TELECOM ParisTech

Une école de l'IMT

Lecture 2 - From text to feature vectors

SD-TSIA214

Chloé Clavel

université PARIS-SACLAY

Reminder

NLP tasks

2 kind of tasks:

- **Classify documents by themes, opinions etc...**
  - Supervised learning
    - Ex : SVM (support vector machines), Naive Bayes
  - Unsupervised learning
    - Ex: Clustering
- **Detect particular expressions**
    - Ex: Named Entities

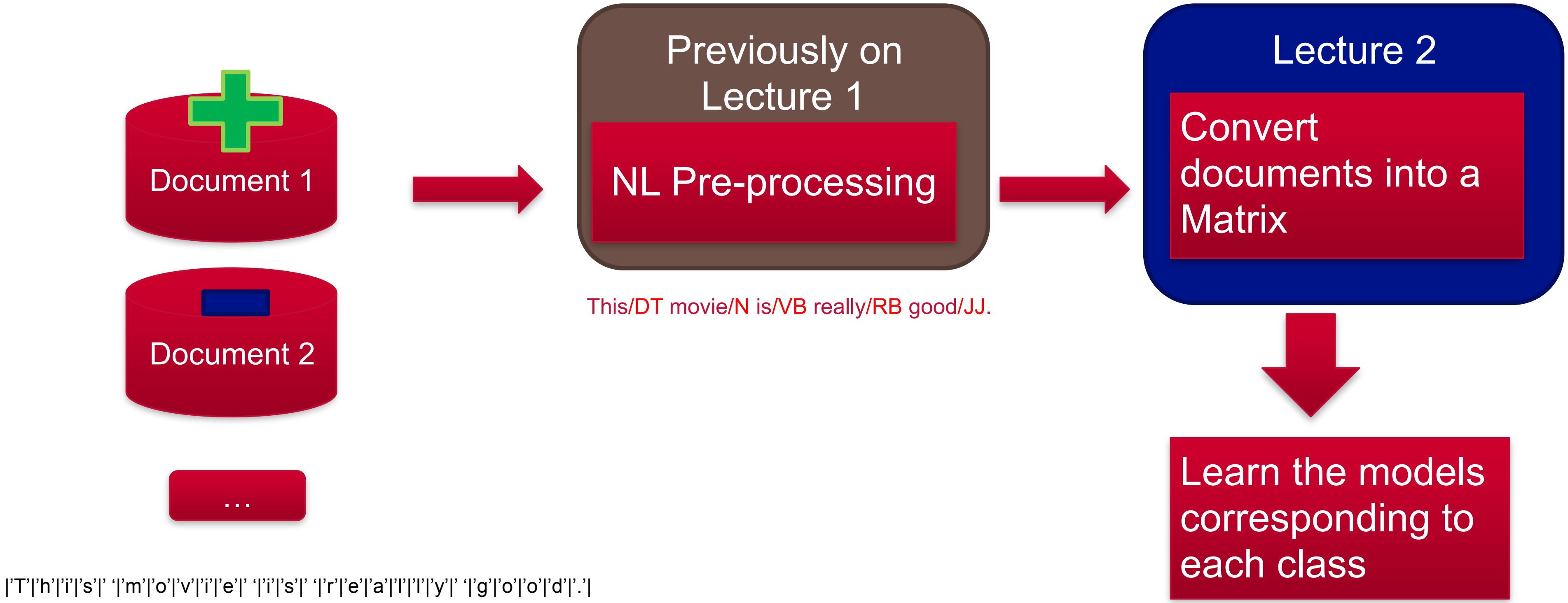
From http://www.tal.univ-paris3.fr/plurital/travaux-2009-2010/bao-2009-2010/MarjorieSeizou-AxelCourt/webservices.html

Classification

- **Phase 1 – learning**
  - Training corpus = set of documents annotated with opinions
    - Annotation : each document is assigned to a class :
      - Ex. Movie reviews: the score attributed by a user (1 to 5)
  - Goal : Learn from this corpus the specific features of each class
- **Phase 2 – classification**
  - Using the learned features, the system is able to assign a class to a new document

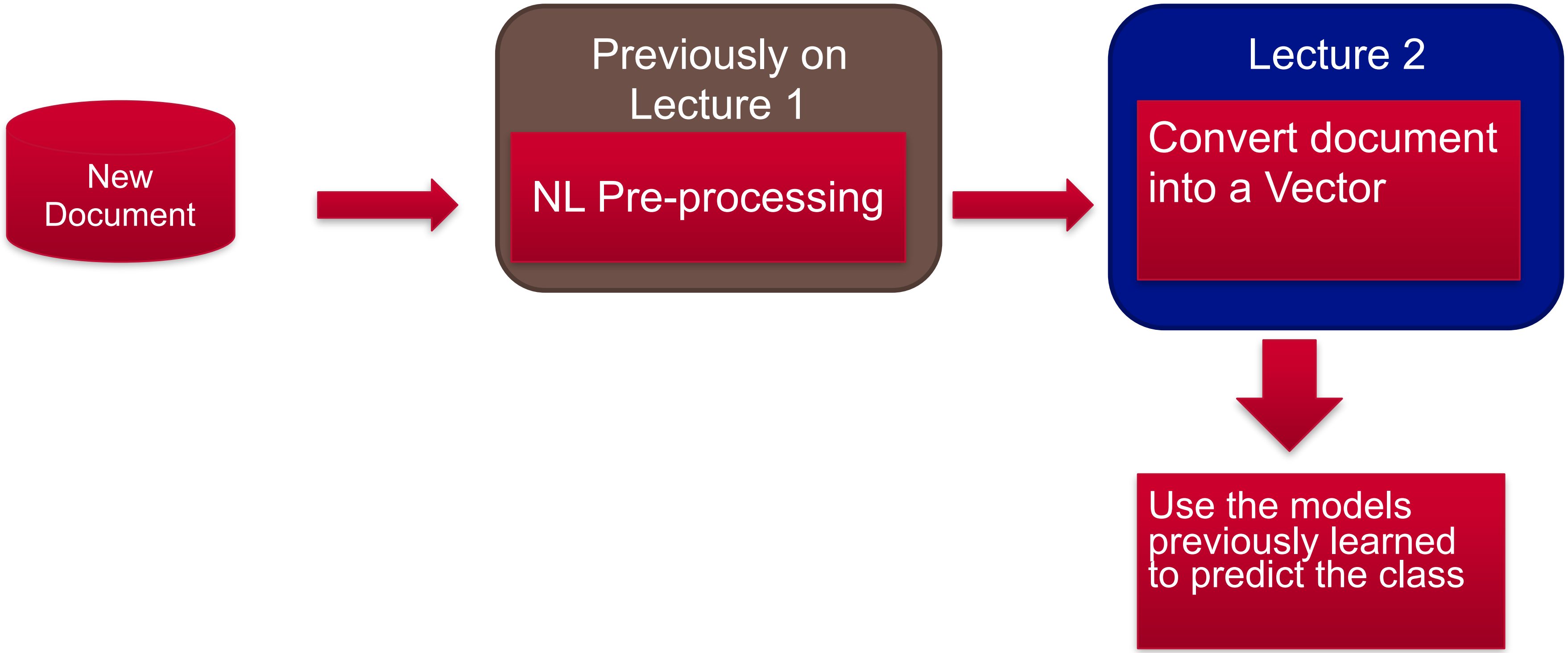
Phase 1 – learning

- **Learning the classes**



Phase 2 – classification

- **Predict the class of a new documents**



Objective of the lecture 2

- **Focus on**
  - text to vector transformation
- **Get familiar with:**
  - Classical transformations : TF-IDF
  - Embedded representations : word2vec

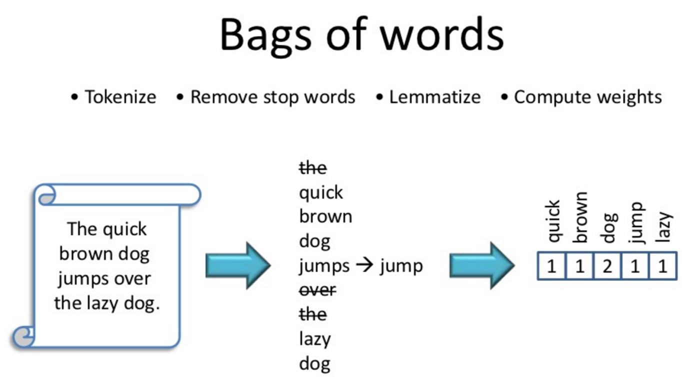
Levels of representations

- **PART 1 : representation at the document level**
  - One document = one vector
- **PART 2 : representation at the word level**
  - One word = one vector

PART 1 Document-based representation

Representation based on word frequencies

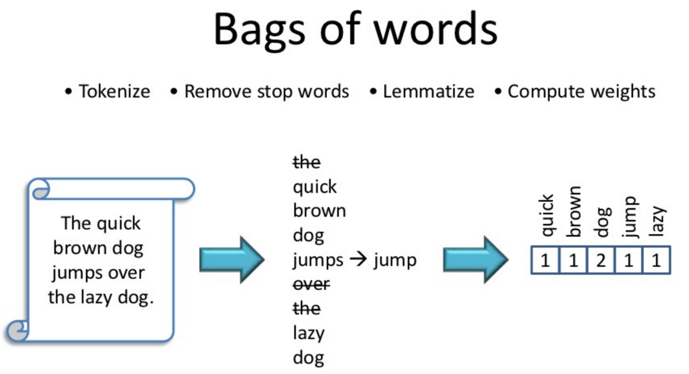
- **Bags of words (BOW) representation**
  - 1 document = 1 vector (a1, …., aN)
    - a_i = number of occurrences of the word w_i in document d



From Miha Grcar "Text mining and Text stream mining tutorial"

Representation based on word frequencies

- **Bags Of Words representation**
  - ALGO
    - From a set of M documents :
      - loop over the M documents and build a vocabulary (w1, …., wN)
      - N = vocabulary size
        - Remember that you can reduce the size of the vocabulary (see Lecture 1 on preprocessing)
      - Count the number occurrences of the word w_i in document d



From Miha Grcar "Text mining and Text stream mining tutorial"

Representation based on word frequencies

- Document set -> term-document matrix
  - Size : N x M

	call	time	date	conference	release	meeting	corporation	earnings
document 1	2	1	3	2	1	1	1	
document 2	1		2	1	2	1	1	1
document 5		1	2		2	1	1	1
document 6	1	2	1	1	3	1	1	1
document 7	1						1	
document 8			1		1		1	1
document 9	2		1	3	1	1	1	1
document 10	2	1		1	1		1	1
document 13					1			2
document 14							3	
document 15	1			2			1	2

From http://theses.ulaval.ca/archimede/fichiers/24972/ch05.html

Representation based on word frequencies

- **TF-IDF-based representation**
  - 1 document = 1 vector (a1, …., aN)
    - a_i = TF-IDF of the word w_i in document d
    - TF-IDF (Term Frequency - Inverse Document Frequency)
      - statistical measure used to evaluate the representativeness of a word for a particular document in a collection of documents

Representation based on word frequencies

- **TF-IDF-based representation**

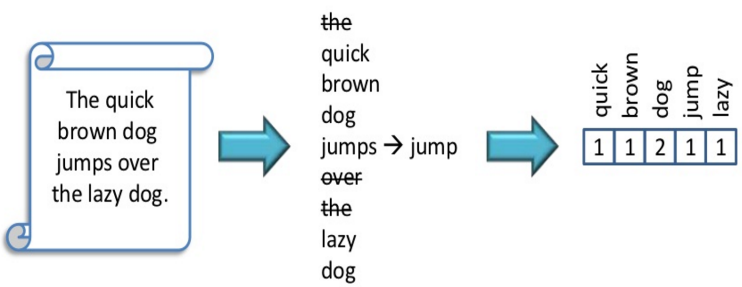
$$
\begin{aligned}
TFIDF(w,d) &= TF_{w,d}.IDF_{w,d} \\
&= TF_{w,d}.\left(\left(\log_2 \frac{M}{DF_w}\right)\right)
\end{aligned}
$$

M : number of documents
TF : Term Frequency
  Number of occurrences of w in d.
  Or boolean: tf(w,d) = 1 if w in d, 0 otherwise
DF : Document Frequency
  Number of documents with the word w
This value grows proportionally to the occurrences of the word in the document (TF) but its effect is countered by the occurrences of the word in every other document (IDF)

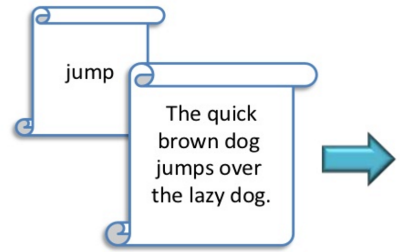
Representation based on word frequencies

Bags of words

- Tokenize
- Remove stop words
- Lemmatize
- Compute weights



Computing weights



	TF	DF	IDF	TFIDF
quick	1	1	0.69	0.69
brown	1	1	0.69	0.69
dog	2	1	0.69	1.39
jump	1	2	0	0
lazy	1	1	0.69	0.69

$$TFIDF = TF \times IDF$$

$$IDF = \log_e \frac{|\mathrm{D}|}{DF}$$

$$|\mathrm{D}| = 2$$

Representation based on word frequencies

- **PRACTICE 1 :** calculate the TF-IDF of the word “director“ for the document d :
  TF-IDF(« director », d) = ?
  - The database contains 1000 documents
  - The document d contains 3 times the word “director”
  - 70 texts contain the word “director“
  - « director » occurs 134 times in the database

$$TFIDF(w,d) = TF_{w,d}.IDF_{w,d}$$
$$= TF_{w,d}.\left(\left(\log_2 \frac{M}{DF_w}\right)\right)$$

M : number of documents
TF : Term Frequency
Number of occurrences of w in d.
Or boolean: tf(w,d) = 1 if w in d, 0 otherwise
DF : Document Frequency
Number of documents with the word w

Representation based on word frequencies

- **PRACTICE 1 :** calculate the TF-IDF of the word “director“ for the document d :
  TF-IDF(« director », d) = ?
  - The database contains 1000 documents
  - The document d contains 3 times the word “director”
  - 70 texts contain the word “director“
  - « director » occurs 134 times in the database

$$3.\left(\log_2 \frac{1000}{70}\right) = 11{,}5$$

Representation based on word frequencies

- **PRACTICE 2 :** calculate the TF-IDF of the word “director“ for the document d :
  TF-IDF(« director », d) = ?
  - The database contains 1000 documents
  - The document d contains 3 times the word “director”
  - 900 documents contain the word “director“
  - « director » occurs 1014 times in the database

$$TFIDF(w,d) = TF_{w,d}.IDF_{w,d}$$
$$= TF_{w,d}.\left(\left(\log_2 \frac{M}{DF_w}\right)\right)$$

M : number of documents
TF : Term Frequency
Number of occurrences of w in d.
Or boolean: tf(w,d) = 1 if w in d, 0 otherwise
DF : Document Frequency
Number of documents with the word w

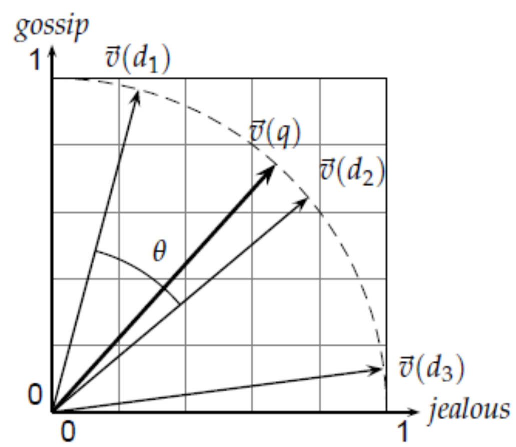
Representation based on word frequencies

- **PRACTICE 2 :** calculate the TF-IDF of the word “director“ for the document d :
  TF-IDF(« director », d) = ?
  - The database contains 1000 documents
  - The document d contains 3 times the word “director”
  - 900 documents contain the word “director“
  - « director » occurs 1014 times in the database

$$3.\left(\log_2 \frac{1000}{900}\right) = 0.45$$

Document-based representation

- In the vector space
  - A set of documents corresponds to a set of vectors in the vector space
  - Vector space: 1 axis per vocabulary term



▶ **Figure 6.10** Cosine similarity illustrated. $\mathrm{sim}(d_1, d_2) = \cos\theta$.

Representation based on word frequencies

- **Drawbacks of Bags of words representations**
  - The term-document matrix scale for big database
    - loop over the document set
  -

	call	time	date	conference	release	meeting	corporation	earnings
document 1	2	1	3	2	1	1	1	
document 2	1		2	1	2	1	1	1
document 5		1	2		2	1	1	1
document 6	1	2	1	1	3	1	1	1
document 7	1						1	
document 8			1		1		1	1
document 9	2		1	3	1	1	1	1
document 10	2	1		1	1		1	1
document 13					1			2
document 14							3	
document 15	1			2			1	2

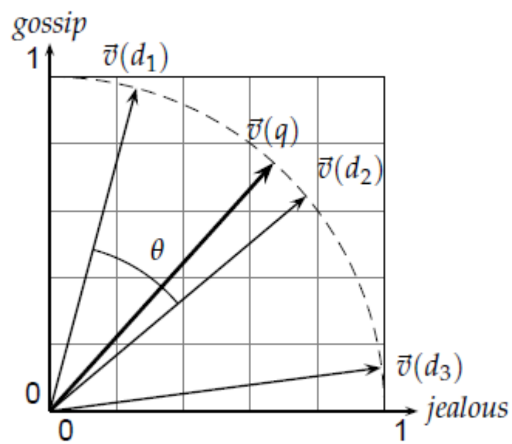
Representation based on word frequencies

- **Drawbacks of Bags of words representations**
  - No capture of the order of the terms in the document

Ex: These two sentences are represented by the same vector
"Mary is quicker than John"
"John is quicker than Mary"

Measuring the similarity btw. two documents

- **Cosine similarity**
  - Similarity between 2 vectors of doc d1 and d2 according to the cosine of the angle



► **Figure 6.10** Cosine similarity illustrated. $\mathrm{sim}(d_1, d_2) = \cos\theta$.

$$\mathrm{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|},$$

Word-based representation

References

- https://nlp.stanford.edu/IR-book
- From Miha Grcar "Text mining and Text stream mining tutorial"
-