiguity (Kooij 1971)
  - How do you stop a fish from **smelling** ?
  - *cut his nose* ?
- Inflective ambiguity (1996; Fowler and Aaron 1998))
  - Bob has devised a **scheme** to save costs by recycling paper. Therefore, Bob is a **schemer**, and should not be trusted.

Words with more than one meaning

Unclear meaning : Odor or smell ?

Words used more than once in a sentence or paragraph, but with different meanings each time.

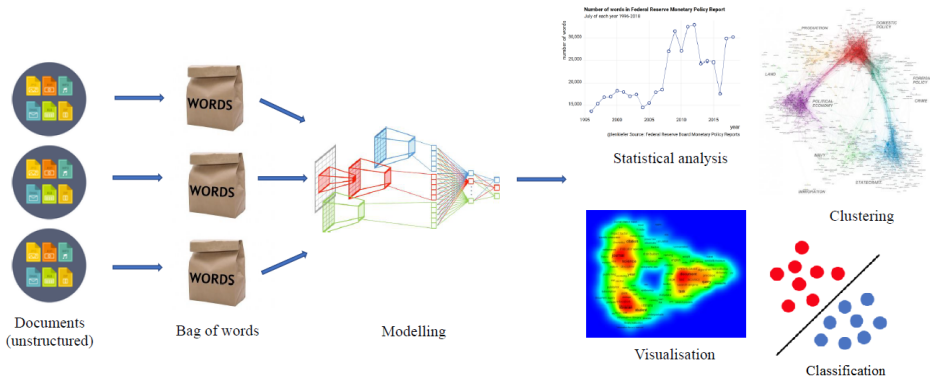
# Outline

- 1 Introduction
- 2 Text mining tasks and processes**
- 3 Analyzing similarities between and within texts
- 4 Word2vec
- 5 Conclusion





# Typical process of text mining (1/2)



# Typical process of text mining (2/2)

## 1 Transform texts and documents into structured data

- Term-Document Matrices (TDM)
- Entities and relations
- etc.

	I	like	hate	maths
D1	1	1	0	1
D2	1	0	1	1

Table: Example of document term matrix

## 2 Apply traditional data mining techniques to the above data structures

- Clustering
- Classification
- Social media analysis
- etc.

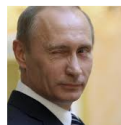
# Text mining tasks

We are going to take a look at some key tasks in text mining :

- Sentiment Analysis
- Document Summarization
- Entity and Relationship Extraction
- Tokenization and vectorization of texts

# Sentiment analysis

- Sentiment Analysis is also known as "**opinion mining**"
- It determines the attitude, polarity, or emotions from documents
- Polarity : positive, negative, neutral
- Emotions: angry, sad, happy, bored, afraid, etc.



## Method

- 1 Identify individual keywords and phrases and map them to different emotions/sentiment scales
- 2 Adjust the sentiment value of a concept based on modifications surrounding it

# Document Summarization

- Extract the major key points of a document to create a summary

## Methods

- **Extraction:** select a subset of existing words, phrases, or sentences to build a summary
- **Abstraction:** use natural language generation techniques to build a summary that is similar to natural language

# Entity and relationship extraction

- Named Entity Recognition (NER) : identify named entities in texts and classify them into pre-defined categories such as person names, organizations, locations, date and times, etc.
- Relationship Extraction: identify association entities.
- Example:
  - Tom lives at 7 bvd Saint Michel, Paris.

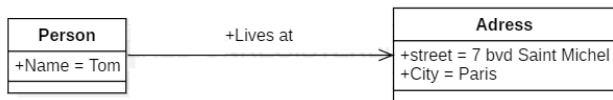
# Entity and relationship extraction

- Named Entity Recognition (NER) : identify named entities in texts and classify them into pre-defined categories such as person names, organizations, locations, date and times, etc.
- Relationship Extraction: identify association entities.
- Example:
  - Tom lives at 7 bvd Saint Michel, Paris.
  - Tom **lives at** 7 bvd Saint Michel, Paris.



# Entity and relationship extraction

- Named Entity Recognition (NER) : identify named entities in texts and classify them into pre-defined categories such as person names, organizations, locations, date and times, etc.
- Relationship Extraction: identify association entities.
- Example:
  - Tom lives at 7 bvd Saint Michel, Paris.
  - Tom **lives at** 7 bvd Saint Michel, Paris.



# Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters into sentences, symbols, or words. For text mining, we are particularly interested in words.

# Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters into sentences, symbols, or words. For text mining, we are particularly interested in words.

## 1) Identifying the words (tokens) of a document

- The word delimiter identification is important. It will often be spaces or punctuation.
- Some characters are less obvious : e.g. "-" might be a delimiter in English language, but not always in French.

# Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters into sentences, symbols, or words. For text mining, we are particularly interested in words.

## I) Identifying the words (tokens) of a document

- The word delimiter identification is important. It will often be spaces or punctuation.
- Some characters are less obvious : e.g. "-" might be a delimiter in English language, but not always in French.

## II) Creating a dictionary

- Potentially, the number of words is very important and there may be redundancies.
- It will be necessary to treat the redundancies. For instance using **stemming** or **lemmatization** : Putting all words to their root form.

# Tokenization: Bag of words (1/2)

Tokenization is the process of splitting a stream of characters into sentences, symbols, or words. For text mining, we are particularly interested in words.

## I) Identifying the words (tokens) of a document

- The word delimiter identification is important. It will often be spaces or punctuation.
- Some characters are less obvious : e.g. "-" might be a delimiter in English language, but not always in French.

## II) Creating a dictionary

- Potentially, the number of words is very important and there may be redundancies.
- It will be necessary to treat the redundancies. For instance using **stemming** or **lemmatization** : Putting all words to their root form.

## III) From text to vector

- The absence or presence of a word is then assessed in each text, thus creating a binary vector.
- You can also count the number of appearance of a word. We are then talking about **weighting** words.

## Tokenization: Bag of words (2/2)

Example of tokenization:

- I like mathematics.
- I hate maths !!!!
- I hate mathematics

	I	like	hate	maths
D1	1	1	0	1
D2	1	0	1	1
D3	1	0	1	1

## Tokenization: The lexical database issue

Even with tokenization or lemmatization, covering a whole language requires very large lexical databases with some semantic structures.

- For English language, **WORDNET** lists, classifies and relates the semantic and lexical content of the language.
- A **Synset** (synonym set) is a database with groups of interchangeable words.
  - Example: car, automobile, machine, motorcar, etc.
  - In the bag of word representation, it can significantly reduce dimensionality.
- **Ontologies** are a representation based on concept trees and hierarchical relationships.
  - Ontologies already exist for many domains and applications
  - They are standardized

# Stemming

## Stemming: Definition

- Stemming consist in reducing a word to its root form (stem) by removing affixes.
- Stemming returns a root that might not be an actual word.

So, what if, instead of thinking about solving your whole life, you just think about adding additional good things. One at a time. Just let your pile of good things grow.

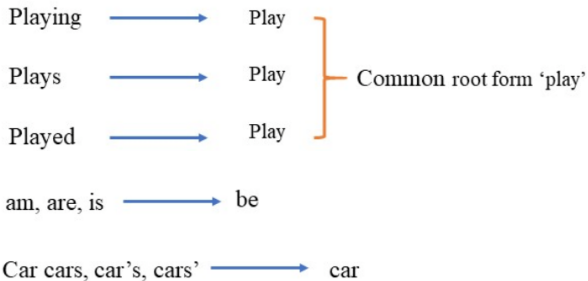
So, what if, instead of think about solv you whol liv, you just think about add add good thing. One at a tim. Just let you pil of good thing grow.



# Lemmatization

## Lemmatization: Definition

- Lemmatization consists in analyzing a term in order to identify its canonical form (lemma). The idea is to ignore all modifications due to conjugations or declinations to keep a single form of the word.
- Unlike stemming, lemmatization has to return an actual word.



### Example:

*the boy's cars are different colors* → *the boy car be different color*

# Part of speech

## Part of Speech: Definition

Part of speech (POS) proposes to distinguish the words of a sentence according to their lexical categories (e.g. noun, verb, adjective, etc.).

Example:

*They refuse to permit us to obtain the refuse permit.*

The first *refuse* is a verb, synonym of denying. The second *refuse* is a noun, synonym of trash.

- *They refuse to permit us to obtain the refuse permit.*
- (they,PRP), (refuse,VB), (to,TO), (permit,VB), (us, PRP), (to, TO), (obtain,VB), (the,DT), (refuse, NN), (permit,NN)

# POS with lemmatization and stemming

- Part of Speech must be used before lemmatization or stemming :
  - ① Use POS to determine the lexical category and if possible grammatical form of the words.
  - ② Use lemmatization or stemming for tokenization or lexical database search
- Lemmatization algorithms are more complex than stemming, but they often prove more effective for text mining.
- In some languages, stemming and lemmatization give the same results: Chinese and Japanese for instance.

## Remark

For translation applications, stemming is still more commonly used than lemmatization.

- Stems are more often the same across languages than normalized words.
- Lemmatization requires extensive databases in both languages

# POS and stemming for basic translation

- 1 Use part of speech to determine the lexical category and if possible grammatical form of the words in the source language
- 2 Apply stemming to the text in the source language
- 3 Search for equivalent stems in the target language
- 4 Reverse part of speech in the target language to reassign lexical roles and grammatical functions

# POS and stemming for basic translation

- 1 Use part of speech to determine the lexical category and if possible grammatical form of the words in the source language
- 2 Apply stemming to the text in the source language
- 3 Search for equivalent stems in the target language
- 4 Reverse part of speech in the target language to reassign lexical roles and grammatical functions

It does not always work well, and this is how you get weird online translations.

# Additional Text Cleaning Considerations

- Handling large documents and large collections of text documents may cause memory issues.
- Some formats such as HTML, XML or PDF have their own structures that require specific splitting methods.
- Languages other than English may cause issues:
  - Accents, apostrophes and cedillas
  - Non latin based characters and ideograms
- Unicode characters may need to be normalized in formats such as UTF8.
- Handling of domain specific words, phrases, and acronyms may prove useful and could require specific dictionaries.
- Handling or removing numbers, such as dates and amounts can be considered.
- Typos and misspellings happen and should be considered !

# Tools to count word occurrences: IDF & TF-IDF

Let us consider  $D = \{d_1, \dots, d_j, \dots\}$  a set of document. Then for any term  $t_i$ , we have:

- $tf_{ij}$  the number of occurrence of  $t_i$  in a document  $d_j$ . It is the **Term Frequency (TF)**.
- The **Inverse Document Frequency (IDF)** for  $t_i$  is :

$$idf_i = \ln \left( \frac{|D|}{|\{d | t_i \in d\}|} \right)$$

With  $|\{d | t_i \in d\}|$  the number of documents where  $t_i$  appears

- The **Term Frequency Inverse Document Frequency (TF-IDF)** :

$$tfidf_{ij} = tf_{ij} \cdot idf_i$$

The TF-IDF is used to know the importance of a term  $t_i$  inside a document  $d_j$  relatively to its importance in the whole corpus  $D$ .

# IDF & TF-IDF : Example of use

- I love Paris
- I love Rome

Term Frequency

	Doc 1	Doc 2
I	1	1
Love	1	1
Paris	1	0
Rome	0	1

IDF

	IDF
I	0
Love	0
Paris	1
Rome	1

TFIDF

	Doc 1	Doc 2
I	0	0
Love	0	0
Paris	1	0
Rome	0	1

Terms that can distinguish different documents are given greater weights.



## Normalizing the term frequency

Since some terms may be a lot more frequent than others in large corpuses, there are several possible normalizations of the TF to reduce scaling issues.

binary	$0, 1$
raw frequency	$t_i$
log normalization	$1 + \log(t_i)$
0.5 max normalization	$0.5 + 0.5 \cdot \frac{t_i}{\operatorname{argmax}_{d \in D} t_i^d}$
max normalization	$K + (1 - K) \cdot \frac{t_i}{\operatorname{argmax}_{d \in D} t_i^d}$

# Outline

- 1 Introduction
- 2 Text mining tasks and processes
- 3 Analyzing similarities between and within texts**
- 4 Word2vec
- 5 Conclusion

# Similarity measures in text mining

- Similarity measures are key for a lot of Machine Learning and Data Mining methods : visualization, clustering, classification, etc.
- Alongside distance functions, they characterize the similarities or dissimilarities between objects.
- In the particular case of text mining, we need to measure similarities between sentences, documents, or even corpuses of documents.
- We have seen that texts can be turned into numerical vectors.

## Similarity measures in text mining

- Similarity measures are key for a lot of Machine Learning and Data Mining methods : visualization, clustering, classification, etc.
- Alongside distance functions, they characterize the similarities or dissimilarities between objects.
- In the particular case of text mining, we need to measure similarities between sentences, documents, or even corpuses of documents.
- We have seen that texts can be turned into numerical vectors.

**Now, which similarity measure should we use ?**

## Reminders on the properties of similarities

A similarity measure between 2 vectors  $u$  and  $v$  must have the following properties:

- Non-negativity :  $s(u, v) \geq 0 \quad \forall u, v$
- Symmetry :  $s(u, v) = s(v, u) \quad \forall u, v$
- Maximality :  $s(u, u) = s(v, v) = 1 \quad \forall u, v$
- Normalization :  $\begin{cases} s(u, v) = 1 \Leftrightarrow u = v \\ s(u, v) < 1 \end{cases} \quad \forall u, v$

## From similarity to distance

The dissimilarity between 2 objects can be seen as the distance between them. Therefore, distance and similarity are directly linked.

$$d(u, v) = 1 - s(u, v)$$

In addition to the non-negativity and symmetry properties that are the same than for similarities, distances also have the following properties:

- Identity of indiscernibles :  $d(u, v) = 0 \Leftrightarrow u = v \quad \forall u, v$
- Triangle inequality :  $d(u, w) \leq d(u, v) + d(v, w) \quad \forall u, v, w$

# Similarity measures: Indian Cricket example (1/3)



Considering only the 3 words from the above documents: 'sachin', 'dhoni', 'cricket'

### Doc Sachin: Wiki page on Sachin Tendulkar

Dhoni - 10  
Cricket - 50  
Sachin - 200

### Doc Dhoni: Wiki page on Dhoni

Dhoni - 400  
Cricket - 100  
Sachin - 20

### Doc Dhoni\_Small: Subsection of wiki on Dhoni

Dhoni - 10  
Cricket - 5  
Sachin - 1

Let  $\vec{u}$  and  $\vec{v}$  be 2 vectors with word counts for 2 documents.

### Euclidian Distance:

$$\|\vec{u} - \vec{v}\|_2 = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}$$

# Similarity measures: Indian Cricket example (1/3)



Considering only the 3 words from the above documents: 'sachin', 'dhoni', 'cricket'

### Doc Sachin: Wiki page on Sachin Tendulkar

Dhoni - 10  
Cricket - 50  
Sachin - 200

### Doc Dhoni: Wiki page on Dhoni

Dhoni - 400  
Cricket - 100  
Sachin - 20

### Doc Dhoni\_Small: Subsection of wiki on Dhoni

Dhoni - 10  
Cricket - 5  
Sachin - 1

Let  $\vec{u}$  and  $\vec{v}$  be 2 vectors with word counts for 2 documents.

### Euclidian Distance:

$$\|\vec{u} - \vec{v}\|_2 = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}$$

### Cosine Similarity:

$$\text{Cos}(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \times \|\vec{v}\|} = \frac{\sum_{i=1}^n u_i \times v_i}{\sqrt{\sum_{i=1}^n u_i^2} \sqrt{\sum_{i=1}^n v_i^2}}$$



# Similarity measures: Indian Cricket example (2/3)



Considering only the 3 words from the above documents: 'sachin', 'dhoni', 'cricket'

### Doc Sachin: Wiki page on Sachin Tendulkar

Dhoni - 10  
Cricket - 50  
Sachin - 200

### Doc Dhoni: Wiki page on Dhoni

Dhoni - 400  
Cricket - 100  
Sachin - 20

### Doc Dhoni\_Small: Subsection of wiki on Dhoni

Dhoni - 10  
Cricket - 5  
Sachin - 1

### Document - Term Matrix (Word Counts)

Word Counts	"Dhoni"	"Cricket"	"Sachin"
<i>Doc Sachin</i>	10	50	200
<i>Doc Dhoni</i>	400	100	20
<i>Doc Dhoni_Small</i>	10	5	1



### Similarity Metrics

Similarity or Distance Metrics	Total Common Words	Euclidean distance	Cosine Similarity
<i>Doc Sachin &amp; Doc Dhoni</i>	70	432.4	0.15
<i>Doc Dhoni &amp; Doc Dhoni_Small</i>	37	204.0	0.23
<i>Doc Sachin &amp; Doc Dhoni_Small</i>	27	401.85	0.77

# Similarity measures: Indian Cricket example (3/3)

$$\|\vec{u} - \vec{v}\|_2 = \sqrt{\sum_{i=1}^n (u_i - v_i)^2}$$

$$\text{Cos}(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \times \|\vec{v}\|} = \frac{\sum_{i=1}^n u_i \times v_i}{\sqrt{\sum_{i=1}^n u_i^2} \sqrt{\sum_{i=1}^n v_i^2}}$$

Document - Term Matrix (Word Counts)

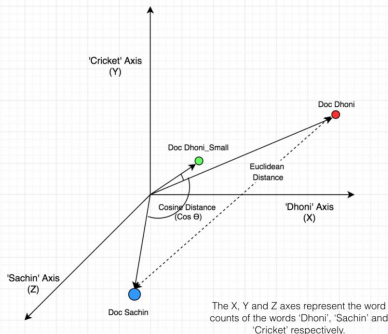
Word Counts	"Dhoni"	"Cricket"	"Sachin"
Doc Sachin	10	50	200
Doc Dhoni	400	100	20
Doc Dhoni_Small	10	5	1



Similarity Metrics

Similarity or Distance Metrics	Total Common Words	Euclidean distance	Cosine Similarity
Doc Sachin & Doc Dhoni	70	432.4	0.15
Doc Dhoni & Doc Dhoni_Small	37	204.0	0.23
Doc Sachin & Doc Dhoni_Small	27	401.85	0.77

## 3D projection in space



# Jaccard index and Jaccard distance

The Jaccard index is very popular and assesses co-occurrences between documents with a normalization mechanism.

$$J(u, v) = \frac{|u \cap v|}{|u \cup v|} = \frac{M_{11}}{p - M_{00}}$$

$$0 \leq J \leq 1$$

- $p$  is the number of terms
- $M_{11}$  is the number of co-occurrences
- $M_{00}$  is the number of co-absences

Document	databases	huge	image	permanently	store
A	1	1	1	0	0
B	1	0	1	1	1
C	1	0	1	0	1
D	1	0	1	0	1

$$J(B, C) = \frac{3}{4} = \frac{3}{5 - 1} = 0.75$$

$$J(C, D) = 1$$

# Jaccard index and Jaccard distance

The Jaccard index is very popular and assesses co-occurrences between documents with a normalization mechanism.

$$J(u, v) = \frac{|u \cap v|}{|u \cup v|} = \frac{M_{11}}{p - M_{00}}$$

$$0 \leq J \leq 1$$

- $p$  is the number of terms
- $M_{11}$  is the number of co-occurrences
- $M_{00}$  is the number of co-absences

Document	databases	huge	image	permanently	store
A	1	1	1	0	0
B	1	0	1	1	1
C	1	0	1	0	1
D	1	0	1	0	1

$$J(B, C) = \frac{3}{4} = \frac{3}{5 - 1} = 0.75$$

$$J(C, D) = 1$$

## Jaccard Distance

Deducing the distance from the Jaccard Index is very easy :

$$d(u, v) = 1 - J(u, v)$$

## Further considerations on similarity indexes

There are **a lot** of similarity measures and distance functions that can be used for text mining.

- Like for any data analysis task, some will work better than other depending on the data features and the type of study.
- For instance, Euclidian distance is perfectly fine when using TF weighting and documents of similar sizes.

## Further considerations on similarity indexes

There are **a lot** of similarity measures and distance functions that can be used for text mining.

- Like for any data analysis task, some will work better than other depending on the data features and the type of study.
- For instance, Euclidian distance is perfectly fine when using TF weighting and documents of similar sizes.

### Some extra indexes for binary analysis

- Dice :  $s(u, v) = \frac{2 \times |u \cap v|}{|u| + |v|}$
- Overlap index :  $s(u, v) = \frac{2 \times |u \cap v|}{\min(|u|, |v|)}$
- Tanimoto index :  $s(u, v) = \frac{|u \cap v|}{|u| + |v| - |u \cap v|}$

## Similarity measures: Chi square (1/3)

In Lecture 3, we saw that the **Chi square** could be used for the correlation analysis of categorical data. Let's see how to use it for similarity between texts.

Corpus of 3 documents (Grossman, page 71)

- D1 : "shipment of gold damaged in a fire"
- D2 : "delivery of silver arrived in a silver truck"
- D3 : "shipment of gold arrived in a truck"

	D1	D2	D3
arrived	0	1	1
damaged	1	0	0
delivery	0	1	0
fire	1	0	0
gold	1	0	1
shipment	1	0	1
silver	0	2	0
truck	0	1	1

## Similarity measures: Chi square (2/3)

Terms	D1	D2	D3	Sum
arrived	0	1	1	2
damaged	1	0	0	1
delivery	0	1	0	1
fire	1	0	0	1
gold	1	0	1	2
shipment	1	0	1	2
silver	0	2	0	2
truck	0	1	1	2
Sum	4	5	4	13

Computing the  $\chi^2$  can give you an idea of the potential association between documents by testing the independence hypothesis.

$$\chi^2 = \sum_{k=1}^K \sum_{l=1}^L \frac{(o_{kl} - e_{kl})^2}{e_{kl}} = 13.65$$

$$0.25 < p < 0.5$$

We can also compute the Total Inertia (total amount of information):

$$\phi^2 = \frac{\chi^2}{N} = \frac{13.65}{13} = 1.05$$



## Similarity measures: Chi square (3/3)

The matrix of standardized residuals identifies attractions and repulsions between terms and documents:

$$r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$$

Terms	D1	D2	D3
arrived	-0.784	<b>0.263</b>	<b>0.490</b>
damaged	<b>1.248</b>	-0.620	-0.555
delivery	-0.555	<b>0.992</b>	-0.555
fire	<b>1.248</b>	-0.620	-0.555
gold	<b>0.490</b>	-0.877	<b>0.490</b>
shipment	<b>0.490</b>	-0.877	<b>0.490</b>
silver	-0.784	<b>1.403</b>	-0.784
truck	-0.784	<b>0.263</b>	<b>0.490</b>

## Similarity measures: Chi square (3/3)

The matrix of standardized residuals identifies attractions and repulsions between terms and documents:

$$r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$$

Terms	D1	D2	D3
arrived	-0.784	<b>0.263</b>	<b>0.490</b>
damaged	<b>1.248</b>	-0.620	-0.555
delivery	-0.555	<b>0.992</b>	-0.555
fire	<b>1.248</b>	-0.620	-0.555
gold	<b>0.490</b>	-0.877	<b>0.490</b>
shipment	<b>0.490</b>	-0.877	<b>0.490</b>
silver	-0.784	<b>1.403</b>	-0.784
truck	-0.784	<b>0.263</b>	<b>0.490</b>

From there, the contribution to the  $\chi^2$  can be computed to assess the impact of associations in the global information quantity.

$$c_{kl} = 100 \times \frac{r_{kl}^2}{\chi^2}$$

Terms	D1	D2	D3
arrived	4.508	0.507	1.761
damaged	<b>11.412</b>	2.818	2.254
delivery	2.254	7.213	2.254
fire	<b>11.412</b>	2.818	2.254
gold	1.761	5.635	1.761
shipment	1.761	5.635	1.761
silver	4.508	<b>14.427</b>	4.508
truck	4.508	0.507	1.761

## Similarity measures: Chi square (3/3)

The matrix of standardized residuals identifies attractions and repulsions between terms and documents:

$$r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$$

Terms	D1	D2	D3
arrived	-0.784	<b>0.263</b>	<b>0.490</b>
damaged	<b>1.248</b>	-0.620	-0.555
delivery	-0.555	<b>0.992</b>	-0.555
fire	<b>1.248</b>	-0.620	-0.555
gold	<b>0.490</b>	-0.877	<b>0.490</b>
shipment	<b>0.490</b>	-0.877	<b>0.490</b>
silver	-0.784	<b>1.403</b>	-0.784
truck	-0.784	<b>0.263</b>	<b>0.490</b>

From there, the contribution to the  $\chi^2$  can be computed to assess the impact of associations in the global information quantity.

$$c_{kl} = 100 \times \frac{r_{kl}^2}{\chi^2}$$

Terms	D1	D2	D3
arrived	4.508	0.507	1.761
damaged	<b>11.412</b>	2.818	2.254
delivery	2.254	7.213	2.254
fire	<b>11.412</b>	2.818	2.254
gold	1.761	5.635	1.761
shipment	1.761	5.635	1.761
silver	4.508	<b>14.427</b>	4.508
truck	4.508	0.507	1.761

We can see that most of the information comes from the attractions (D2,silver) and (D1,[damaged,fire])

# Principle of Correspondence Analysis

Let us consider  $R$  the matrix of standard residual  $r_{kl} = \frac{o_{kl} - e_{kl}}{\sqrt{e_{kl}}}$ , and  $M$  a standardized matrix so that  $M = \frac{R}{|D|}$ , with  $|D|$  the number of documents.

The Correspondence Analysis (CA) consists in computing the singular value decomposition (SVD) of  $M$ , so that :  $M = U\Delta V^T$

- $U_{(K \times K)}$  contains the  $K$  left eigenvectors (for the lines)
- $\Delta_{(K \times L)}$  is a matrix whose diagonal elements are the singular values (eigenvalues can be acquired by squaring them)
- $V_{(L \times L)}$  contains the  $L$  right eigenvectors (for the columns)

Each singular value  $\delta_h$  is defined so that:

$$\begin{cases} M\vec{v}_k = \delta_k \vec{u}_k \\ M^T \vec{u}_k = \delta_k \vec{v}_k \end{cases}$$

Then we get the eigenvalues:

$$\lambda_k = \frac{\delta_k^2}{K-1}$$

## Goal

From there, CA aims at finding projections of the terms (or documents) that maximizes the dispersion. It is similar to PCA and correlation circles.

# Correspondence Analysis: Example

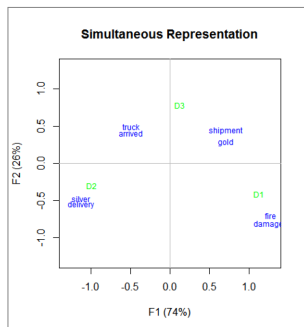
The projection is made possible thanks to the transition relationship (almost barycentric), meaning: The coordinates of a column term can be obtained based on all lines terms and vice versa.

Coordinates of term  $k$  on the 1st component:

$$F_{k1} = \frac{1}{\sqrt{\lambda_1}} \sum_{l=1}^L \frac{o_{kl}}{n_k} \times G_{l1}$$

Coordinates of document  $l$  on the 1st component:

$$G_{l1} = \frac{1}{\sqrt{\lambda_1}} \sum_{k=1}^K \frac{o_{kl}}{n_l} \times F_{k1}$$



## Correspondence Analysis: Example

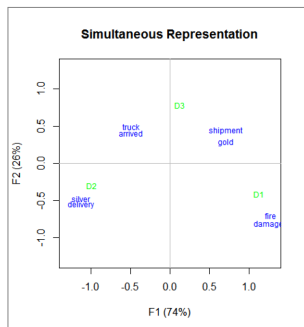
The projection is made possible thanks to the transition relationship (almost barycentric), meaning: The coordinates of a column term can be obtained based on all lines terms and vice versa.

Coordinates of term  $k$  on the 1st component:

$$F_{k1} = \frac{1}{\sqrt{\lambda_1}} \sum_{l=1}^L \frac{o_{kl}}{n_k} \times G_{l1}$$

Coordinates of document  $l$  on the 1st component:

$$G_{l1} = \frac{1}{\sqrt{\lambda_1}} \sum_{k=1}^K \frac{o_{kl}}{n_l} \times F_{k1}$$



### Remark

Each term is located based on all the documents (and vice versa). Thus, a cross-reading between documents and terms may be misleading ! For instance it would be wrong to conclude that D3 is linked to (truck) ... (see the  $\chi^2$  contribution table).

# Correspondence Analysis: Conclusions

- We have seen that the document-term matrix can be considered as a contingency table.
- The total quantity of information can be measured using the total inertia :  
$$\phi^2 = \frac{\chi^2}{N}.$$
- CA makes it possible to make a projection of the terms and their relationships (based on the documents).
- CA makes it possible to make a projection of the documents and their relationships (based on the terms).
- It also helps to visualize term–documents associations, while keeping in mind  $\chi^2$  results.
- CA relies on Singular Value Decomposition and the matrix of standardized residuals from the  $\chi^2$ .

# Outline

- 1 Introduction
- 2 Text mining tasks and processes
- 3 Analyzing similarities between and within texts
- 4 Word2vec**
- 5 Conclusion



# Machine Learning and text mining

- The text mining approaches we have seen so far are limited to statistical considerations.
- What if we want to do classification or clustering on texts ?

# Machine Learning and text mining

- The text mining approaches we have seen so far are limited to statistical considerations.
- What if we want to do classification or clustering on texts ?

Nowadays, among the many Machine Learning methods, **Deep Learning** is the hot and trendy technology for text analysis when dealing with clustering or classification.

## Word2Vec

Word2Vec is one of the most popular technique to learn word embeddings using *deep neural networks*. It was developed by Tomas Mikolov in 2013 at Google.

# Machine Learning and text mining

- The text mining approaches we have seen so far are limited to statistical considerations.
- What if we want to do classification or clustering on texts ?

Nowadays, among the many Machine Learning methods, **Deep Learning** is the hot and trendy technology for text analysis when dealing with clustering or classification.

## Word2Vec

Word2Vec is one of the most popular technique to learn word embeddings using *deep neural networks*. It was developed by Tomas Mikolov in 2013 at Google.

Question : How to input text data to machine learning algorithms ?

# Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?

# Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**

# Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**
  - Or it is very rare ...

# Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**
  - Or it is very rare ...
- What does Machine Learning use as inputs ?

# Text as input for Deep Learning

- Can text be used as a direct input for Machine Learning methods ?
  - **NO !**
  - Or it is very rare ...
- What does Machine Learning use as inputs ?
  - **Numbers !**

## Remark

Conveniently, with lemmatization and tokenization, we have seen a few techniques to turn text into vectors of numbers by counting words, or their frequencies, or IDF, or TF-IDF.

- We know how to encode texts and words into vectors !



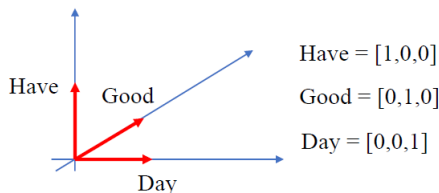
# The issue with trivial encoding

"Have a good day" and "Have a great day"  $\rightarrow V = \{\text{Have, good, great, day}\}$

## One-hot encoded vector:

Have = [1,0,0,0,0] ; a = [0,1,0,0,0] ; good = [0,0,1,0,0] ; great = [0,0,0,1,0] ; day = [0,0,0,0,1]

- If we try to visualize these encodings, we can think of a 5 dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest (no projection along the other dimensions).
- It means that "great" and "good" are as different as "day" and "have", which is not true !



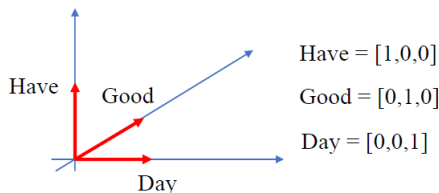
# The issue with trivial encoding

"Have a good day" and "Have a great day"  $\rightarrow V = \{\text{Have, good, great, day}\}$

## One-hot encoded vector:

Have = [1,0,0,0,0] ; a = [0,1,0,0,0] ; good = [0,0,1,0,0] ; great = [0,0,0,1,0] ; day = [0,0,0,0,1]

- If we try to visualize these encodings, we can think of a 5 dimensional space, where each word occupies one of the dimensions and has nothing to do with the rest (no projection along the other dimensions).
- It means that "great" and "good" are as different as "day" and "have", which is not true !



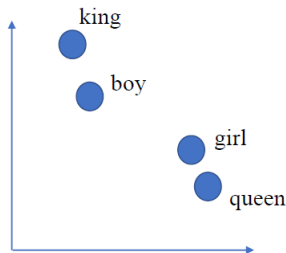
- This representation does not show similarities between words !
- Direct word counts or trivial embedding are therefore not a good encoding method as it loses a lot of information, in particular semantic information.

# Word Embedding

## Word Embedding: Definition

Word embedding consists in encoding words as dense vector which respect similarities

Unique word	encoding	Embedding
king	[1,0,0,0]	[1,2]
queen	[0,1,0,0]	[2,3]
girl	[0,0,1,0]	[1,5]
boy	[0,0,0,1]	[3,4]



Word2Vec is based on the notion of word embedding which considers similarities coming from neighbor words.

# Word2Vec: The Skip-Gram model

- We have seen that the notion of similarity may be important for word embedding.
- The question is : How to train a neural networks and teach it word similarities ?

# Word2Vec: The Skip-Gram model

- We have seen that the notion of similarity may be important for word embedding.
- The question is : How to train a neural networks and teach it word similarities ?
  - Feeding a large database of similarities is a bad idea !

# Word2Vec: The Skip-Gram model

- We have seen that the notion of similarity may be important for word embedding.
- The question is : How to train a neural networks and teach it word similarities ?
  - Feeding a large database of similarities is a bad idea !

## The skip-Gram model

The skip-gram model consists in feeding the neural networks with pairs of words that may be similar based on their close positions in the source text.

- The skip-Gram model uses window sizes :
  - Size 1 : only words in direct contact are paired
  - Size 2 : Words in direct contacts, or with one word between them are paired
  - etc.

# Word2Vec: The Skip-Gram model - Window size=1

“King brave man”

“Queen intelligent woman”

Word	neighbor
King	brave
brave	king
brave	man
man	brave
Queen	intelligent
intelligent	Queen
intelligent	woman
Woman	intelligent

# Word2Vec: The Skip-Gram model - Window size=2

“King brave man”

“Queen intelligent woman”

Word	neighbor
King	brave
King	man
brave	king
brave	man
man	brave
man	King
Queen	intelligent
Queen	woman
intelligent	Queen
intelligent	woman
Woman	intelligent
Woman	Queen



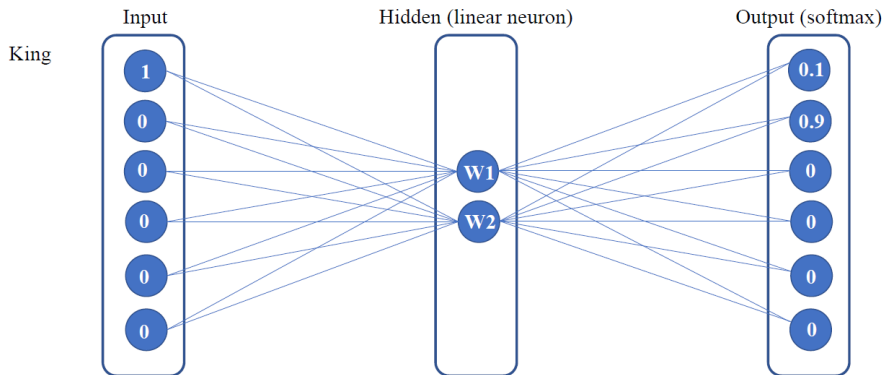
# Word2Vec: What we have - window size = 2

Word	Word one hot encoding	neighbor	Neighbor one hot encoding
King	[1,0,0,0,0,0]	brave	[0,1,0,0,0,0]
King	[1,0,0,0,0,0]	man	[0,0,1,0,0,0]
brave	[0,1,0,0,0,0]	king	[1,0,0,0,0,0]
brave	[0,1,0,0,0,0]	man	[0,0,1,0,0,0]
man	[0,0,1,0,0,0]	brave	[0,1,0,0,0,0]
man	[0,0,1,0,0,0]	King	[1,0,0,0,0,0]
Queen	[0,0,0,1,0,0]	intelligent	[0,0,0,0,1,0]
Queen	[0,0,0,1,0,0]	woman	[0,0,0,0,0,1]
intelligent	[0,0,0,0,1,0]	Queen	[0,0,0,1,0,0]
intelligent	[0,0,0,0,1,0]	woman	[0,0,0,0,0,1]
Woman	[0,0,0,0,0,1]	intelligent	[0,0,0,0,1,0]
Woman	[0,0,0,0,0,1]	Queen	[0,0,0,1,0,0]

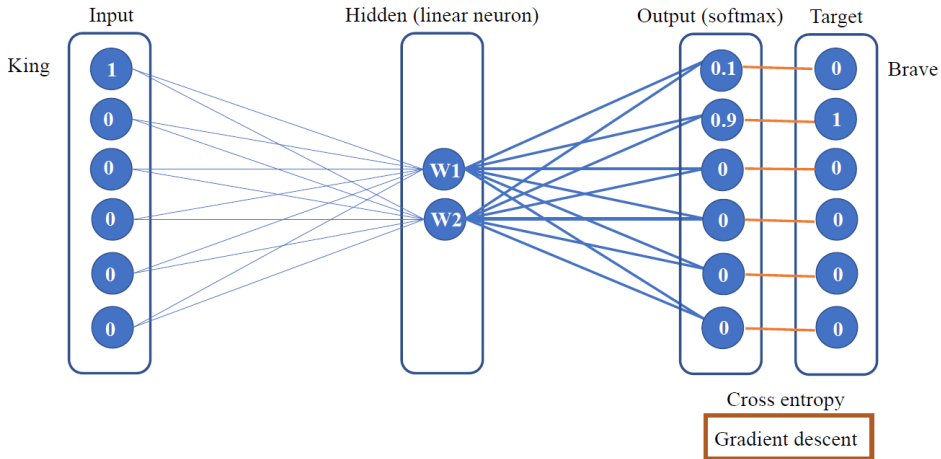
# Word2Vec: What we want - window size = 2

Input (Word one hot encoding)	Target (Neighbor one hot encoding)
[1,0,0,0,0,0]	[0,1,0,0,0,0]
[1,0,0,0,0,0]	[0,0,1,0,0,0]
[0,1,0,0,0,0]	[1,0,0,0,0,0]
[0,1,0,0,0,0]	[0,0,1,0,0,0]
[0,0,1,0,0,0]	[0,1,0,0,0,0]
[0,0,1,0,0,0]	[1,0,0,0,0,0]
[0,0,0,1,0,0]	[0,0,0,1,0]
[0,0,0,1,0,0]	[0,0,0,0,1]
[0,0,0,0,1,0]	[0,0,0,1,0,0]
[0,0,0,0,1,0]	[0,0,0,0,1]
[0,0,0,0,0,1]	[0,0,0,0,1,0]
[0,0,0,0,0,1]	[0,0,0,1,0,0]

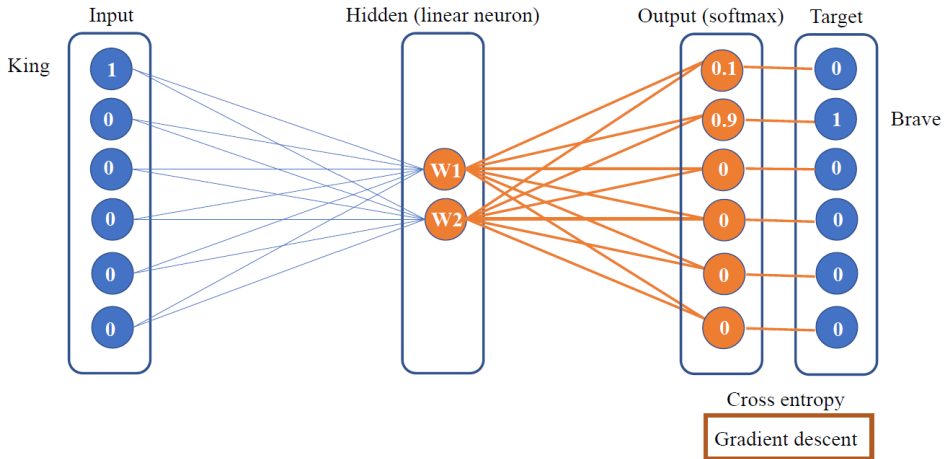
# Training Word2Vec (1/8)



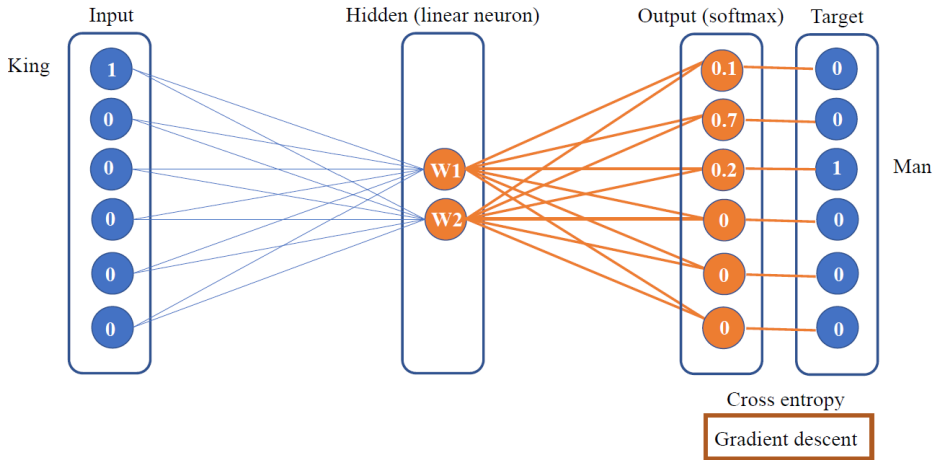
# Training Word2Vec (2/8)



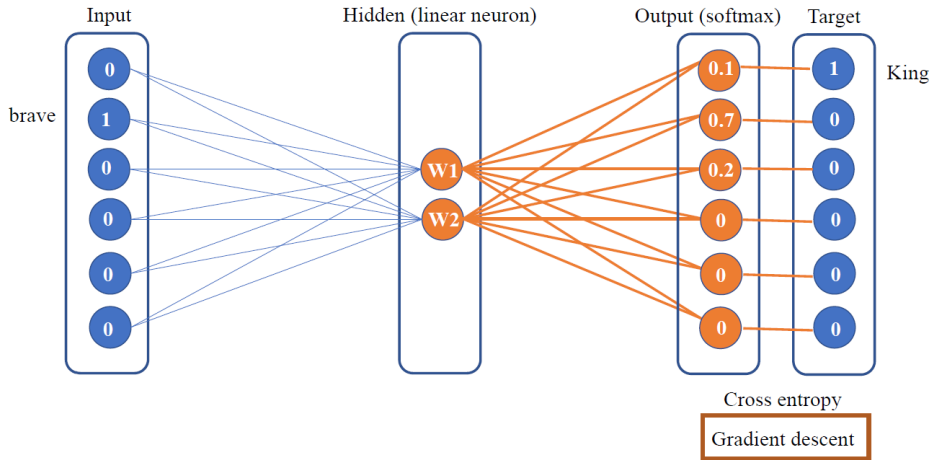
# Training Word2Vec (3/8)



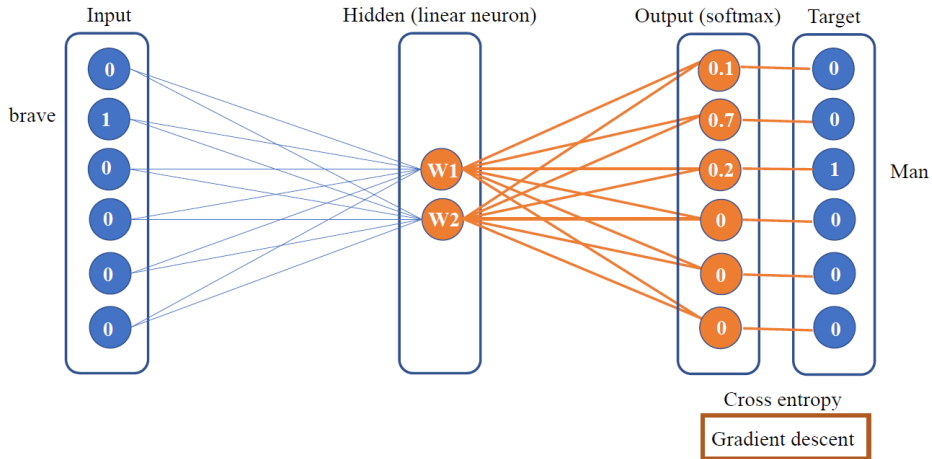
# Training Word2Vec (4/8)



# Training Word2Vec (5/8)

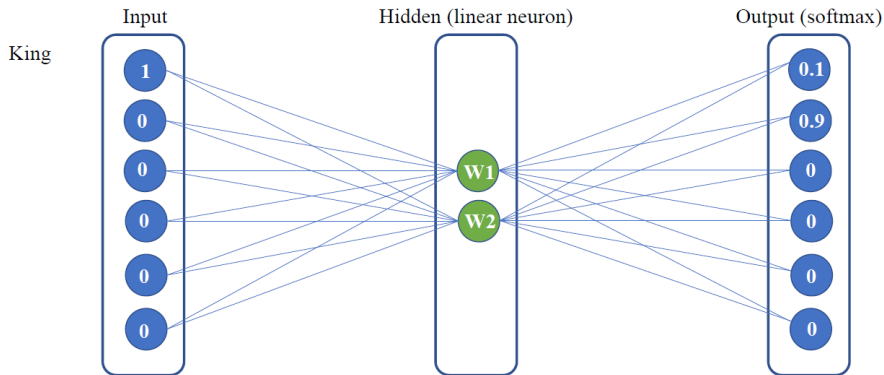


# Training Word2Vec (6/8)



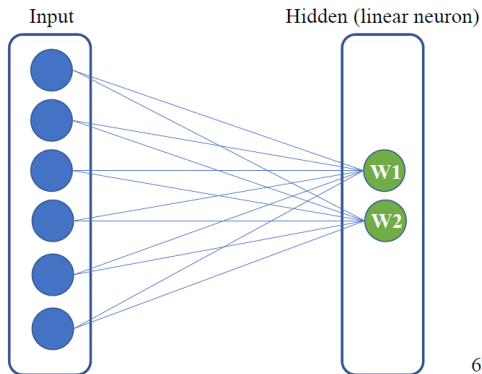


# Training Word2Vec (7/8)



Once the network is trained, the hidden layer can be used for Word2Vec embedding !

# Training Word2Vec (8/8)



Unique Word	Embedding
King	[1,1]
brave	[1,2]
man	[1,3]
Queen	[5,5]
intelligent	[5,4]
Woman	[5,6]

6-dimensional vector to two-dimensional vector

Once the network is trained, the hidden layer can be used for Word2Vec embedding !

# Word2Vec: Conclusion

- We have seen that Word2Vec is a neural network tool that can be used for smart word embedding that detects similarities and links between words.
- Word2Vec generated vectors are better inputs for Machine Learning Algorithms than regular "one-hot encoded" texts.

## On the importance of Parameters for Word2Vec

- The Window size for the N-Gram model matters:
  - Too small and it may miss some key links
  - Too big and the complexity explode and artificial links may be created
- The size of the hidden layer is important too: It determines the encoding size
  - Too small and the embedding compression will fail
  - Too big and it may not converge

# Outline

- 1 Introduction
- 2 Text mining tasks and processes
- 3 Analyzing similarities between and within texts
- 4 Word2vec
- 5 Conclusion**

## What we learned

We have seen some of the basics of text mining:

- How to turn text data into numerical data so that they can be processed by any Machine Learning algorithms
- We have studied various similarity and distances measures that can be used with text mining.
- We have seen two different methods that can link close words together in text corpuses: Correspondance Analysis and Word Embedding with Word2Vec.

# Bibliography

- <http://eric.univlyon2.fr/~ricco/ricco.html>
- Grossman D.A., Frieder O. "Information retrieval Algorithms and heuristics", Second Edition, Springer, 2004.
- Aggarwal C., Zhai C., "Mining Text Data", Springer, 2012.
- Lebart L., Salem A., "Statistique textuelle", Dunod , 1994.