Une école de l'IMT

Opinion Analysis

Chloé Clavel, enseignante et chercheuse
à Telecom-Paris

Introduction

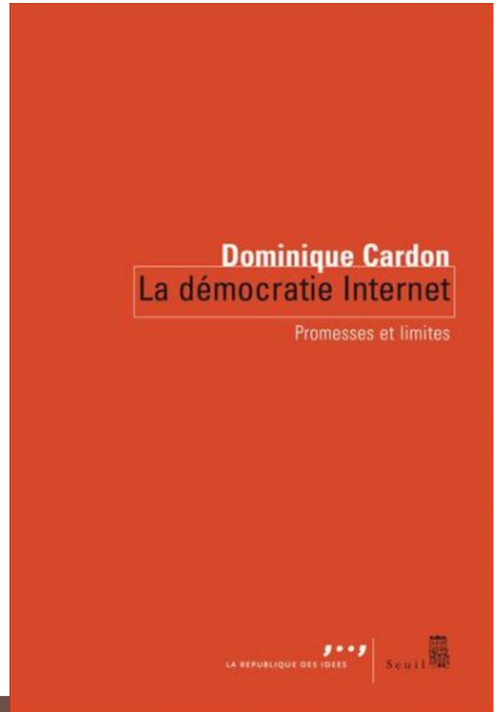
Introduction

- **Different terminologies**
  - *Opinion extraction, opinion mining, sentiment analysis, subjectivity analysis, affect sensing, emotion detection*
- **Applications**
  - Social Network analysis
  - Human-agent interaction : ex: chatbot

Social data and opinion analysis

- **Social data:**
  - Expressions of the citizens on the web
- **Context :**
  - Renewal of opportunities for criticism and action via the Internet

Lecture : « La démocratie Internet »
Dominique Cardon



Social data and opinion analysis

- **Challenges**
  - Analysis of societal trends
  - Analysis of citizens' opinions on candidates in elections
  - Review of movie reviews (movie reviews)
  - Analysis of the opinions of Internet users on a product
  - Analysis of the e-reputation of a brand, a product
  - Identify target clients / referral systems
  - Evaluate the success of communication campaign

Social Data and opinion analysis

- **Disciplines :**
  - Sociology:
    - qualitative / manual / sociological analysis of small corpora selected to form a panel of studies
  - Computer science :
    - development of automatic large corpus analysis methods

Human-agent/robot interaction

- **Robotics and artificial agents**
  - Analyze and reproduce human behaviors to interact socially with humans. Animated conversational agents,
  - Robots & "emotional avatar"

[GRETA Platform, Pelachaud]

[Softbank robotics]

Human-agent interaction

- **Virtual assistant**

https://www.youtube.com/watch?v=TaY9zt_qx_c

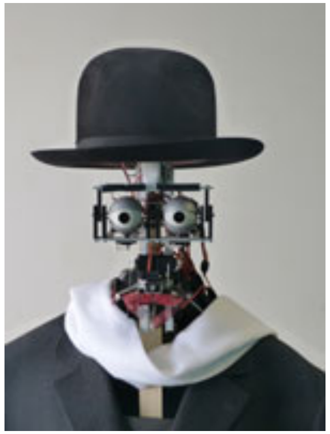
Human-machine interaction

- **Kirobo: the Japanese robot who left 18 months in space to keep an astronaut company**

Interaction humain-agent: LiveChat et relation client

Human-robot interaction

- **Berenson robot at Quai Branly**
  - "Visitors were invited to observe and interact with Berenson's behavior, helping to define the criteria for aesthetic appreciation of this amateur art robot. "



At the end of the course...

- You will master the main linguistic issues for NLP and sentiment analysis
- You will be able to describe and implement the different methods for text representation into vectors
- You will be able to build a text classification framework

Today program

- Opinion detection – task description
- A simple method lexical affinity – Naive Bayes classifier
- How to obtain labelled data?
- Representation based on word frequencies

Next lecture's program

- Tokenization for obtaining vocabulary space
- Deep learning approaches
- Introducing knowledge in sentiment analysis method – overview
- Preprocessing using linguistic knowledge for machine learning
- Knowledge-based methods for opinion analysis

Lab : Knowledge-based approaches for sentiment analysis on twitter

Opinion detection – Task description and challenges

Task description and challenges

EXO: Is the review positive or negative? Highlight the expressions corresponding to the expression of an opinion. Do they seem positive or negative in general?

- **“This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can’t hold up.”**
- **« Well as usual Keanu Reeves is nothing special, but surprisingly, the very talented Laurence Fishbourne is not so good either, I was surprised. »**

Task description and challenges

- "This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it can't hold up."
- Well as usual Keanu Reeves is nothing special, but surprisingly, the very talented Laurence Fishbourne is not so good either, I was surprised.

Task description and challenges

- more complex than a simple positive vs. negative word counts.
  - conditional tense
  - discourse markers
  - negation processing (I don't like this movie)
  - modifiers and intensifiers (the plot is not very good)
  - dealing with metaphors (global warming vs. climate change [Ahmad et al., 2011])

Task description and challenges

- **Identification of opinion target**
  - « Je suis satisfait des contacts que j'ai eus avec le service client mais pas des tarifs pratiqués »
  - Detected concepts
    - Opinion : satisfaction
    - Topics: contact et prix
  - Challenge :
    - Target analysis: be able to automatically detect what the opinion is about
    - Anaphora solving : "il les adore"

Opinion analysis : two tasks

- Task 1 : Classification of documents in opinion categories

New Document

Classe : positive or negative

Use models previously learned to predict the class

- Variants :
  - Binary classification (positive, negative)
  - Multi-class classification (fear, anger, sadness)
  - Multi-label classification (positive, joy)

Opinion analysis : two tasks

- Task 2 : Sequential annotation of opinion, source and target using sequential approaches

The	committee	,	as	usual	,	has
O	O	O	B_ESE	I_ESE	O	B_DSE
refused	to	make	any	statements	.	
I_DSE	I_DSE	I_DSE	I_DSE	I_DSE	O	

Figure 2: from (Irsoi and Cardie)

Overview of opinion analysis methods

Classification/sequential annotation Methods

- **By hand : the sentiment class is attributed by a human**
  - E.g. Yahoo in the old days
    - Very accurate and consistent assuming experts
    - ✗ Super slow, expensive, does not scale

Classification/sequential annotation Methods

- **Rule-based**
  - Linguistic/syntactic patterns ~ Advanced search criteria using advanced regular expressions

(manque|~negation-patt|(il/#NEG/y/avoir/~negation-patt))/(#PREP_DE)?/ (conseil|contact|~services-lex)*

  - Accuracy high if rule is suitable
  - ✗ Need to manually build and maintain rule-based system.

Classification/Sequential annotation Methods

- **Machine learning**
  - **Classification :**
    - Multi-Layer Perceptron, Support Vector Machine,
  - **Sequential annotation**
    - Conditional Random Fields, recurrent neural networks, Hidden-Markov Model, etc.

    - Scales well, can be very accurate, automatic
    - ✗ Requires classified training data. Sometimes a lot!

ML vs. Rule-based

- ML advantages
  - few linguistic expertise is required to build the model from the annotated data,
  - a higher interoperability of the models
- ML drawback
  - require a labelled dataset (big dataset for deep learning approaches) while:
    - **annotating data in opinions is a difficult task**
  - difficult interpretation of trained models
  - difficult to transfer model on different data (the model is corpus-dependent)

The simplest ML approach : Lexical affinity

The simplest ML approach : Lexical affinity

- Assign to the different words a probability of belonging to a category of opinion (« probabilistic affinity »)
  - Ex : « réchauffement » is considered the negative class with a probability of 75%
  - 'accident' 75% probability of being indicating a negative effect, [Cambria & White (2014)]
    - as in 'car accident' or 'hurt by accident'
    - but not in "I met him by accident"
  - These probabilities are learned on annotated corpora (for example using Naive Bayes classification method)

Naive Bayes Classifier for text

- **Classification Principle**
  - Choose the class c maximizing
    - Given an observation o = document

$$\hat{c} = \arg\max_{c} P(c \mid o)$$

    - Bayes rule + the fact that P(o) is independent from the class =>

$$\hat{c} = \arg\max_{c} P(c \mid o) = \arg\max_{c} \frac{P(o \mid c)P(c)}{P(o)} = \arg\max_{c} P(o \mid c)P(c)$$

Naive Bayes Classifier for text

$$\hat{c} = \arg\max_{c} P(c \mid o) = \arg\max_{c} \frac{P(o \mid c)P(c)}{P(o)} = \arg\max_{c} P(o \mid c)P(c)$$

- Naive : assumptions of strong independance between the features
  - o=doc and (m1,…mN) the words of document o
  - P(o|c)=p(m1,…mN/c)=∏p(mi/c) -> use the log

$$\hat{c} = \arg\max_{c \in \mathbb{R}} [log(P(c)) + \sum_{i=1}^{N} log(P(m_i/c))]$$

Naive Bayes Classifier for text

$$\hat{c} = \arg\max_{c \in \mathbb{R}} [log(P(c)) + \sum_{i=1}^{N} log(P(m_i/c))]$$

- Training on the labelled database
  - Estimating p(c) and p(mi/c)
    - p(c) = number of doc. in class c / total number of doc.
    - p(mi/c) = frequency of the word mi in class c

TP Sentiment analysis avec NB

```
TrainMultinomialNB(ℂ, 𝔻)
V ← ExtractVocabulary(𝔻)
N ← CountDocs(𝔻)
for each c ∈ ℂ
do N_c ← CountDocsInClass(𝔻, c)
    prior[c] ← N_c / N
    text_c ← ConcatenateTextOfAllDocsInClass(𝔻, c)
    for each t ∈ V
    do T_ct ← CountTokensOfTerm(text_c, t)
    for each t ∈ V
    do condprob[t][c] ← (T_ct + 1) / Σ_t'(T_ct' + 1)
return V, prior, condprob

ApplyMultinomialNB(ℂ, V, prior, condprob, d)
W ← ExtractTokensFromDoc(V, d)
for each c ∈ ℂ
do score[c] ← log prior[c]
    for each t ∈ W
    do score[c] += log condprob[t][c]
return arg max_{c∈ℂ} score[c]
```

p(mi/c)
+ Laplace smoothing

▶ **Figure 13.2** Naive Bayes algorithm (multinomial model): Training and testing.

The simplest ML approach : Lexical affinity

- **Limits :**
  - Operates at the level of the word and not at the level of the sentence
    - does not deal with negation or the semantic context
    - Ex from [Moilanen 2007]

« The senators supporting(+) the leader(+) failed(-) to praise(+) his hopeless(-) HIV(-) prevention program.”

  - The probabilities learned depend strongly on the corpus used for learning the probabilities (and thus on its domain)
    - Ex : “the power of the graphism” (for video games)
    - Vs. “the contract power” (for electricity)

Some advices to build an opinion database

How to build a labelled database?

- **The two steps**
  - **Data collection**
  - **Data annotation**

Data collection

- **Importance of data**
  - Quality of models learned <= sufficient quantity and quality of data
  - Collect data close to the intended application
  - Sometimes difficult, *eg* fear

Data collection

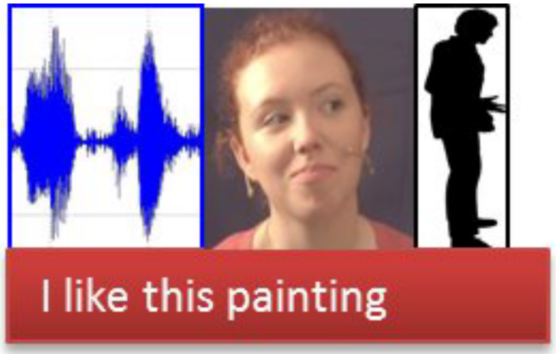
- **Different types of sources**
  - Open data available on the web and on social networks
  - Data collected within companies
    - Ex : call-center transcripts, chatbot interactions, complaint emails

Opinion annotation : challenges

- **Subjective phenomenon**
  - We do not all have the same perception of an opinion expressed by the other
- **Complex phenomenon:**
  - Opinions and emotions and other phenomena are often mixed in natural interactions (*e.g.* fear and anger)

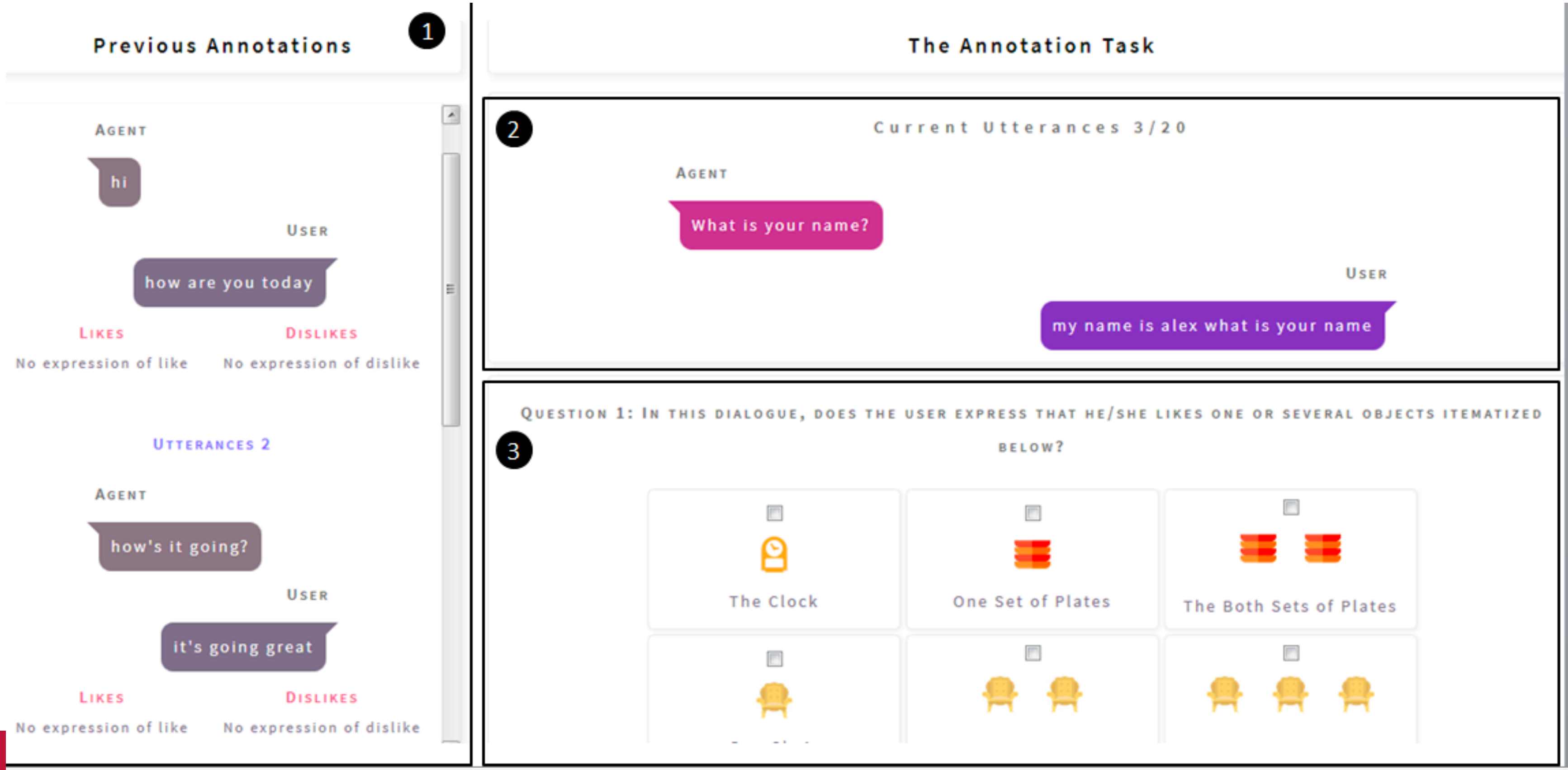
Opinion annotation : challenges

- Perception depends on different types of contexts :
  - Situation
    - ex: societal context
  - Speakers (gender, name, age, role, etc.)
  - Different modalities of expression
    - Verbal content, prosody, Gesture, Posture, Facial expressions, physiological signal



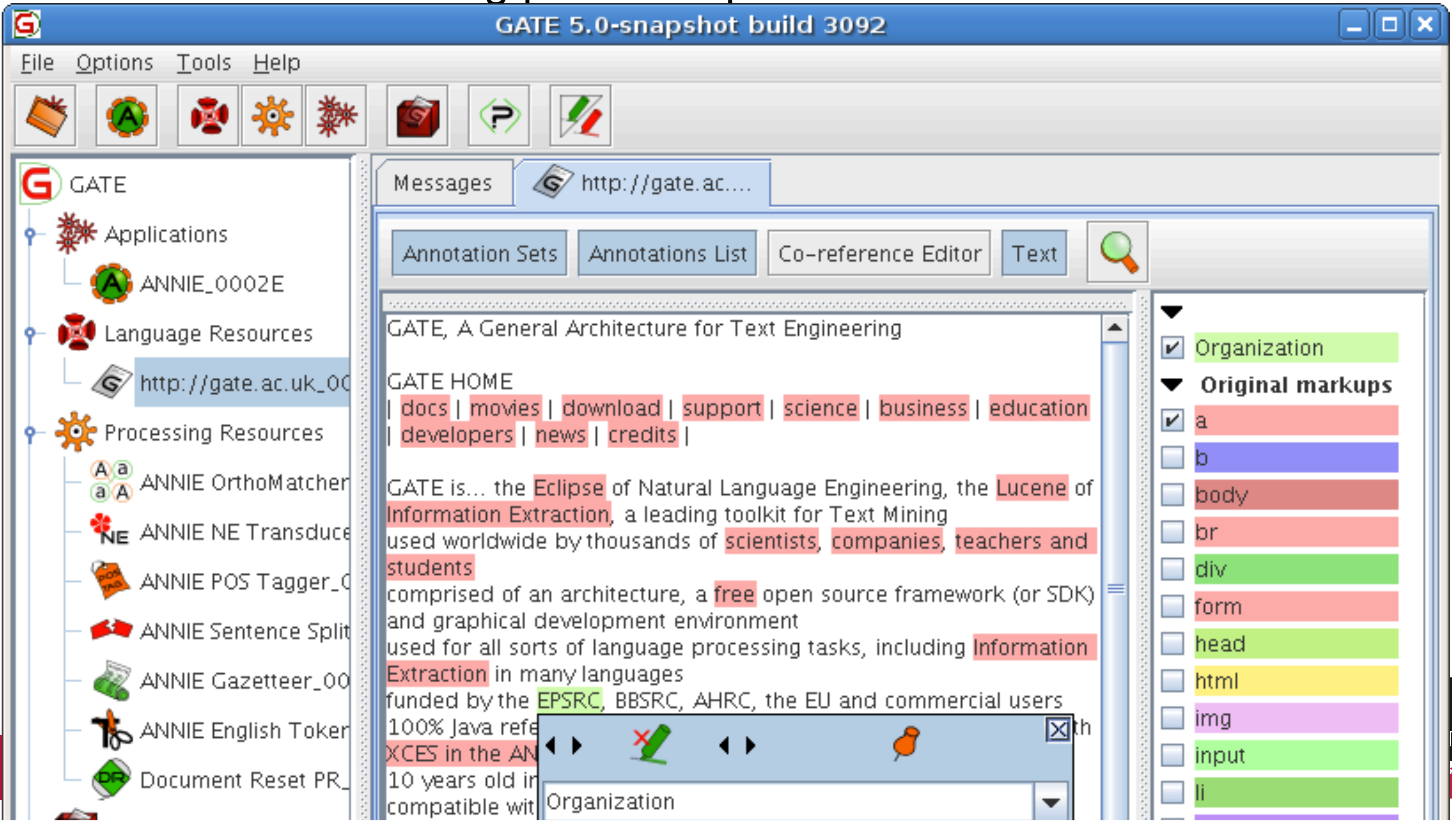
Building an annotation schema

- Perception depends on the context => Annotation of the context
  - Ex : annotation of the context of an interaction
- Use of questions to guide the annotation process



Opinion annotation

- **Use Annotation tools and crowdsourcing platforms**
  - Ex of annotation tool : Gate
  - Ex of crowdsourcing platform : prolific.ac



Measure the reliability of annotations

- **Opinion/Emotion phenomenon = subjective phenomenon**
  - Annotate multiple annotators
  - Assess the degree of reliability of the annotations

Measure the reliability of annotations

- **Measures**
  - Cohen's kappa [Carletta, 1996]:
    - agreement corrected for what it would be under the mere fact of chance

$$\kappa = \frac{\bar{p_o} - \bar{p_e}}{1 - \bar{p_e}}$$

    - Po is the proportion of agreement observed and Pe the probability that the annotators agree by chance

Mesure de la fiabilité des annotations

EXO

$$\kappa = \frac{\bar{p}_o - \bar{p}_e}{1 - \bar{p}_e}$$

- **Kappa values ?**
  - When annotators agree as much as chance
  - When the annotators agree totally
- **Exercice**
  - 50 text sequences annotated by 2 people (Ann1 / Ann2) in 2 categories positive / negative
  - Calculating kappa between the two annotators

Ann1\Ann2	Positive	Negative
Positive	20	5
Negative	10	15

Measure the reliability of annotations $\kappa = \frac{\bar{p}_o - \bar{p}_e}{1 - \bar{p}_e}$

- **Po = (20+15)/50 = 0,7**
- **Calculate Pe:**
  - Ann1 uses positive label 50% of the time
  - Ann2 uses positive label 60% of the time
  - Probability that Ann1 and Ann2 use the positive label: 0.5*0.6=0.3
  - Probability that Ann1 and Ann2 use the negative label : 0.5*0.4 = 0.2
  - Probability to agree by chance : 0.2+0.3 = 0.5
- **Kappa computation:**
  - Kappa = 0.2/0.5 = 0.4

Measure the reliability of annotations

- **Moderate agreement = standard for emotions [Landis et Koch, 1977]**

Accord	Kappa
Excellent	$\geq 0,81$
Bon	0,80-0,61
Modéré	0,60-0,41
Médiocre	0,40-0,21
Mauvais	0,20-0
Très mauvais	< 0

TAB. 4.4 – *Degré d'accord en fonction des valeurs de Kappa*

- **Other measure: Cronbach's Alpha [Cronbach, 1951] for dimensions**

Evaluation of opinion detection systems

Performances and evaluation

- Performances depend on :
  - The type of task :
    - Ex: target identification, polarity classification
  - number and type of classes
    - positive vs. Negative
    - Multi-class
    - Fear vs. anger more subtle than fear vs. Joy
  - the train/test corpus (diversity of data)

Performances and evaluation

- Provide comparison of Human vs. system performance
- See evaluation campaings :
  - Semeval : series of evaluations of computational semantic analysis systems including sentiment analysis tasks

Evaluation scores for classification systems

- **In the task of correct assignment to class c**
  - R = Recall : (number of system's correct assignments to class c) / (number of documents labelled c)
    - A system that tends to infrequently assign class c (high system *silence* for class c) will have a low recall

Evaluation scores for classification systems

- **In the task of correct assignment to class c**
  - P = Precision : (number of system's correct assignments to class c) / (number of system's assignments to class c)
    - A system that tends to allocate class c too frequently (system *noise* is high for class c) will have a low precision

Evaluation scores for classification systems

- **In the task of correct assignment to class c**
  - F-score : harmonic mean between recall and precision $= 2\times(P \times R) / (P + R)$

- Multiclass : average over the classes

Opinion analysis in interactions

Representation based on word frequencies

Phase 1 – learning

- **Learning the classes**

	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

Story of text representation

- **Historical representations : ex :TF-IDF**

	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

Based on word frequencies in the document
Still used in some NLP tasks

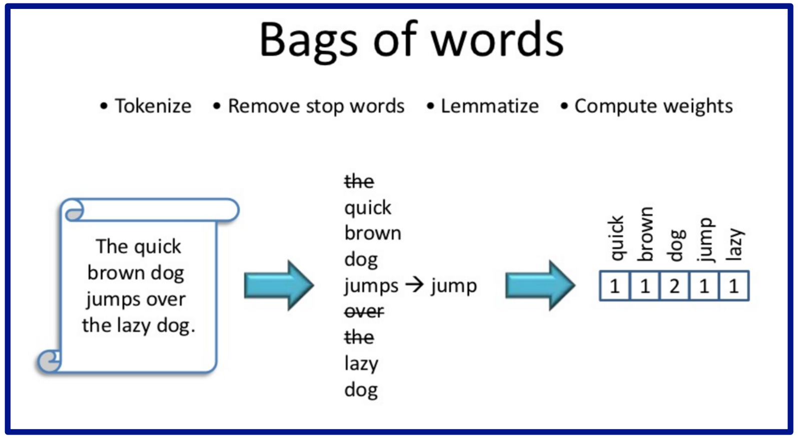
- **Representations currently used : word embeddings**

(Mikolov et al., NAACL HLT, 2013)

Based on the word context of occurrences, representing the semantic content of words
Initially used for language modelling

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**

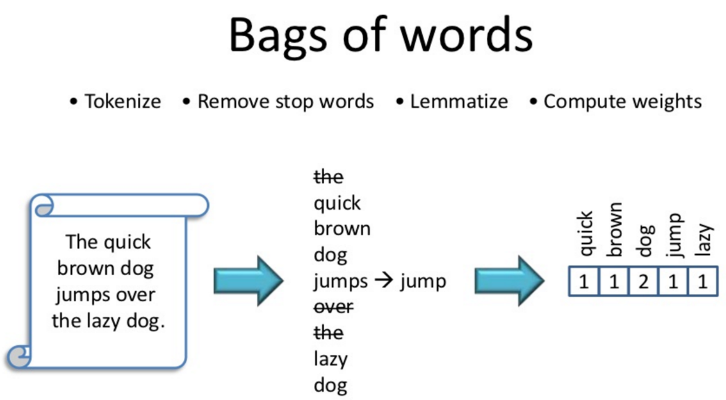


From Miha Grcar "Text mining and Text stream mining tutorial"

Note: such methods require a first step of tokenization

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**

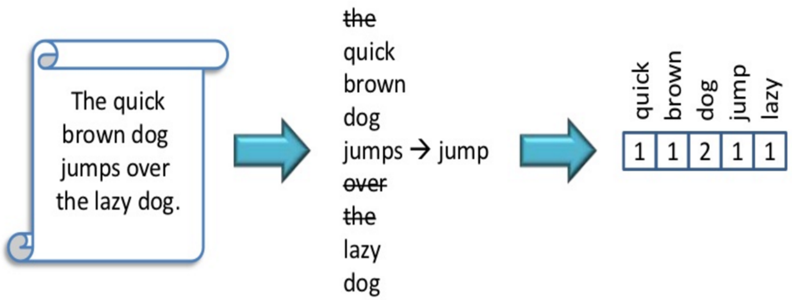
$$TFIDF(w,d) = TF_{w,d}.IDF_{w,d}$$
$$= TF_{w,d}.\left(\left(\log_2 \frac{M}{DF_w}\right)\right)$$

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
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)

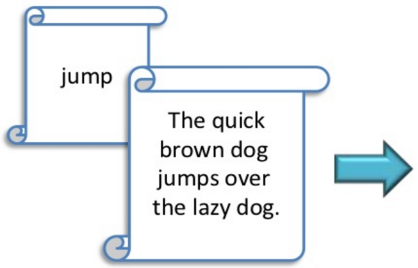
Representation based on word frequencies

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**

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

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

Representation based on word frequencies

- **PRACTICE 2 :** calculate the TF-IDF of the word "director" for the document d :

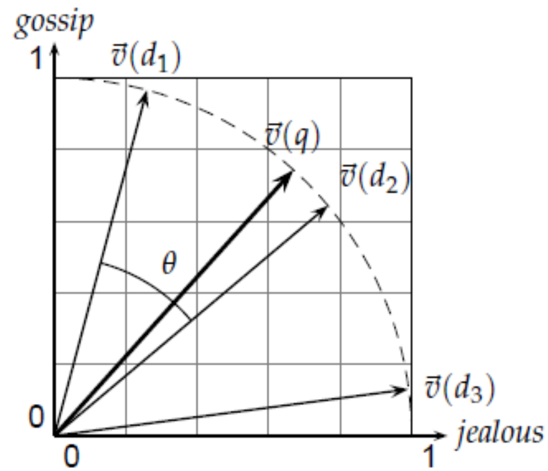
  TF-IDF(« director », d) = ?

  - 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

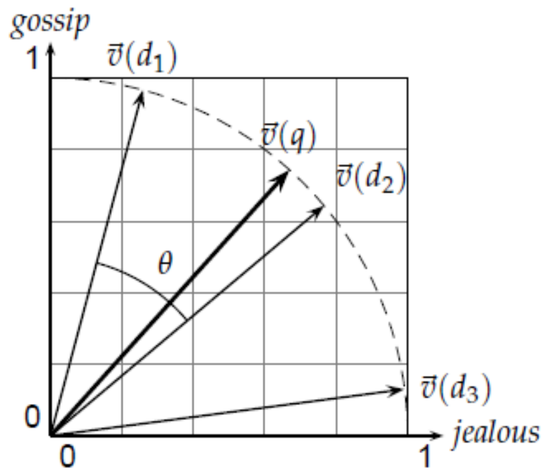
- In the vector space
  - A set of documents corresponds to a set of vectors in the vector space
  - Vector space: 1 axis per vocabulary term



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

Measuring the similarity btw. two documents

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



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

► **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**

	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"