RITAL

Information retrieval and natural language processing
Recherche d'information et traitement automatique de la langue

Master 1 DAC, semestre 2

Nicolas Thome

Bag of Words (BOW) for document classification (2)

Recap: Processing Chain for Document Classification

1. Preprocessing

- encoding (latin, utf8, ...)
- ponctuation
- stemming
- lemmatization
- tokenization
- capitals/lower case
- regex
- ...

Trade-off between:

- Accurate BoW representation (expressive vocab, N-gram, etc)
- Fighting overfitting: limit dimension explosion

2. BoW Model

- Dictionary
- Vectorial format
- TF-IDF
- Binary/non-binary
- N-grams

3. Learning

- Doc / sentence / paragraph classification
- Linear/non-linear classifiers?
- Naive Bayes, logistic regression, SVM

4. Hyper-parameter optimization

http://sametmax.com/lencoding-en-python-une-bonne-fois-pour-toute/

- Sur le disque, les fichiers sont encodés de manière spécifique...
- En python, les strings sont encodées de manière spécifique...

Pre-processing: encoding

http://sametmax.com/lencoding-en-python-une-bonne-fois-pour-toute/

- Sur le disque, les fichiers sont encodés de manière spécifique...
- En python, les strings sont encodées de manière spécifique...
- L'ouverture des fichiers est souvent associée à un encodage !!!

$\Rightarrow$ Comment gérer cela?

`http://sametmax.com/lencoding-en-python-une-bonne-fois-pour-toute/`

- Sur le disque, les fichiers sont encodés de manière spécifique...
- En python, les strings sont encodées de manière spécifique...
- L'ouverture des fichiers est souvent associée à un encodage !!!

$\Rightarrow$ Comment gérer cela?

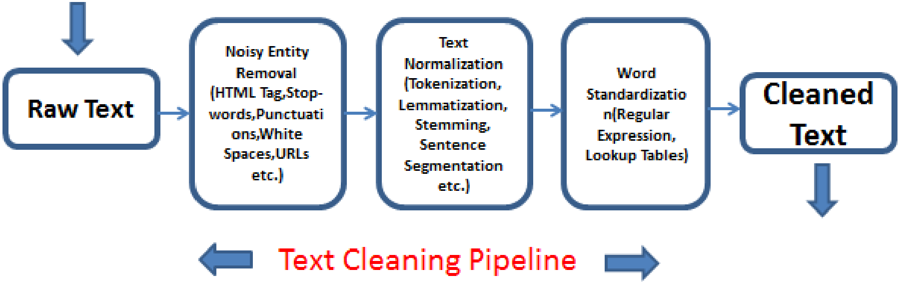
Solution 1

1. Ouverture en binaire des fichiers (e.g. en python)
2. Conversion des strings depuis un encodage connu
   `str.decode('utf8')`
   `unicodedata`, `unidecode`

Solution 2

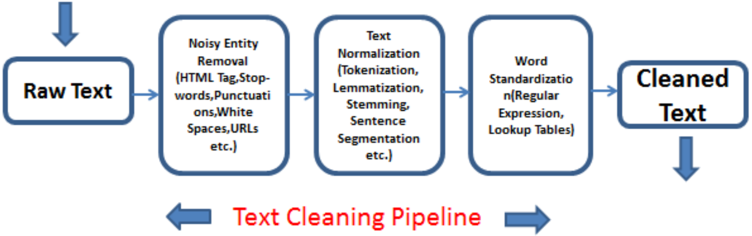
1. Vérifier le type d'encodage + convertir avant

- Some input token: noise to the classification tasks.
  - Noise: task-dependent, and on the training dataset size / robustness to overfitting



Pre-processing: noisy entity removal

- Ponctuation, capitals/lower case: remove or keep it?
- Stop word: empty meaning word, *e.g.* "the"
  - Use pre-defined "black list" (nltk) or upper frequency bound on target corpus
- Removing rare words (occurring less than a threshold)



Lemmatization:

in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form

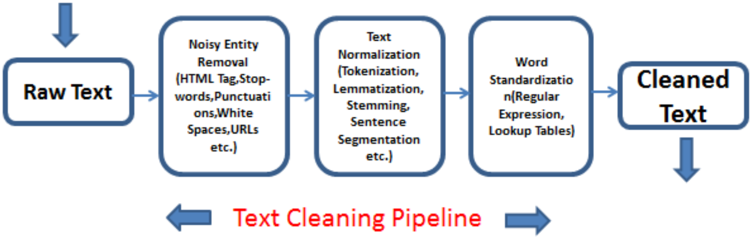
$\Rightarrow$ Requires advanced linguistic ressources including words and inflected forms.

E.g. : lions $\Rightarrow$ lion; are $\Rightarrow$ be; ...

Stemming:

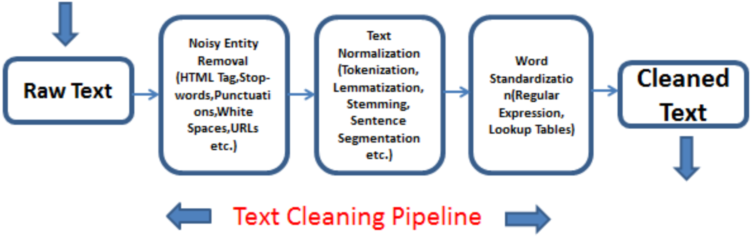
A statistical approximated process of the lemmatization

E.g. : removing **s** or **ly** at the end of the words



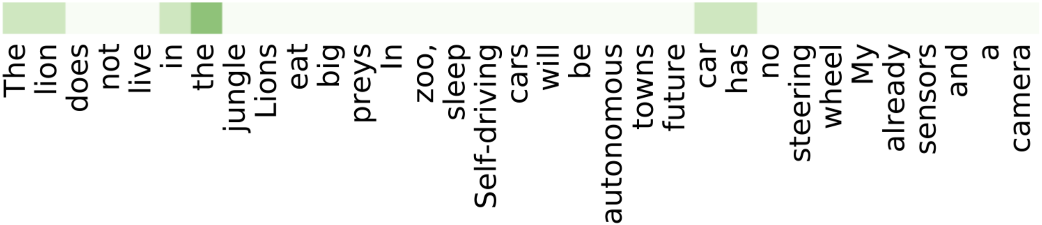
Pre-processing: word standardization

- Regular expression, *e.g.* for removing "." or expanding words' contractions ("I'll" $\rightarrow$ "I will")



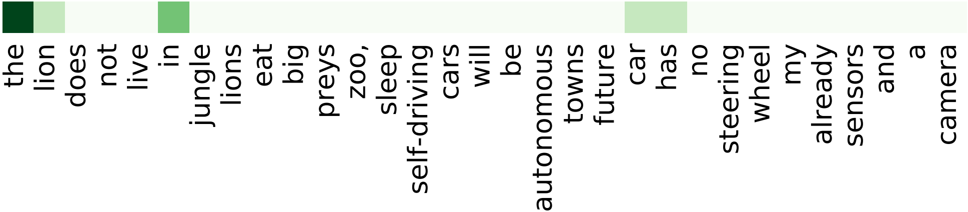
```
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

Original dictionary:



```
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

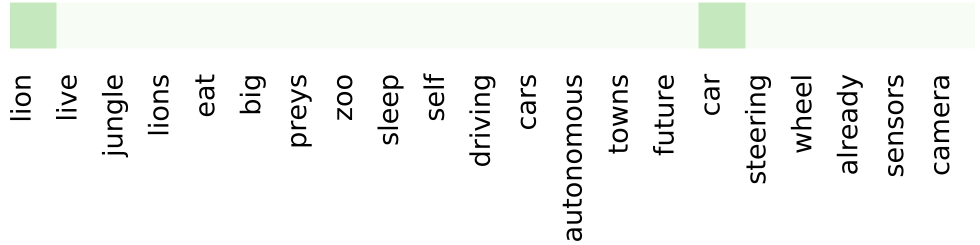
Removing capitals:



Dimensionality (and noise) reduction

```
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

Removing stop words:



Implementation: black list (nltk) or upper frequency bound

Dimensionality (and noise) reduction

```
    documents = ['The lion does not live in the jungle',\
    'Lions eat big preys',\
    'In the zoo, the lion sleep',\
    'Self-driving cars will be autonomous in towns',\
    'The future car has no steering wheel',\
    'My car already has sensors and a camera']
```

Removing rare words occurring less than a threshold :

Dictionary = {lion, car}

⇒ too extreme in this toy example...

But a good idea in real situations.

Remainder: rare words represent a large part of the dictionary

⇒ Tricky setting of the thresholds (upper & lower bounds)

Dimensionality (and noise) reduction

```
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

Lemmatization:

in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form

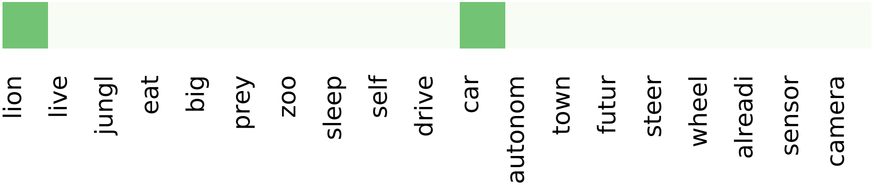
⇒ Requires advanced linguistic ressources including words and inflected forms.
E.g. : lions ⇒ lion; are ⇒ be; ...

```
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

Stemming:

A statistical approximated process of the lemmatization

E.g. : removing **s** or **ly** at the end of the words

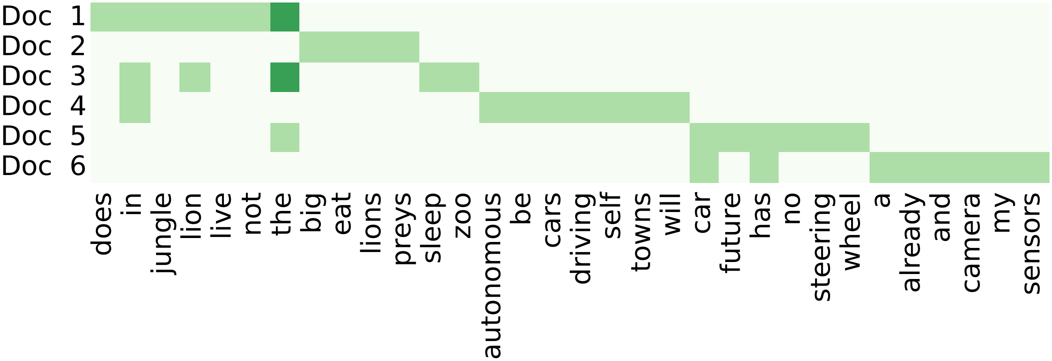


Note: it is not a problem to create invalid words... If they are stable!

Corpus representation

```
documents = ['The lion does not live in the jungle',\
    'Lions eat big preys',\
    'In the zoo, the lion sleep',\
    'Self-driving cars will be autonomous in towns',\
    'The future car has no steering wheel',\
    'My car already has sensors and a camera']
```

Corpus mapping on the basic dictionary:



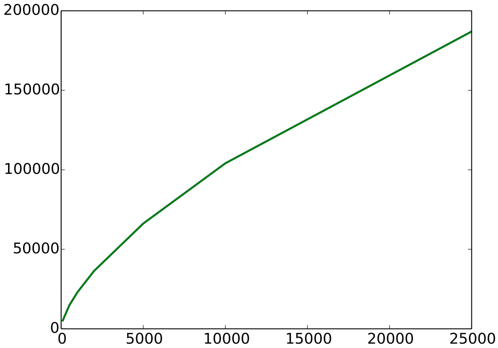
Vocabulary size:

$$|V| \propto \log(N), \qquad N = \text{number of documents}$$

On movie reviews :

$|V|$ with respect to $\#$ reviews
Let's have a closer look on the axes !!!

25k docs $\Leftrightarrow$ 200k words !!



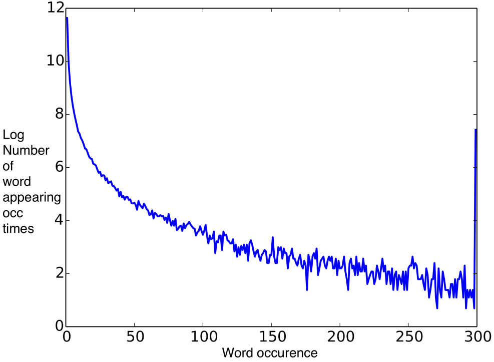
Vocabulary size:

$$|V| \propto \log(N), \qquad N = \text{number of documents}$$

Word occurence distribution:

$Occ_i = \{w |\ occurence(w) = i\}$

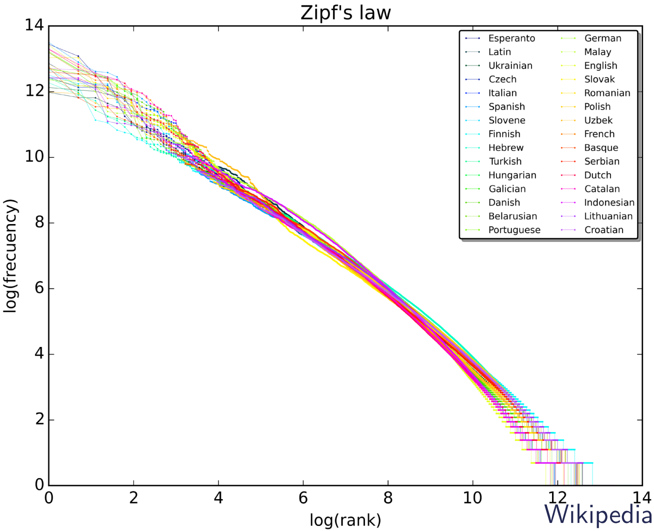
Plot $= \log(|Occ_i|)$ wrt i



Zipf law in several languages

$$f(n) = \frac{k}{n}$$

- n # documents
- k constant

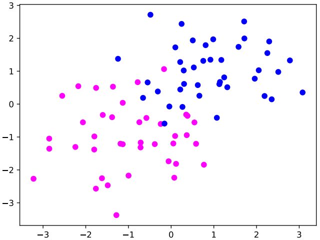


Wikipedia statistics

SCIENCES SORBONNE UNIVERSITÉ

A classical toy example to illustrate the curse of dimensionality:

Original dataset:

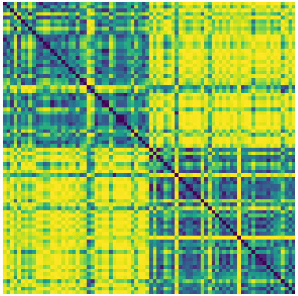


Matrix view

Matrix of raw points

Distance matrix

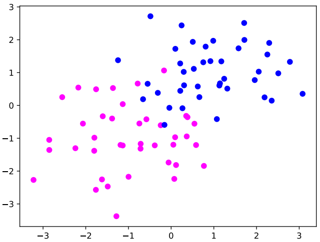
Matrix of distance between points



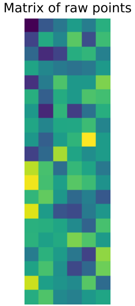
Easy problem / classes are clearly separated

A classical toy example to illustrate the curse of dimensionality:

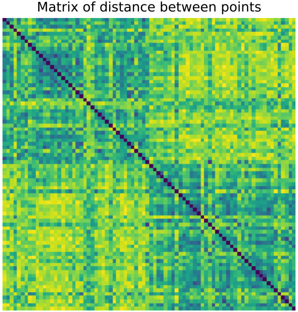
Original dataset:



Matrix view



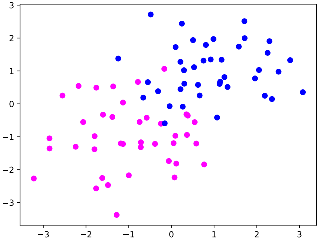
Distance matrix



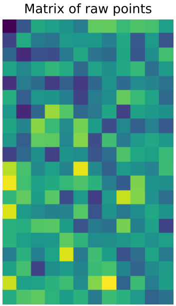
Adding some noisy dimensions in the dataset

A classical toy example to illustrate the curse of dimensionality:

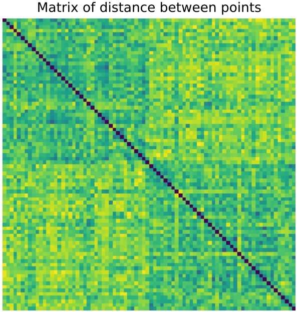
Original dataset:



Matrix view



Distance matrix

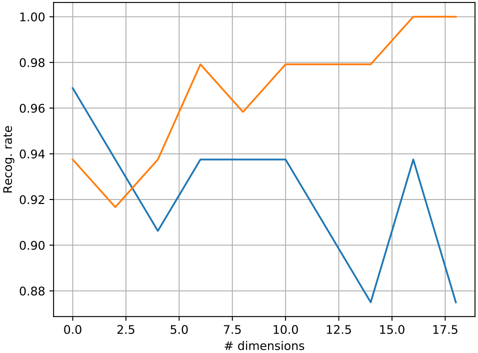


Adding **more** noisy dimensions in the dataset

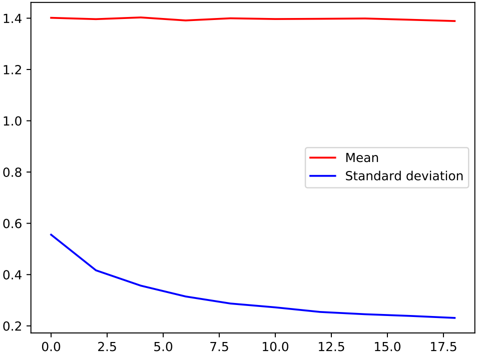
$\Rightarrow$ Euclidian distance is very sensitive to the dimensionality issue

A classical toy example to illustrate the curse of dimensionality:

Basic classifier on those datasets



Distances between points in the dataset



$\Rightarrow$ Learn accuracy $\nearrow$, test accuracy $\searrow$ = **overfitting**

$\Rightarrow$ All points tend to lay on an hypersphere (they become equidistant)

Given documents $\mathrm{d}_i \in \mathbb{R}^{|D|}$ and $d_j \in \mathbb{R}^{|D|}$ with the dictionary D.

First idea : **Euclidian metrics**

$$d(\mathrm{d}_i, \mathrm{d}_j) = \|\mathrm{d}_i - \mathrm{d}_j\| = \sqrt{\sum_k (d_{ik} - d_{jk})^2}$$

But:

$$d(\mathrm{d}_i, \mathrm{d}_j) = \sqrt{\|\mathrm{d}_i\|^2 + \|\mathrm{d}_j\|^2 - 2\mathrm{d}_i \cdot \mathrm{d}_j}$$

$\Rightarrow$ Sensitive to the norm of d or to the ratio $\mathrm{d}_i \cdot \mathrm{d}_j$ vs $\|\mathrm{d}\|$

- **Euclidian distance**
  $\Rightarrow$ not robust enough
- **Inner product**

$$sim(\mathrm{d}_i, \mathrm{d}_j) = \frac{\mathrm{d}_i \cdot \mathrm{d}_j}{\|\mathrm{d}_i\| \|\mathrm{d}_j\|} = \cos(\widehat{\vec{\mathrm{d}}_i, \vec{\mathrm{d}}_j}) \propto \sum_k d_{ik} d_{jk}$$

  $\Rightarrow$ focusing on common non-zeros dimensions
- **Kullback Leibler**
  Assuming that each document can be seen as a distribution over words (ie, $\forall i, \ \sum_k d_{ik} = 1$)

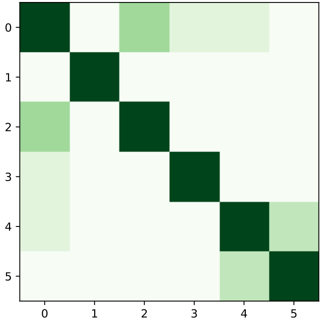
$$D_{\mathrm{KL}}(\mathrm{d}_i \| \mathrm{d}_j) = \sum_k d_{ik} \log \frac{d_{ik}}{d_{jk}}$$

  $\Rightarrow$ not very stable, take care of $d_{ik} = 0$ or $d_{jk} = 0$

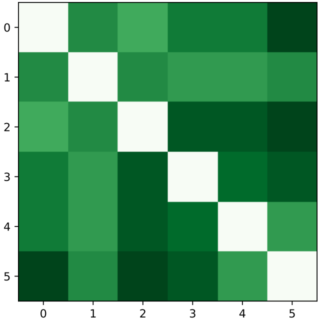
Metrics on our example

```python
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

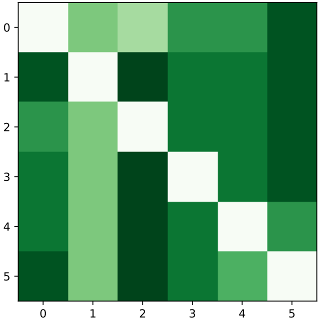
Cos



Euclidian



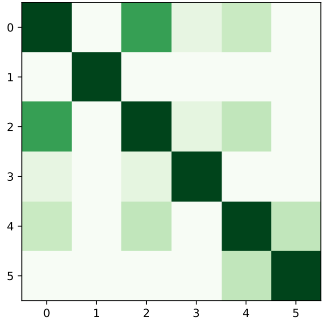
KL



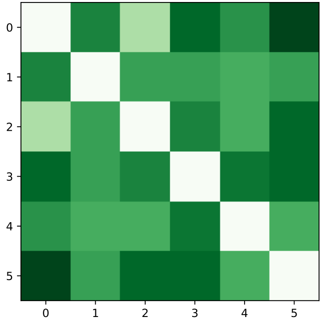
Basic representation of texts... **Too much noise !**

```
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

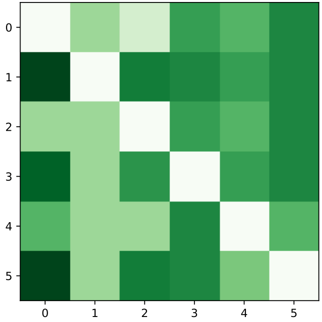
Cos



Euclidian



KL

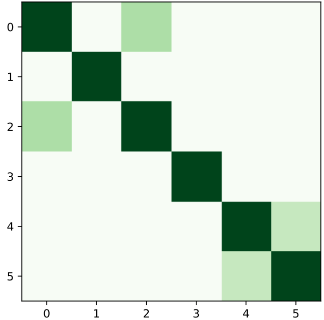


Preprocessing (removing capitals/puntuation) $\Rightarrow$ situation still confuse

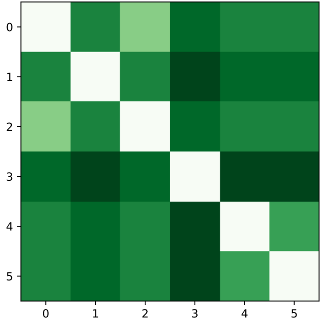
SCIENCES SORBONNE UNIVERSITÉ

```python
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

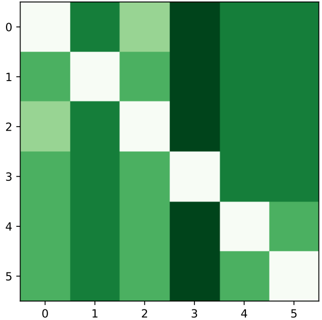
Cos



Euclidian



KL

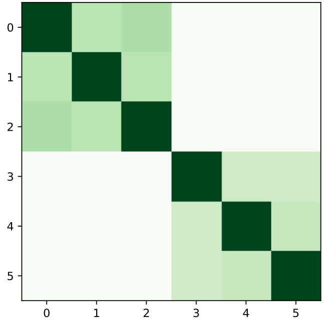


Preprocessing + removing stop words $\Rightarrow$ slight improvment

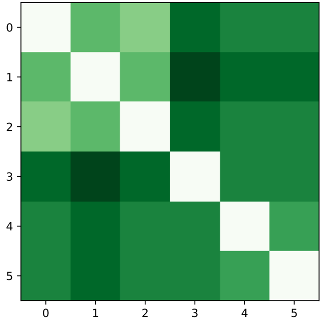
Metrics on our example


```python
documents = ['The lion does not live in the jungle',\
'Lions eat big preys',\
'In the zoo, the lion sleep',\
'Self-driving cars will be autonomous in towns',\
'The future car has no steering wheel',\
'My car already has sensors and a camera']
```

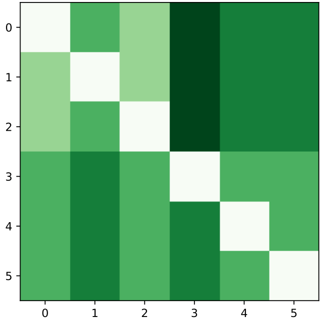
Cos



Euclidian



KL



Preprocessing + removing stop words + stemming $\Rightarrow$ distinction between classes

SCIENCES SORBONNE UNIVERSITÉ

IR main task :

Answering a query $q = \{q_1, \ldots, q_n\}$ by selecting documents d according to metrics : $dist(q, d)$

Most common metrics: BM25

$$\text{score}(q, d) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{freq(q_i, d) \cdot (k_1 + 1)}{freq(q_i, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)}$$

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}, \qquad b = 0.75, k_1 \in [1.2,\ 2.0]$$

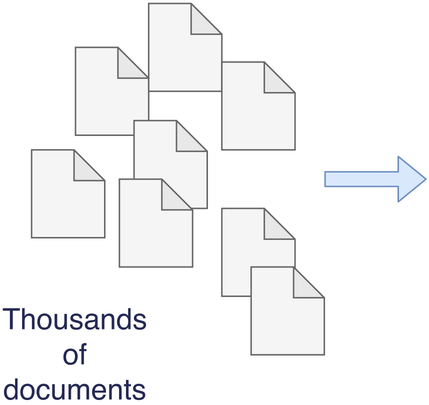
N : Corpus size, $n(q_i)$: Number of documents with q_i

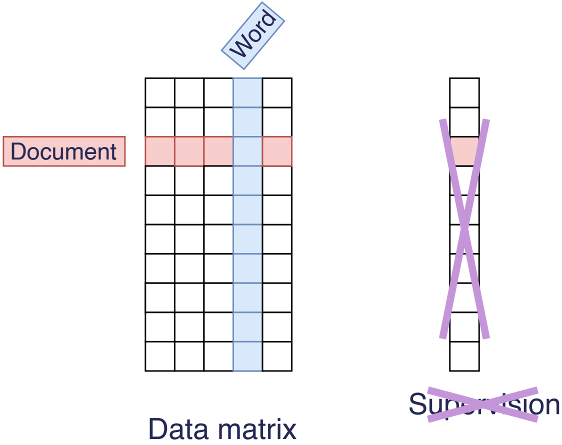
IR subtasks:

- Enforcing senrendipity
- Modeling source authority (PageRank)

- Eliminating word according to a criterion (still preprocessing)
  - Saliency : $S_{tf-idf}(i) = \frac{\sum_j \mathrm{tf-idf}(i,j)}{|\{\mathrm{tf-idf}(i,j) \neq 0\}|}$ (word i, word j)
  - Odds ratio: $S_{odds}(i) = \frac{p_i/(1-p_i)}{q_i/(1-q_i)} = \frac{p_i(1-q_i)}{q_i(1-p_i)}$. (often in log). Where p_i is the frequency of t_i in class 1 and q_i is the frequency of t_i in class 2.
  - Other criteria :Fisher, Mallows... Based on **separability**
- Regularization (improving robustness in learning)
  - cf after

Semantic modeling

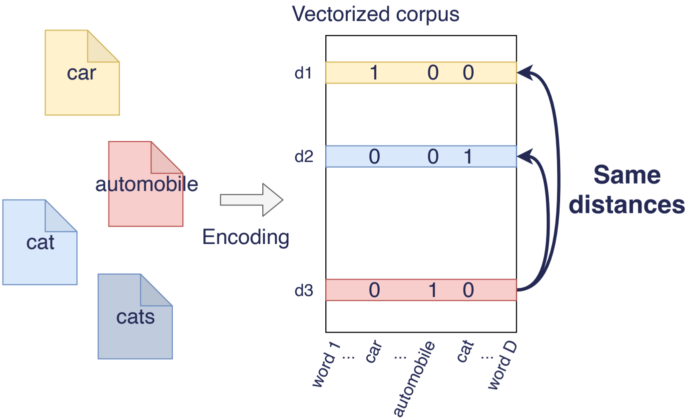




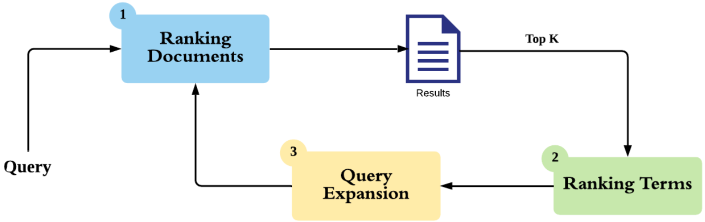
Clustering

Semantic analysis

- No context modeling
  - Negative form
  - Disambiguation
- Semantic gap



- N-gram encoding $\Rightarrow$ group of words
  - *very good*
  - *not good*
  - Combinatorial dictionary $\Rightarrow$ dimension issue !
- Lemmatization/stemming
  - 1 lexical stem = 1 column
  - Semantic / lexical ambiguities, *e.g.* polysemy (set , arm, head)
- Rocchio's strategy
  - Pseudo Relevance Feedback
  - Query expansion



SCIENCES SORBONNE UNIVERSITÉ

Objective

Understanding (automatically) word meaning

... And eliminating the semantic gap

$\Rightarrow$ Applications

- Information Retrieval
- Topic classification (& extraction)
- Information extraction
- Automated Summary
- Opinion classification
- ...

WordNet

- Description: Hierarchical description of words
  - Nouns
  - Verbs
  - Adjectives

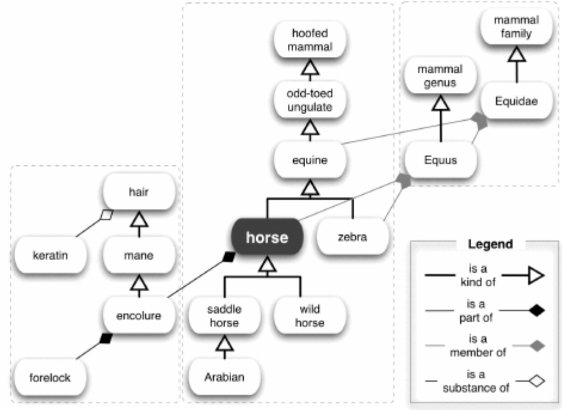
WordNet

- Description: Hierarchical description of words
  - Nouns
    - **hypernyms**: Y is a hypernym of X if every X is a (kind of) Y (canine is a hypernym of dog)
    - **hyponyms**: Y is a hyponym of X if every Y is a (kind of) X (dog is a hyponym of canine)
    - coordinate terms: Y is a coordinate term of X if X and Y share a hypernym (wolf is a coordinate term of dog, and dog is a coordinate term of wolf)
    - **meronym**: Y is a meronym of X if Y is a part of X (window is a meronym of building)
    - **holonym**: Y is a holonym of X if X is a part of Y (building is a holonym of window)
  - Verbs
  - Adjectives

SCIENCES SORBONNE UNIVERSITÉ

WordNet

- Description: Hierarchical description of words



  - Nouns
  - Verbs
  - Adjectives

WordNet

- Description: Hierarchical description of words
  - Nouns
  - Verbs
    - **hypernym**: the verb Y is a hypernym of the verb X if the activity X is a (kind of) Y (to perceive is an hypernym of to listen)
    - **troponym**: the verb Y is a troponym of the verb X if the activity Y is doing X in some manner (to lisp is a troponym of to talk)
    - **entailment**: the verb Y is entailed by X if by doing X you must be doing Y (to sleep is entailed by to snore)
    - **coordinate terms**: those verbs sharing a common hypernym (to lisp and to yell)
  - Adjectives

WordNet

- Description: Hierarchical description of words
  - Nouns
  - Verbs
  - Adjectives
    - **Antomyms / Synonyms**

- Metrics in WordNet
  - Length of the shortest path in the graph
  - Length of the shortest path in the *synonym* graph,
  - Distance of the first common ancestor,
  - cf: Leacock Chodorow (1998), Jiang Conrath (1997), Resnik (1995), Lin (1998), Wu Palmer (1993)
- WordNet & metrics are available in NLTK

SCIENCES SORBONNE UNIVERSITÉ

- Fully depend on static resources
  - New expressions + technical/specialized vocabulary may lack
  - Social network mining, Hashtags ...

Existing extensions:

- Several translations
- More generally : **a powerful diffusion tool**
  - Characterizing one part of the vocabulary
    + using WordNet to spread characterization (synonyms...)
- Applications
  - IR: Information Retrieval
  - Word Desambiguation
  - Text Classification
  - Machine Translation
  - Summarization

The General Inquirer

- Home page: `http://www.wjh.harvard.edu/~inquirer`
- List of Categories: `http://www.wjh.harvard.edu/~inquirer/homecat.htm`
- Spreadsheet: `http://www.wjh.harvard.edu/~inquirer/inquirerbasic.xls`
- Categories:
  - Positive (1915 words) and Negative (2291 words)
  - Strong vs Weak, Active vs Passive, Overstated versus Understated
  - Pleasure, Pain, Virtue, Vice, Motivation, Cognitive Orientation, etc
- Free for Research Use

Philip J. Stone, Dexter C Dunphy, Marshall S. Smith, Daniel M. Ogilvie. - MIT Press, 1966
The General Inquirer: A Computer Approach to Content Analysis

LIWC (Linguistic Inquiry and Word Count)

- Home page: `http://www.liwc.net/`
- 2300 words, >70 classes
- Affective Processes
  - negative emotion (bad, weird, hate, problem, tough)
  - positive emotion (love, nice, sweet)
- Cognitive Processes
  - Tentative (maybe, perhaps, guess), Inhibition (block, constraint)
  - Pronouns, Negation (no, never), Quantifiers (few, many)
- $30 or $90 fee

Pennebaker, J.W., Booth, R.J., & Francis, M.E. 2007. Austin, TX
Linguistic Inquiry and Word Count: LIWC

MPQA Subjectivity Cues Lexicon

- Home page: `http://www.cs.pitt.edu/mpqa/subj_lexicon.html`
- 6885 words from 8221 lemmas
  - 2718 positive
  - 4912 negative
- Each word annotated for intensity (strong, weak)
- GNU GPL

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann, EMNLP 2005
Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis

Bing Liu Opinion Lexicon

- Bing Liu's Page on Opinion Mining
  `http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`
- 6786 words
  - 2006 positive
  - 4783 negative

Minqing Hu and Bing Liu. ACM SIGKDD-2004.
Mining and Summarizing Customer Reviews

SentiWordNet

- Home page: `http://sentiwordnet.isti.cnr.it/`
- All WordNet synsets automatically annotated for degrees of:
  - positivity, negativity, and neutrality/objectiveness
- Many contexts investigated

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. LREC-2010
SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining

With an example: **short**

ADJECTIVE

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. LREC-2010
SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining

[C. Potts] Disagreements between polarity lexicons

	Opinion Lexicon	General Inquirer	SentiWordNet	LIWC
MPQA	33/5402 **(0.6%)**	49/2867 **(2%)**	1127/4214 **(27%)**	12/363 **(3%)**
Opinion Lexicon		32/2411 **(1%)**	1004/3994 **(25%)**	9/403 **(2%)**
General Inquirer			520/2306 **(23%)**	1/204 **(0.5%)**
SentiWordNet				174/694 **(25%)**
LIWC				

Target:

- Extracting the meaning of words and patterns of words
- ... Namely, understanding the message and deducing the polarity

$\Rightarrow$ Building Universal Models

Important tasks and subtasks:

- Building/learning/using lexical resources
- Extracting complex sentiment patterns
- Dealing with different problems related to sentiment defintion ($(e_j, a_{jk}, so_{ijkl}, h_i, t_l)$, entity, feature, polarity, holder, time)

Stanford NLP tools : `http://nlp.stanford.edu`
Named Entity Recognition, Dependency Tree Building, POS Tagging...

- **Input:** handmade opinion reference list, *i.e.* a list a word with their polarity (binary or continuous)
- **Output**: the polarity propagated to other words/token

$\Rightarrow$ **How to perform diffusion/propagation of polarity?**

1. Option 1: using WordNet to propagate to synonymes, and hypernyms / hyponyms
2. Option 2: diffusion with external sources / specific methods
   - **Solution:** compute co-occurence between tagged words and others

$\Rightarrow$ **How to measure co-occurences?**

SCIENCES SORBONNE UNIVERSITÉ

Mutual Information:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right),$$

kind of similarity between X et Y.

Pointwise Mutual Information:

$$PMI(X,Y) = \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$

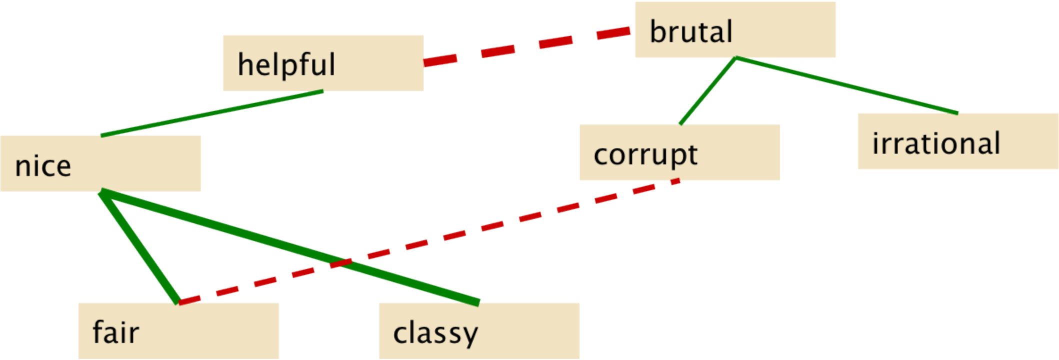
How much more do events x and y co-occur than if they were independent? (i.e. $PMI = 0$ in case of independence)

Goal : input nouns with associated ($\oplus$ /$\ominus$) polarity

- Adjectives separated by **and** $\Rightarrow$ same polarity
  - Fair **and** legitimate, corrupt **and** brutal
  - fair **and** brutal, corrupt **and** legitimate
- Adjectives separated by **but** $\Rightarrow$ different polarity
  - fair **but** brutal
- Initialization: 1336 adjectives ($\approx$ 50/50 positive/negative)

$$\boxed{\textbf{Polarity propagation:} \frac{PMI(adjectif, positive\ words)}{PMI(adjectif, negative\ words)}}$$

Hatzivassiloglou McKeown 1997
Predicting the Semantic Orientation of Adjectives



+ clustering

Hatzivassiloglou McKeown 1997
Predicting the Semantic Orientation of Adjectives

Hatzivassiloglou and McKeown 1997

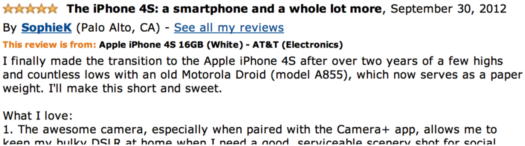
Results :

- **Positive**
  bold decisive disturbing generous good honest important large mature patient peaceful positive proud sound stimulating straightforward strange talented vigorous witty...
- **Negative**
  ambiguous cautious cynical evasive harmful hypocritical inefficient insecure irrational irresponsible minor outspoken pleasant reckless risky selfish tedious unsupported vulnerable wasteful...

Hatzivassiloglou McKeown 1997
Predicting the Semantic Orientation of Adjectives

Hu & Liu 2004

- Initialization from an annotated corpus (user reviews)



★★★★★ **The iPhone 4S: a smartphone and a whole lot more**, September 30, 2012

By **SophieK** (Palo Alto, CA) - See all my reviews

This review is from: Apple iPhone 4S 16GB (White) - AT&T (Electronics)

I finally made the transition to the Apple iPhone 4S after over two years of a few highs and countless lows with an old Motorola Droid (model A855), which now serves as a paper weight. I'll make this short and sweet.

What I love:
1. The awesome camera, especially when paired with the Camera+ app, allows me to keep my bulky DSLR at home when I need a good, serviceable scenery shot for social

    - Part of Speech analysis
    - Adjectives annotated from document label
- frequential filtering

summarization system:

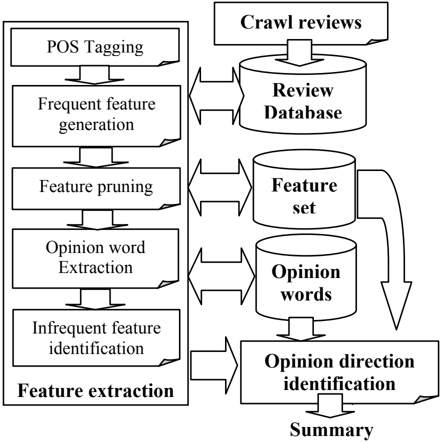


Figure 1: The opinion summarization system

Hu and Liu, AAAI NCAI 2004
Mining opinion features in customer reviews

Pointwise Mutual Information ,Turney, 2002

1. Documents ⇒ small patterns (=phrases)

First Word	Second Word	Third Word (not extracted)
JJ	NN or NNS	anything
RB, RBR, RBS	JJ	Not NN nor NNS
JJ	JJ	Not NN or NNS
NN or NNS	JJ	Nor NN nor NNS
RB, RBR, or RBS	VB, VBD, VBN, VBG	anything

2. Phrases evaluation
   - Positive phrases co-occur more with *excellent*
   - Negative phrases co-occur more with *poor*
3. Score aggregation at the document level

Turney, ACL 2002
Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews

PMI [Turney, 2002] : Results

Positive Reviews:

Phrase	POS tags	Polarity
online service	JJ NN	2.8
online experience	JJ NN	2.3
direct deposit	JJ NN	1.3
local branch	JJ NN	0.42
…		
low fees	JJ NNS	0.33
true service	JJ NN	−0.73
other bank	JJ NN	−0.85
inconveniently located	JJ NN	−1.5
Average		0.32

Negative Reviews:

Phrase	POS tags	Polarity
direct deposits	JJ NNS	5.8
online web	JJ NN	1.9
very handy	RB JJ	1.4
…		
virtual monopoly	JJ NN	−2.0
lesser evil	RBR JJ	−2.3
other problems	JJ NNS	−2.8
low funds	JJ NNS	−6.8
unethical practices	JJ NNS	−8.5
Average		−1.2

$\Rightarrow$ External resources: finding some pattens that are topic-related and not universal

- 410 reviews from Epinions
  - 170 (41%) negative
  - 240 (59%) positive
  - 106,580 phrases
- Majority class baseline: 59%
- Turney algorithm: 74%
- Only 66% on movie reviews
  (average is not a good solution...)

Key points:

- Phrases rather than words
- Learns domain-specific information
- Fast & require no labeled dataset

Domain of Review	Accuracy
Automobiles	84.00 %
Honda Accord	83.78 %
Volkswagen Jetta	84.21 %
Banks	80.00 %
Bank of America	78.33 %
Washington Mutual	81.67 %
Movies	65.83 %
The Matrix	66.67 %
Pearl Harbor	65.00 %
Travel Destinations	70.53 %
Cancun	64.41 %
Puerto Vallarta	80.56 %
All	74.39 %

Same methodology as Turney... But introducing other analysis axes :

$$\text{Evaluative factor: } EVA(m) = \frac{d(m, bad) - d(m, good)}{d(good, bad)} \quad (1)$$

$$\text{Potency factor: } POT(m) = \frac{d(m, weak) - d(m, strong)}{d(strong, weak)} \quad (2)$$

$$\text{Activity factor: } ACT(m) = \frac{d(m, passive) - d(m, active)}{d(active, passive)} \quad (3)$$

Quantitative results: 61% $\rightarrow$ 71%
Qualitative analysis: comparison with the General Inquirer

J. Kamps, MJ Marx, R.J Mokken et M. De Rijke, LREC 2004
Using wordnet to measure semantic orientations of adjectives

- Way to include semantics
- Can be combined with BoW models
  - Polarity score combined with BoW vector

Unsupervised approaches

Outline

- Modeling: Word count (and BoW storage)

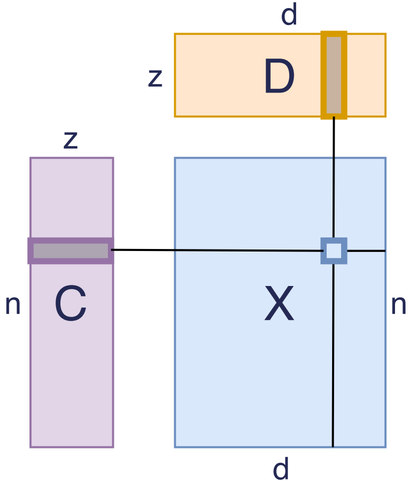
$$X = \begin{array}{cc} & \begin{array}{ccc} & \mathbf{t}_j & \\ & \downarrow & \end{array} \\ \mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,D} \end{pmatrix} \end{array}$$

- Basic proposal: semantics = metrics = similarity between columns in BoW

$$s(j,k) = \langle t_j, t_k \rangle, \qquad \text{Normalized: } s_n(j,k) = \cos(\theta) = \frac{\mathbf{t}_j \cdot \mathbf{t}_q}{\|\mathbf{t}_j\| \, \|\mathbf{t}_q\|}$$

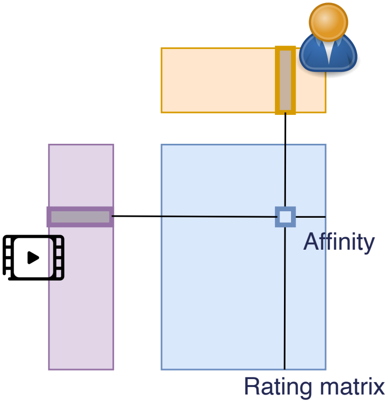
  - If two terms appear in the same document, they are similar

Matrix factorization = basic tool to understand the data



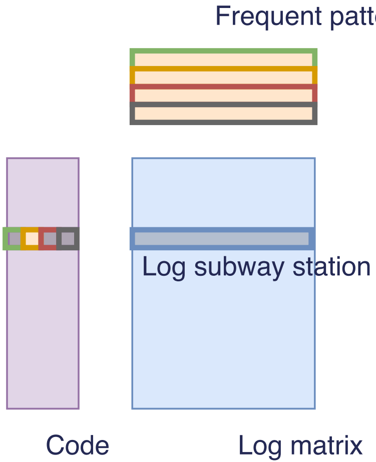
- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

Matrix factorization = basic tool to understand the data



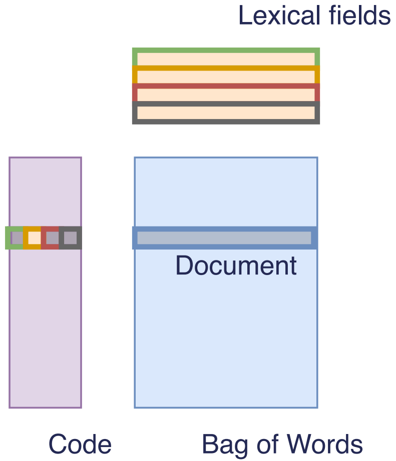
- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

Matrix factorization = basic tool to understand the data



- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

Matrix factorization = basic tool to understand the data



- Extract a compact representation
  - for words
  - for documents
- = focus on high-energy phenomenon
  - Eliminate noise in the data
- Optimal data compression [Mean Square criterion]

SVD : Singular Value decomposition

- In NLP : SVD = LSA: Latent Semantic Analysis
- Idea : grouping similar documents / learning a representation of documents

$$
\begin{array}{ccccccc}
 & X^T & = & U & & \Sigma & V^T \\
 & \begin{array}{c}\mathbf{d}_i \\ \downarrow\end{array} & & & & & \begin{array}{c}\hat{\mathbf{d}}_i \\ \downarrow\end{array} \\
\mathbf{t}_j \rightarrow & \begin{pmatrix} x_{1,1} & \dots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{D,1} & \dots & x_{D,N} \end{pmatrix} & = & \begin{pmatrix} \begin{pmatrix} \\ \mathbf{u}_1 \\ \\ \end{pmatrix} \dots \begin{pmatrix} \\ \mathbf{u}_l \\ \\ \end{pmatrix} \end{pmatrix} & & \begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{pmatrix} & \begin{pmatrix} (\quad \mathbf{v}_1 \quad) \\ \vdots \\ (\quad \mathbf{v}_l \quad) \end{pmatrix}
\end{array}
$$

- Good news: functions well on sparse matrices

Factorization = robustness & clustering ability

S. Deerwester, et al., JSIS 1990
Indexing by latent semantic analysis

Selecting the k greatest singular values gives a rank-k approximation of the occurence matrix.

- Each $\mathbf{u} \in \mathbb{R}^D$ is a weight vector associated to the vocabulary
- The base $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ is orthogonal
  - Each $\mathbf{u}$ corresponds to a different **lexical field**
- The new document representation v is a weight vector associated to the lexical fields
  - Clustering issue: the strongest weight gives the document class

Thomas K. Landauer, Peter W. Foltz et Darrell Laham, Discourse Processes, vol. 25, 1998
Introduction to Latent Semantic Analysis

Many applications

Usages:

- Clustering (each eigen vector describes a *topic)*
- Semantics: words have a representation over the topics
- IR Improvement:
  - Query expansion based on the topic definition
  - Detection of polysemic terms
- new representation $\Rightarrow$ new metrics
  - opportunities in question answering
    - Finding the part of a document relating to a specific topic
  - Automated summarization
    - Document segmentation + sentence extraction
  - TDT : Topic detection & Tracking

LSA Limits

- Fully based on BOW: no word dependency modeling
  - issues regarding negative formulation
  - depends on document sizes
  - Not robust to stop words
    - associated to high singular values
    - + appear in many topics
- Topic modeling is link to a corpus
  - problem with rare words in small corpus
  - bias of the corpus

LSA variation : k-means

- Still a BOW modeling

$$X = \begin{array}{cc} & \mathbf{t}_j \\ & \downarrow \\ \mathbf{d}_i \rightarrow & \begin{pmatrix} x_{1,1} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,D} \end{pmatrix} \end{array}$$

- Algorithm that scale up well
  - Possible **on-line** version of the algorithm
  - Can be linked to chinese restaurant / indian buffet process
    - $\Rightarrow$ Discover k in an online process
- Orthogonality is not longer enforced

SCIENCES SORBONNE UNIVERSITÉ

New vision of k-means :

- k clusters
- A priori probabilities : $\pi_k = p(\theta_k)$
- Probability of a word in a cluster : $p(w_j|\theta_k) = \mathbb{E}_{d\in\mathcal{D}_k}[w_j]$
- Document hard assignment in a cluster: $p(\theta_k|d_i) = 1/0$

$$y_i = \arg\max_k p(\theta_k)p(d_i|\theta_k) = \arg\max_k \log(\pi_k) + \sum_{w_j\in d_i} \log p(w_j|\theta_k)$$

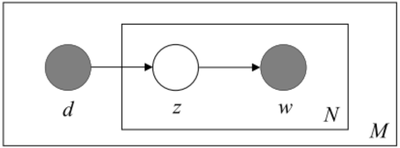
$$y_i = \arg\max_k \sum_j t_{ij}\theta_{jk}, \text{ with } :\theta_{jk} = \log p(w_j|\theta_k) \text{ and uniform prior}$$

Algorithm:

Init. Random or expert knowledge
C/E Cluster assignment
M Parameter update (mean re-computation)

Probabilistic Latent Semantic Analysis

- Idea: CEM $\Rightarrow$ EM (more complex / finer)
- All documents belongs to all clusters... With a weight $p(z|d)$
- Graphical model



- Doc d is drawn from $P(d)$
- Topic z is drawn from $P(z|d)$
- Word w is drawn from $P(w|z)$

We estimate the following parameters:

- $p(d)$
- $p(\alpha|d)$
- $p(w|\alpha)$

Maximizing the log-likelyhood:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{W} n(d, w) \log P(d, w)$$

- Expectation (probability of the missing variables)
- Maximization

Maximizing the log-likelyhood:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{W} n(d, w) \log P(d, w)$$

- **E**xpectation (probability of the missing variables)

$$P(\alpha|d, w) = \frac{P(d)P(\alpha|d)P(w|\alpha)}{\sum_{\alpha' \in \mathcal{A}} P(d)P(\alpha'|d)P(w|\alpha')}$$

- **M**aximization

Maximizing the log-likelyhood:

$$\mathcal{L} = \sum_{d=1}^{D} \sum_{w=1}^{W} n(d, w) \log P(d, w)$$

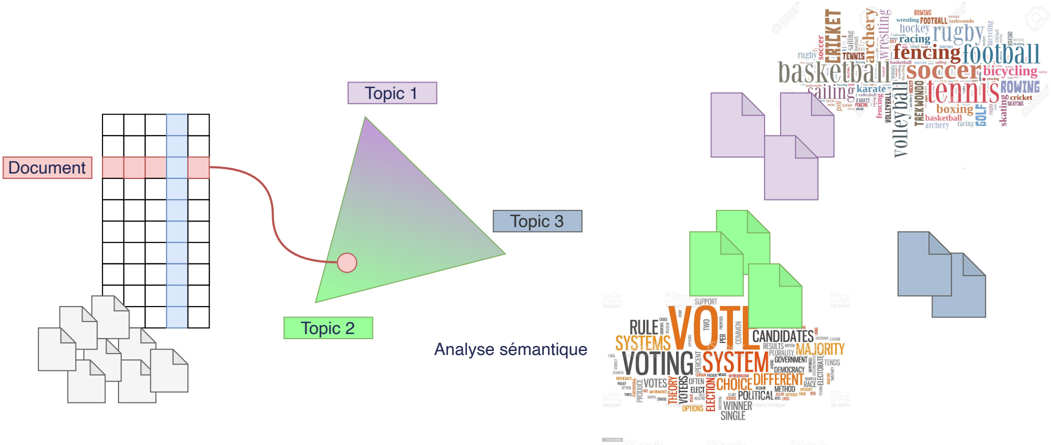
- Expectation (probability of the missing variables)
- Maximization

$$P(d) = \frac{\sum_{w \in \mathcal{W}} n(d, w)}{\sum_{d' \in \mathcal{D}} \sum_{w \in \mathcal{W}} n(d', w)}$$

$$P(\alpha|d) = \frac{\sum_{w \in \mathcal{W}} n(d, w) P(\alpha|d, w)}{\sum_{\alpha' \in \mathcal{A}} \sum_{w \in \mathcal{W}} n(d, w) P(\alpha'|d, w)}$$

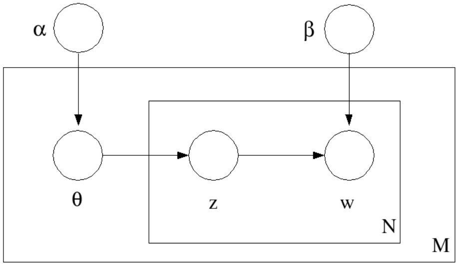
$$P(w|\alpha) = \frac{\sum_{d \in \mathcal{D}} n(d, w) P(\alpha|d, w)}{\sum_{w' \in \mathcal{W}} \sum_{d \in \mathcal{D}} n(d, w') P(\alpha|d, w')}$$

SCIENCES SORBONNE UNIVERSITÉ



Outline

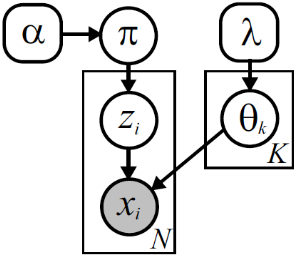
Latent Dirichlet Allocation:

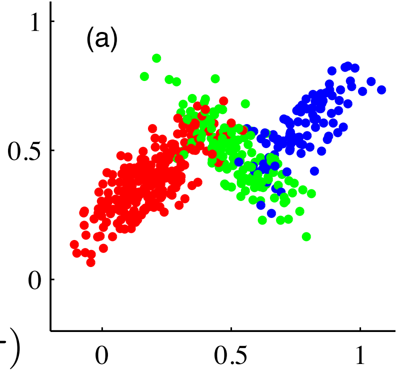


- Idea: adding a prior on the topic distribution
  - A document is supposed to belong to a topic **strongly or not**
- Learning through Gibbs sampling ($\sim$ MCMC)

not to be confused: LDA: Latent Dirichlet Allocation *vs* Linear Discriminant Analysis

On an example:





$$\theta_k = \{\mu_k, \Sigma_k\}$$
$$p(z_i \mid \pi) = \mathrm{Cat}(z_i \mid \pi)$$
$$p(x_i \mid z_i, \mu, \Sigma) = \mathcal{N}(x_i \mid \mu_{z_i}, \Sigma_{z_i})$$

SCIENCES SORBONNE UNIVERSITÉ

Given mixture weights $\pi^{(t-1)}$ and cluster parameters $\{\theta_k^{(t-1)}\}_{k=1}^K$ from the previous iteration, sample a new set of mixture parameters as follows:

1. Independently assign each of the N data points x_i to one of the K clusters by sampling the indicator variables $z = \{z_i\}_{i=1}^N$ from the following multinomial distributions:

$$z_i^{(t)} \sim \frac{1}{Z_i} \sum_{k=1}^{K} \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})\, \delta(z_i, k) \qquad Z_i = \sum_{k=1}^{K} \pi_k^{(t-1)} f(x_i \mid \theta_k^{(t-1)})$$

2. Sample new mixture weights according to the following Dirichlet distribution:

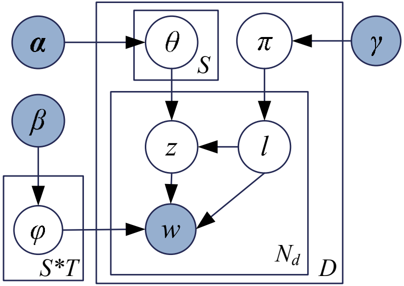
$$\pi^{(t)} \sim \mathrm{Dir}(N_1 + \alpha/K, \ldots, N_K + \alpha/K) \qquad N_k = \sum_{i=1}^{N} \delta(z_i^{(t)}, k)$$

3. For each of the K clusters, independently sample new parameters from the conditional distribution implied by those observations currently assigned to that cluster:

$$\theta_k^{(t)} \sim p(\theta_k \mid \{x_i \mid z_i^{(t)} = k\}, \lambda)$$

When λ defines a conjugate prior, this posterior distribution is given by Prop. 2.1.4.

- Graphical models = easy to adapt



- For each document d, choose a distribution $\pi_d \sim \mathrm{Dir}(\gamma)$.
- For each sentiment label l under document d, choose a distribution $\theta_{d,l} \sim \mathrm{Dir}(\alpha)$.
- For each word w_i in document d
  - choose a sentiment label $l_i \sim \mathrm{Mult}(\pi_d)$,
  - choose a topic $z_i \sim \mathrm{Mult}(\theta_{d,l_i})$,
  - choose a word w_i from $\varphi_{z_i}^{l_i}$, a Multinomial distribution over words conditioned on topic z_i and sentiment label l_i.

1. Quantitative results
   - Clustering
   - Major issue with frequent words
   - Human required in the loop (init., cluster selection, etc...)
   - Evaluation issue (purity, perplexity, ...)
2. Qualitative analysis
   - Word similarity
   - Lexical field extraction