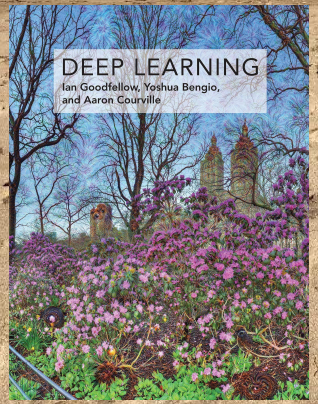
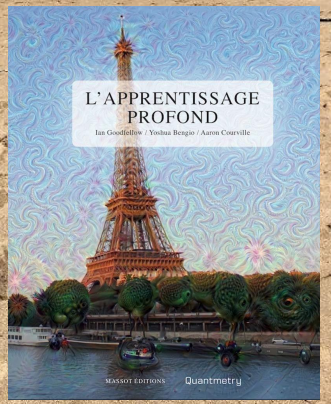
Information extraction (IE)
Traitement du langage naturel (TALN)

Emanuela Boros
University of La Rochelle, France

emanuela.boros@univ-lr.fr

December 2021





Organization

1. 1.5h Courses: Word Embeddings & Text Classification
2. 1.5h Lab work (dataset provided)

- Course (slides) are available on Moodle
- Lab work: Google Classroom + Google Colab + Jupyter Notebook (python)

TALN

Traitement du langage naturel = le domaine d'étude qui se concentre sur les interactions entre le langage humain et les ordinateurs. Il se situe à l'intersection de l'informatique, de l'intelligence artificielle et de la linguistique informatique (Wikipédia).

Information Extraction/Retrieval - Text Mining = le processus d'organisation et d'extraction automatique d'informations pertinentes à partir de texte non structuré (documents, commentaires des clients, médias sociaux, e-mail, etc.).

TP : Ayant les textes classés (classification des textes) et regroupés par type, le système essaie d'en extraire des informations utiles.

TALN

- **Text Classification:** la tâche de choisir l'étiquette de classe correcte pour une entrée donnée
  - Décider si un est un spam ou non (détection de spam)
  - Décider si le sujet d'un article d'actualité fait partie d'une liste fixe de domaines tels que « sport », « technologie » et « politique » (classification des documents)
- **Sentiment Analysis (opinion mining):** identifier et extraire des informations subjectives dans le matériel source
  - Reviews, notes et recommandations sur les sites de médias sociaux pour les entreprises qui cherchent à commercialiser leurs produits, à identifier de nouvelles opportunités et à gérer leur réputation

Information Retrieval

- **Topic Modeling:** algorithmes pour découvrir les thèmes principaux qui imprègnent une collection importante et autrement non structurée de documents
- **Named Entity Recognition:** méthodes d'identification des entités dans les données (noms, lieux, organisations, etc.)

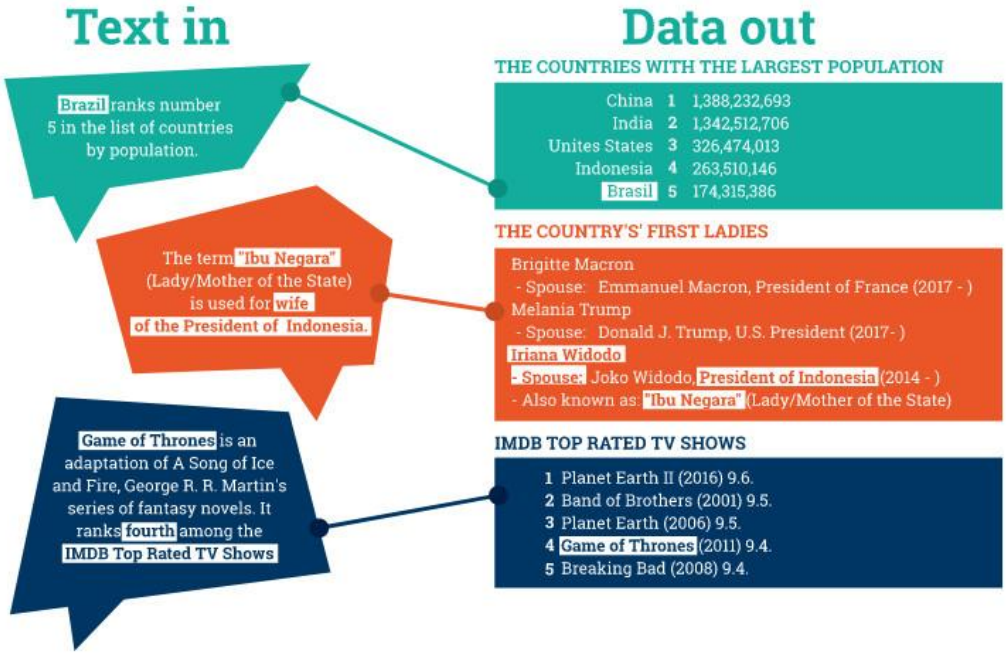
Information Extraction

- **Relation Extraction:** discerne les relations qui existent entre les entités détectées dans un texte
- **Event Extraction:** discerne les relations qui existent entre les entités détectées dans un texte

Information Extraction

Les systèmes **Information Extraction** extraient des informations claires et factuelles→ *Who did what to whom when?*

- identifier les informations pertinentes à partir de documents (actualités financières, ou informations touristiques, documents médicaux), extraire des informations de diverses sources et les agréger sous une forme structurée



Information Extraction - examples

Named Entity Recognition (NER)

Une tâche très importante : rechercher et classer des entités dans le texte, par exemple :

"We had a very nice stay, the hotel is nicely situated in walking distance to the Eiffel tower and the exhibition halls of Porte de Versailles. Clean, nice hotel with a great neighborhood. We were about an hour walk to most of the big attractions (Eiffel Tower, Notre Dame, The Louvre). Some days we walked and some days we ubered or took the metro. Both Uber and Metro are easily navigated here. There is a metro very close by that runs conveniently through the city. The breakfast buffet has a large choice of fruits, bread etc. Can only recommend!"

Information Extraction - examples

Named Entity Recognition (NER)

Une tâche très importante : rechercher et classer des entités dans le texte, par exemple :

"We had a very nice stay, the hotel is nicely situated in walking distance to the exhibition halls of **Porte de Versailles**. Clean, nice hotel with a great neighborhood. We were about an hour walk to most of the big attractions (**Eiffel Tower**, **Notre Dame**, **Louvre**). Some days we walked and some days we ubered or took the metro. Both **Uber** and **Metro** are easily navigated here. There is a metro very close by that runs conveniently through the city. The breakfast buffet has a large choice of fruits, bread etc. Can only recommend!"

Information Extraction - examples

Named Entity Recognition (NER)

Une tâche très importante : rechercher et classer des entités dans le texte, par exemple :

"We had a very nice stay, the hotel is nicely situated in walking distance to the exhibition halls of **Porte de Versailles**. Clean, nice hotel with a great neighborhood. We were about an hour walk to most of the big attractions (**Eiffel Tower**, **Notre Dame**, **Louvre**). Some days we walked and some days we ubered or took the metro. Both **Uber** and **Metro** are easily navigated here. There is a metro very close by that runs conveniently through the city. The breakfast buffet has a large choice of fruits, bread etc. Can only recommend!"

Location
Organization

Information Extraction - examples

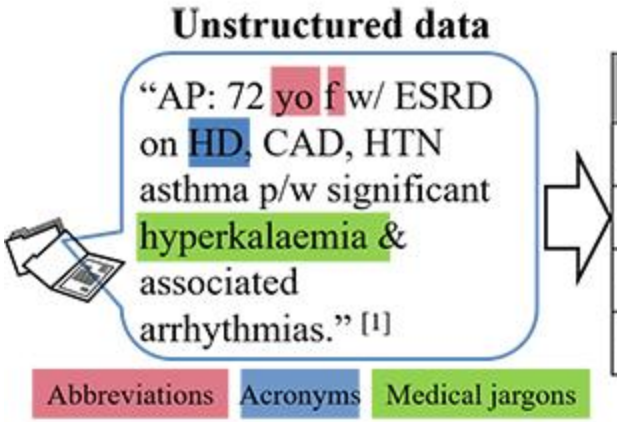
Named Entity Recognition (NER)

Type	Sample Categories	Example
People	Individuals, fictional Characters	***Turing*** is often considered to be the father of modern computer science.
Organization	Companies, parties	***Amazon*** plans to use drone copters for deliveries.
Location	Mountains, lakes, seas	The highest point in the ***Catalinas*** is ***Mount Lemmon*** at an elevation of 9,157 feet above sea level.
Geo-Political	Countries, states, provinces	The Catalinas, are located north, and northeast of ***Tucson***, ***Arizona***, ***United States.***
Facility	Bridges, airports	In the late 1940s, ***Chicago Midway*** was the busiest airport in the United States by total aircraft operations.
Vehicles	Planes, trains, cars	The updated ***Mini Cooper*** retains its charm and agility.

En pratique, la reconnaissance d'entités nommées peut être étendue à des types qui ne figurent pas dans le tableau ci-dessus, tels que des expressions temporelles (heure et dates), des gènes, des protéines, des concepts liés à la médecine (maladie, traitement et événements médicaux), etc.

Information Extraction - examples

Named Entity Recognition (NER)



Meaningful information

Attributes	Medical text
Age	72 years old (yo)
Gender	female (f)
Condition	hypertensive disease (HD)
Symptom	hyperkalemia

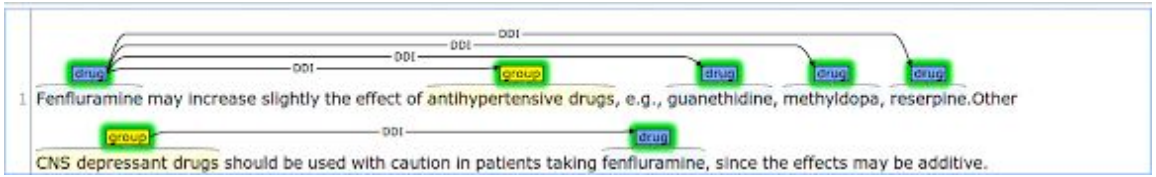
- concepts liés à la médecine (maladie, traitement et événements médicaux) et etc.
- les forums de patients en ligne peuvent fournir des informations supplémentaires précieuses sur l'efficacité des médicaments et leurs effets secondaires

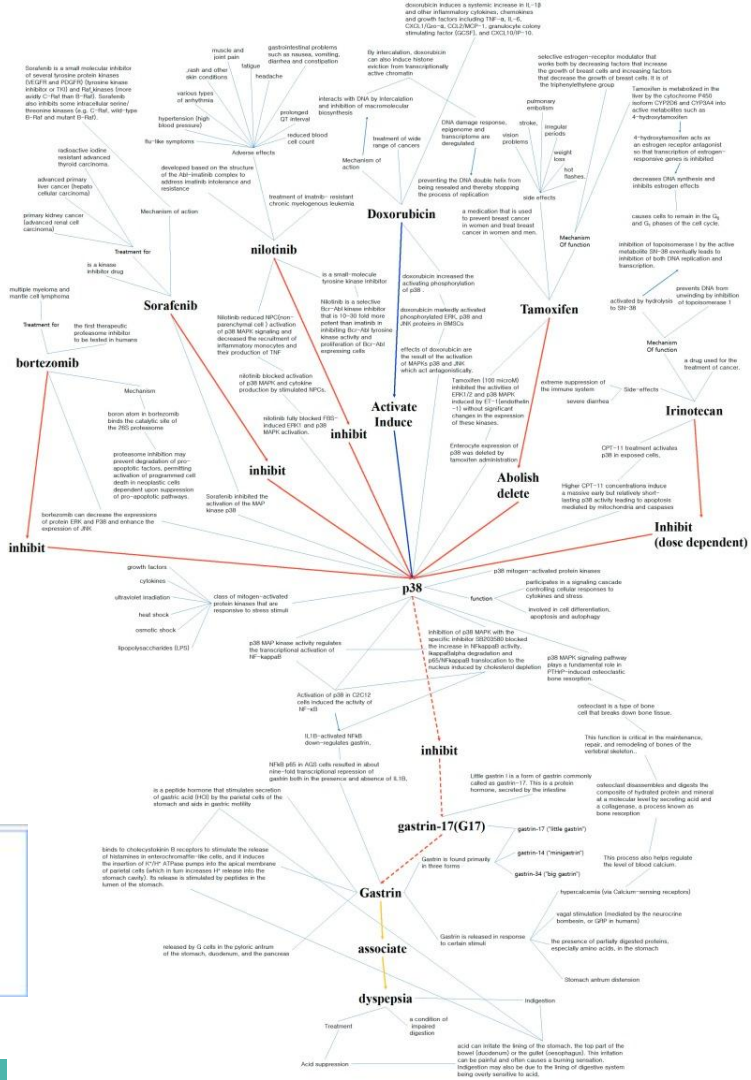
Information Extraction - examples

Relation Extraction (RE)

L'extraction de relations discerne les relations qui existent entre les entités détectées dans un texte.

- Extraction de textes médicaux : résumés de sortie, dossiers narratifs des patients
- Relations de liaison aux protéines utiles pour la découverte de médicaments
- Détection des relations gène-maladie à partir de la littérature biomédicale
- Trouver des relations entre les effets secondaires des médicaments dans les dossiers de santé



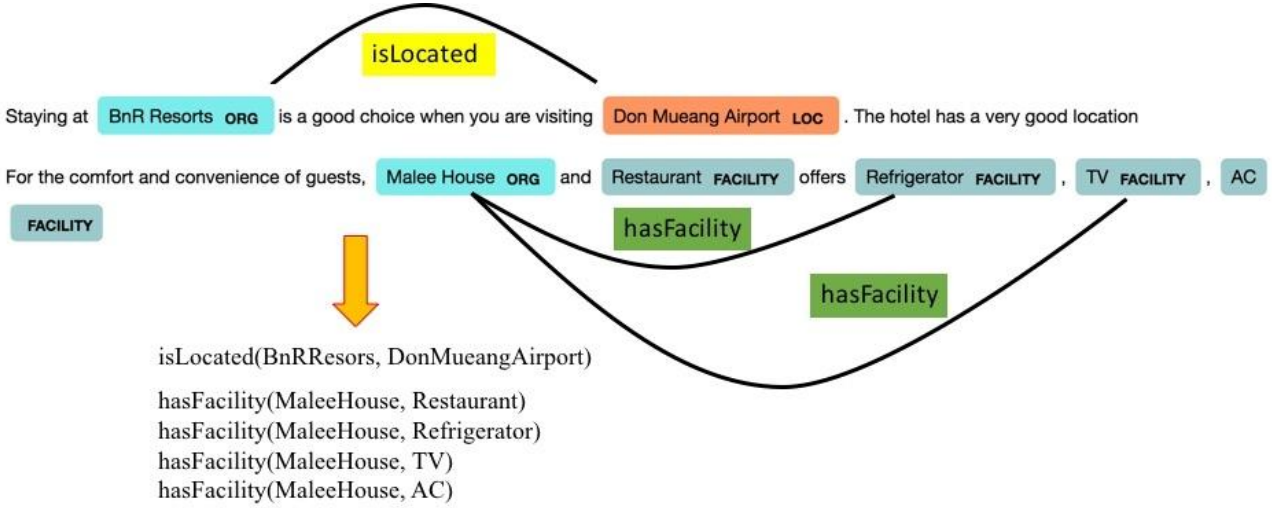


Information Extraction - examples

Relation Extraction (RE)

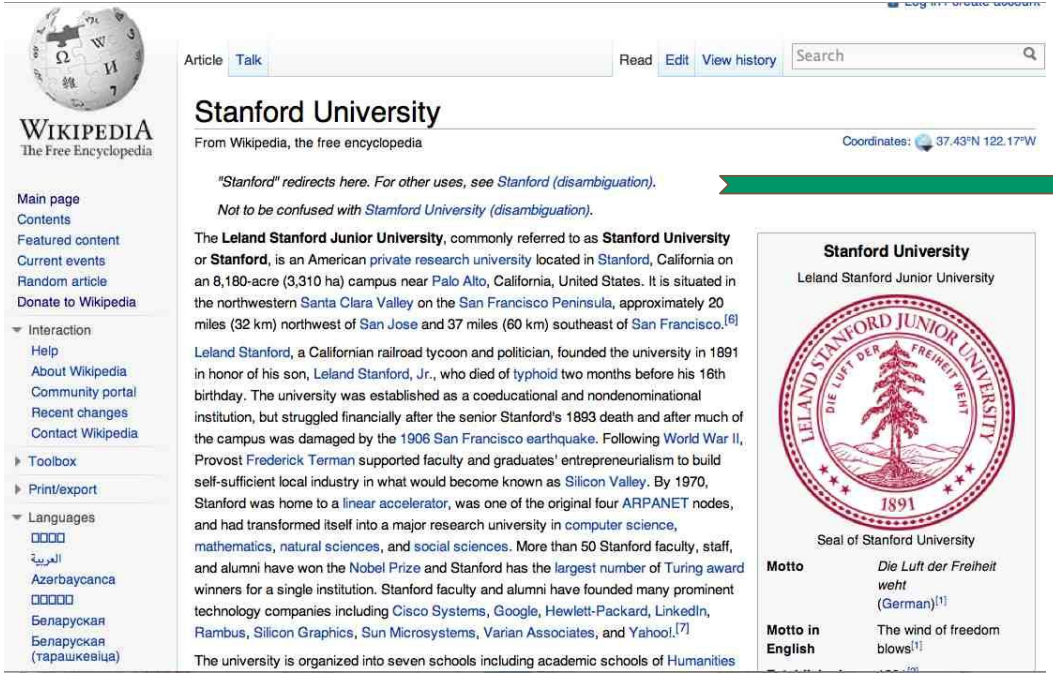
L'extraction de relations discerne les relations qui existent entre les entités détectées dans un texte.

- **Textes sur le tourisme minier : documents, retours clients, réseaux sociaux, email**



Information Extraction - examples

Relation Extraction (RE)



Article Talk | Read Edit View history | Search

Stanford University

From Wikipedia, the free encyclopedia

Coordinates: 37.43°N 122.17°W

"Stanford" redirects here. For other uses, see Stanford (disambiguation).

Not to be confused with Stamford University (disambiguation).

The **Leland Stanford Junior University**, commonly referred to as **Stanford University** or **Stanford**, is an American private research university located in Stanford, California on an 8,180-acre (3,310 ha) campus near Palo Alto, California, United States. It is situated in the northwestern Santa Clara Valley on the San Francisco Peninsula, approximately 20 miles (32 km) northwest of San Jose and 37 miles (60 km) southeast of San Francisco.[6]

Leland Stanford, a Californian railroad tycoon and politician, founded the university in 1891 in honor of his son, Leland Stanford, Jr., who died of typhoid two months before his 16th birthday. The university was established as a coeducational and nondenominational institution, but struggled financially after the senior Stanford's 1893 death and after much of the campus was damaged by the 1906 San Francisco earthquake. Following World War II, Provost Frederick Terman supported faculty and graduates' entrepreneurialism to build self-sufficient local industry in what would become known as Silicon Valley. By 1970, Stanford was home to a linear accelerator, was one of the original four ARPANET nodes, and had transformed itself into a major research university in computer science, mathematics, natural sciences, and social sciences. More than 50 Stanford faculty, staff, and alumni have won the Nobel Prize and Stanford has the largest number of Turing award winners for a single institution. Stanford faculty and alumni have founded many prominent technology companies including Cisco Systems, Google, Hewlett-Packard, LinkedIn, Rambus, Silicon Graphics, Sun Microsystems, Varian Associates, and Yahoo![7]

The university is organized into seven schools including academic schools of Humanities

Stanford University

Leland Stanford Junior University

Seal of Stanford University

Motto	*Die Luft der Freiheit weht* (German)[1]
Motto in English	The wind of freedom blows[1]

Stanford **LOC-IN** California
Stanford **IS-A** research university
Stanford **LOC-NEAR** Palo Alto
Stanford **FOUNDED-IN** 1891
Stanford **FOUNDER** Leland Stanford

Information Extraction - examples

Event Extraction = Named Entity Recognition + Relation Extraction

[S1] ... by special urban troops, four terrorists have been arrested in soacha.

[S2] They are responsible for the car bomb attack on the Newspaper El Espectador, to a series of bogota dynamite attacks, to the freeing of a group of paid assassins.

[S3] The terrorists are also connected to the murder of Teofilo Forero Castro, ...

[S4] General Ramon is the commander of the 13th infantry brigade.

[S5] He said that at least two of those arrested have fully confessed to having taken part in the accident of Luis Carlos Galan Sarmiento in soacha, Cundinamarca.

[S6] .. triumph over organized crime, its accomplices and its protectors.

...

Perpetrator Individual	four terrorists
Perpetrator Organization	-
Target	Newspaper El Espectador
Victim	Teofilo Forero Castro, Luis Carlos Galan Sarmiento
Weapon	car bomb, dynamite

Attack event:
- entities
- relations

Information Extraction - examples

Named Entity Recognition (NER)

- Game of Thrones: **TV Show**
- A Song of Ice and Fire: **Book**
- George R. R. Martin: **Author**

Relation Extraction (RE)

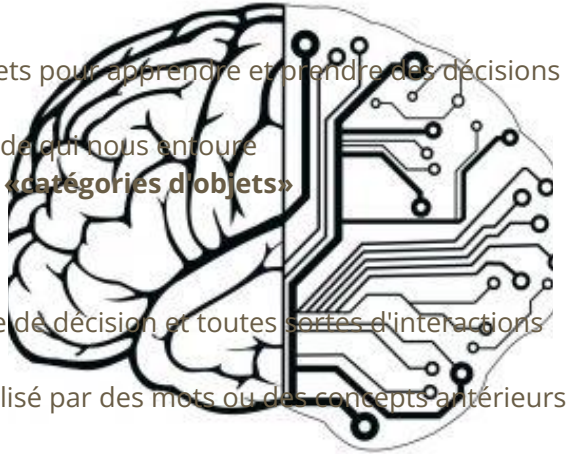
$$(\text{A Song of Ice and Fire} \xrightarrow{\text{Book-Author}} \text{George R. R. Martin})$$

Event Extraction (EE)

$$(\text{Game of Thrones} \xrightarrow{\text{TV Show-Author}} \text{George R. R. Martin} \xrightarrow{\text{Author-Book}} \text{A Song of Ice and Fire})$$

Text Classification

- Les **cerveaux humains** sont câblés pour reconnaître les patterns et classer les objets pour apprendre et prendre des décisions
- .. ils ne peuvent pas traiter chaque objet comme **unique**
- .. nous n'avons pas beaucoup de **ressources mémoire** pour pouvoir traiter le monde qui nous entoure
- → nos cerveaux développent des **«concepts»** ou des représentations mentales de **«catégories d'objets»**



- La classification est fondamentale dans le langage, la prédiction, l'inférence, la prise de décision et toutes sortes d'interactions environnementales
- Langue: par exemple, comment le sens des mots d'une phrase peut être contextualisé par des mots ou des concepts antérieurs

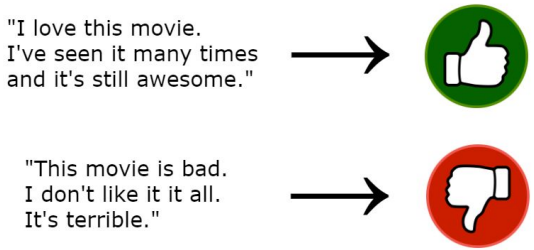
QUICK COMMENT:

AI can be sexist and/or racist
Racist data? It's the human bias that is Infecting the AI development

Text Classification: Introduction

- La **classification des objets** consiste à donner une classe à un objet.
- Ces objets peuvent être du type:
- texte, image, audio, vidéo, etc.

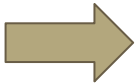
- Nous faisons de la classification tout le temps:
  - Nous pouvons reconnaître le chemin du retour de l'université
  - On peut reconnaître un chat qui est noir même si on n'a vu que des chats blancs et oranges auparavant
  - On peut reconnaître quand quelqu'un est ironique ou pas
  - Nous pouvons reconnaître la voix de nos amis
  - On peut même faire la distinction entre un Chihuahua et un muffin



Introduction

- Pour savoir comment classifier un objet, il est important de connaître les **caractéristiques** qui définissent une classe.

- Si on considère les **caractéristiques** :
  - nombre de roues
  - selle
  - guidon

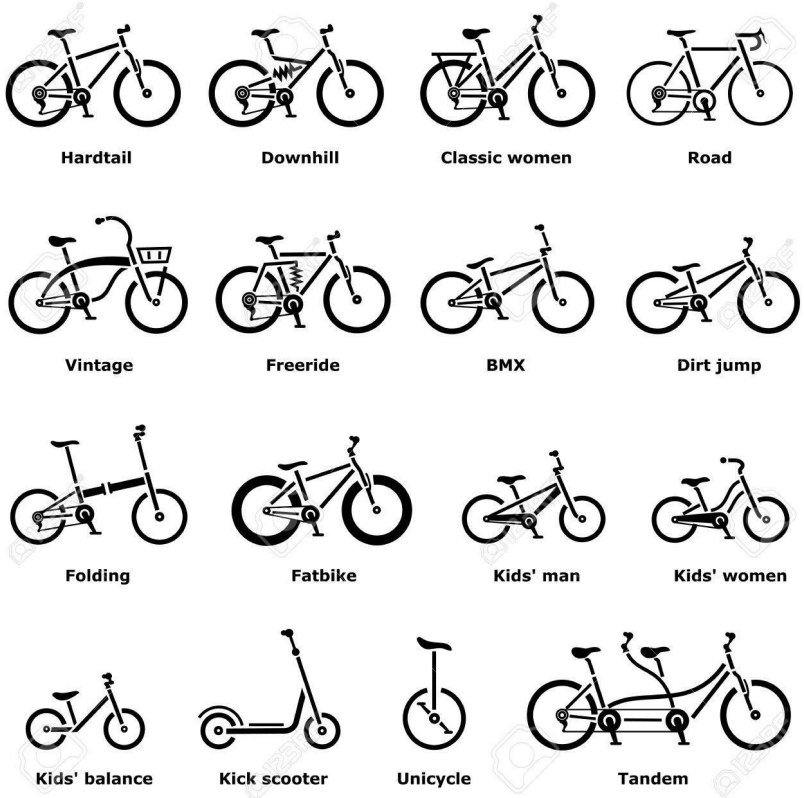


- Un *vélo* est composé de 2 roues, des freins, une selle et un guidon.
- Une *voiture* à 4 roues, des freins mais pas de selle ni guidon.
- Une *moto* est aussi composée de 2 roues, une selle, des freins et un guidon.

Avec ces caractéristiques, **la moto et le vélo ont la même représentation**.

Introduction

- Nous pourrions compliquer encore plus cette tâche et essayer de classifier les vélos suivants :



Introduction

- Un texte (message, sms, livre, les paroles d'une musique, etc.) peut être classifié dans plusieurs types de classes.

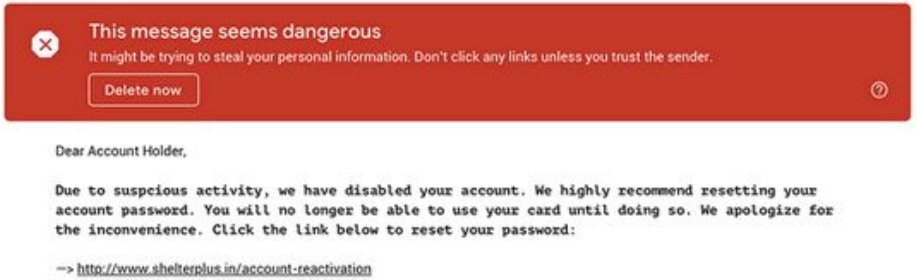
- Le contenu d'un livre peut être considéré comme :
  - Romantique
  - Comédie
  - Suspense
  - Fantastique
  - Science-fiction

- Un commentaire peut être:
  - Positif
  - Négatif
  - Neutre

- Un mail/message peut être:
  - Spam
  - Pas spam

Classification de textes : filtrage du spam

- Première **application industrielle à grande échelle** du machine learning
- Problème de classification
- Succès du modèle Naïve Bayes



Classification de textes : des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Classification de textes : des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

Happy

loved great delicious heavenly

Not Happy

mediocre dirty miserable

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Classification de textes : des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

Happy

loved great delicious heavenly

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

Not Happy

mediocre dirty miserable

Check-in was **smooth** and **fast**, and the staff were **nice**. But there were some serious **flaws**. The room was **dirty** and had **great** noise. The smell of **smoke** was extremely strong.

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Classification de textes : des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

Happy

loved great delicious heavenly

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

Not Happy

mediocre dirty miserable

Check-in was **smooth** and **fast**, and the staff were **nice**. But there were some serious **flaws**. The room was **dirty** and had **great** noise. The smell of **smoke** was extremely strong.

Not Happy

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Représentation vectorielle de documents

La représentation textuelle est importante car elle permet non seulement d'analyser ce type d'informations, mais aussi de transformer les textes en **données numériques** pour les algorithmes d'apprentissage automatique!

- D'une part, certains types de représentations sont très simples et rapides à calculer, d'autre part ils ne contiennent pas beaucoup d'informations sur ces mots.
- En revanche, les représentations les plus riches sont plus lentes à calculer mais elles contiennent plusieurs caractéristiques sur les mots.

Représentations de mots les plus importantes:

1. **One-hot encoding**
2. **Bag of words**
3. **TF-IDF (term frequency–inverse document frequency)**
4. **Word embeddings**

Représentation vectorielle de documents

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

15

Représentation vectorielle de documents

Je vous envoie ma nouvelle adresse. Je vous remercie.

Modèle binomial : présence / absence de mot

Modèle multinomial : comptage de mots

tf_i (term frequency)

One hot encoding

- La représentation **one hot encoding** est une de représentations les plus simples car tous les mots sont indépendants entre eux.
- La taille de la représentation augmente avec le corpus.
- Chaque vecteur est équidistant de tous les autres vecteurs

"A friend in need is a friend indeed."

`V = [a, friend, in, need, is, a, indeed], |V| = 7`

a	friend	in	need	is	a	indeed
1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

- Imaginez que nous ayons un vocabulaire de 50,000.
  (Il y a environ un million de mots en anglais.)

- Chaque mot est représenté par 49,999 zéros et un seul 1
  → nous avons besoin de $50{,}000^2$ = 2,5 milliards d'unités d'espace mémoire.
- **Pas efficace** en termes de calcul.

Bag-of-words (représentation de documents)

- La représentation **bag-of-words** est une représentation de documents en tenant compte de la fréquence des mots dans le document.
- La taille de la représentation augmente avec le corpus.

DOC1: "He is not a friend in need."

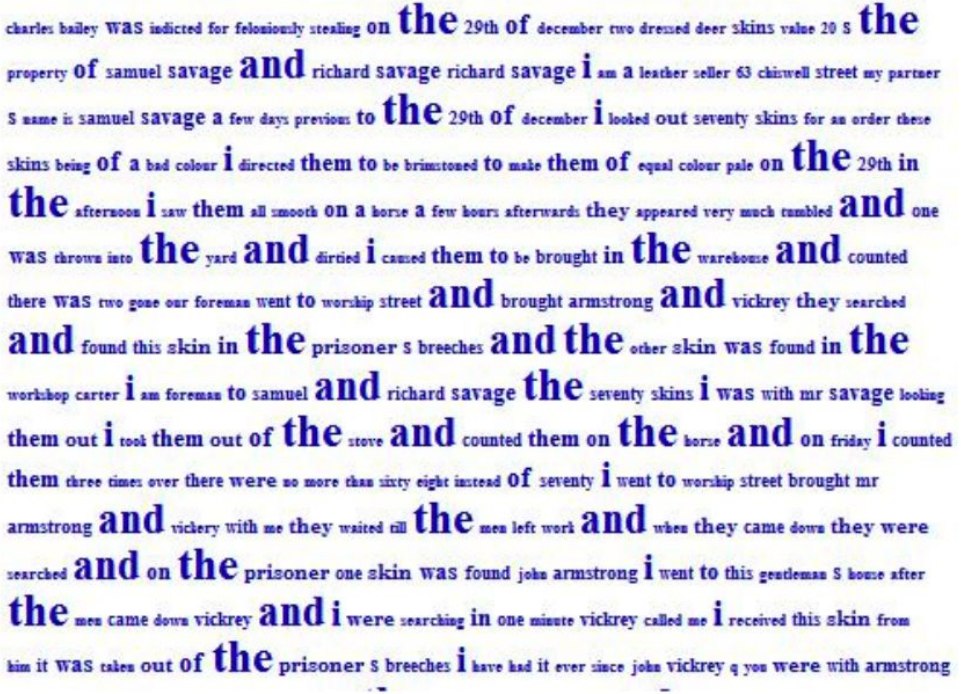
DOC2: "A friend in need is a friend indeed."

	he	friend	in	need	is	a	indeed
DOC1	1	1	1	1	1	1	0
DOC2	1	2	1	1	1	2	1

Représentation vectorielle de documents

- Problème du comptage brut : les mots « vides » sont les plus fréquents

et, ou, je, j’, le, la..



Représentation vectorielle de documents

- **Solution 1** : supprimer les mots vides (blacklist) => **text pre-processing**
  - **Suppression des mots vides** : Les mots vides sont un ensemble de mots couramment utilisés dans une langue. Des exemples de mots vides en anglais sont « and », « the », « is », « are » et etc. L'intuition derrière l'utilisation de mots vides est qu'en supprimant les mots à faible information du texte, nous pouvons nous concentrer sur les mots importants .
  - **Stemming** : La racine est le processus de réduction de l'inflexion des mots à leur forme racine. La "racine" dans ce cas peut ne pas être un vrai mot racine, mais juste une forme canonique du mot original.
    - *Connection, connections, connected, connecting => connect*
  - **Lemmatisation** : très similaire au **Stemming**, où le but est de supprimer les inflexions et de mapper un mot à sa forme racine. La seule différence est que la lemmatisation essaie de le faire correctement. Il ne fait pas que couper les choses, il transforme en fait les mots en la racine réelle.
    - Better => good

(lab work/TP)

Level of text preprocessing needed

	Domain Specific / Noisy Texts	General / Well Written Texts
Lots of data	- Moderate pre-processing - Text enrichment could be helpful	- Light pre-processing - Text enrichment could be helpful, but not critical
Sparse data	- Heavy pre-processing - Text enrichment is important	- Moderate pre-processing - Text enrichment could be helpful

By: Kavita Ganesan

Représentation vectorielle de documents

- **Solution 1** : supprimer les mots vides (blacklist) = **text pre-processing**
- **Solution 2** : pénaliser les mots qui apparaissent dans beaucoup de documents

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

`idf`$_i$ = inverse document frequency

`|D|` : nombre total de documents dans le corpus.
$|\{d_j : t_i \in d_j\}|$: nombre de document où le mot `t`$_i$ apparaît.
Représentation très utilisée : **`tf`$_i$ `*idf`$_i$**

Représentation vectorielle de documents: TF-IDF

- **T**erm **F**requency–**I**nverse **D**ocument **F**requency
- Applications :
  - Recherche d'information (Information Extraction)
  - Fouille de textes (Text Mining)
- Cette mesure statistique permet d'évaluer **l'importance d'un terme** contenu dans un document, **relativement à une collection** de textes.
- Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document.
- Il varie également en fonction de la fréquence du mot dans le corpus.

Représentation vectorielle de documents: TF-IDF

TF = $\frac{\text{Nombre de répétitions d'un mot dans un texte)}}{\text{Nombre de mots dans un texte}}$

IDF = $\log\left[\frac{\text{Nombre de textes/docs}}{\text{Nombre de textes contenant le mot}}\right]$

DOC1: "He is not a friend in need."

DOC2: "A friend in need is a friend indeed."

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

	TF		IDF	TF*IDF	
	DOC1	DOC2		DOC1	DOC2
he	1/7 = 0,14	0	log(2/1)=0,3	0,04	0
is	0,14	⅛=0,12	0	0	0
not	0,14	0	0,3	0,04	0
a	0,14	2/8 = 0,25	log(2/2)=0	0	0
friend	0,14	0,25	0	0	0
in	0,14	0,12	0,3	0,04	0,03
need	0,14	0,12	0,3	0,04	0,03
indeed	0	0,12	0,3	0	0,03

Représentation vectorielle de documents

Comment représenter les similarités de termes ?

L'altitude du Mont Blanc est 4 810 mètres.
La hauteur du Mont Blanc est de 4 810 mètres.

Dans une représentation vectorielle classique :

```
Distance (altitude, hauteur) = Distance (altitude, lavabo)
```

Aucune prise en compte de la proximité sémantique !

Représentation vectorielle de documents

Comment apprendre le sens des mots ?

1. • Approches par dictionnaires, ontologies
2. • Approches par corpus

Firth (1957): *”You shall know a word by the company it keeps!”*

→ **Apprentissage du sens d’un mot par ses contextes d’usage**

John Rupert Firth (1890-1960) était un linguiste anglais et un chercheur de premier plan en linguistique britannique dans les années 1950.

Représentation vectorielle de documents

“Tesgüino” ?

1. lac finlandais

2. boisson mexicaine

3. manga japonais

Nida, E. A. 1975. Componential analysis of meaning: An introduction to semantic structures.

Représentation vectorielle de documents

Tesgüino ?

Une bouteille de tesgüino est sur la table.

Le tesgüino est produit en Sierra Madre occidentale au Mexique.

Boire du tesgüino rend ivre.

On fabrique le tesgüino à partir de maïs.

Représentation du sens d'un mot par ses contextes d'usage

Nida, E. A. 1975. Componential analysis of meaning: An introduction to semantic structures.

Représentation vectorielle de documents

Matrice terme-document :
représenter les mots par les documents dans lesquels ils apparaissent

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Représentation vectorielle de documents

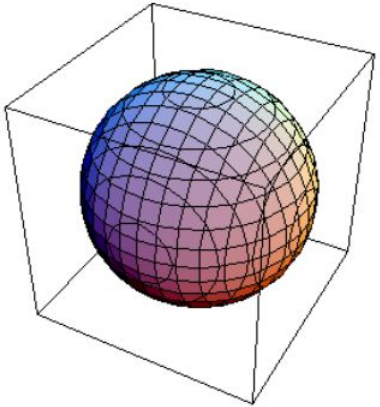
Limite : il faudrait beaucoup de documents pour bien représenter les mots

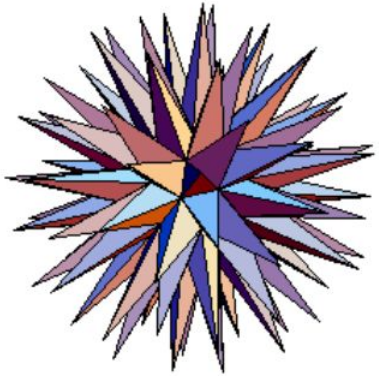
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
minors	0	0	0	0	0	0	0	1	1

- *human* et *user* ne partagent aucun contexte (document)
- *human* et *minors* non plus
- `distance (human, user) > distance (human, minors)`

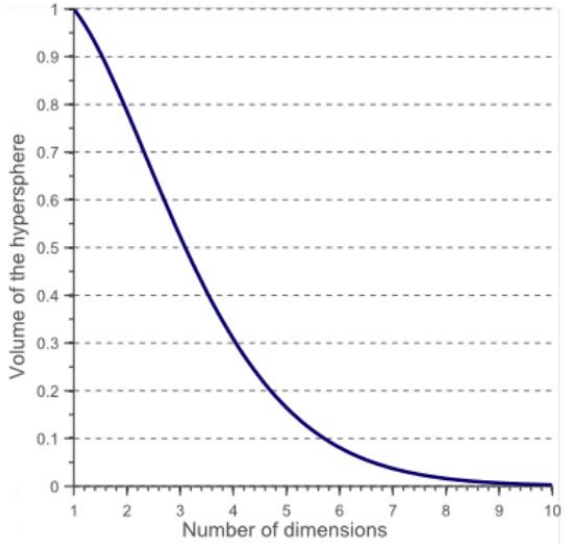
Malédiction de la dimensionalité

- Dans un espace en haute dimension, tous les points sont loins les uns des autres.
- Le nombre de données nécessaires pour couvrir l'espace augmente de manière exponentielle.





Malédiction de la dimensionalité



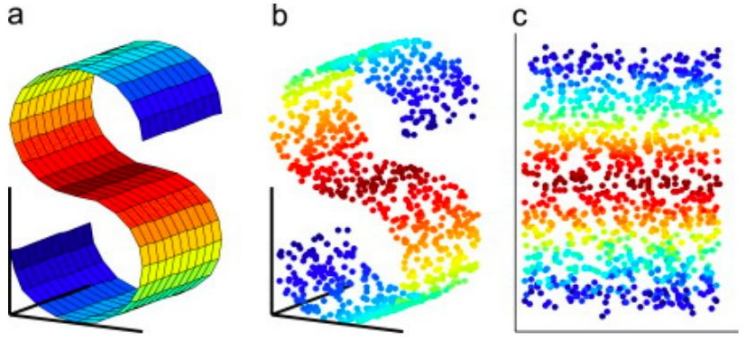
- Exemple : volume d'une sphère de rayon 0.5 inscrite dans un cube de coté 1

- dimension 2 :
  V cube = 1 V sphère = π/4
- dimension 3 :
  V cube = 1 V sphère = π/6
- dimension d :
  V cube = 1 V sphère = $\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}0.5^d.$

- Lorsque la dimension augmente, le volume de la sphère devient négligeable par rapport au volume du cube : tous les points de l'espace sont dans les coins, éloignés les uns des autres.
- Plus la dimension augmente, plus il faut de points pour couvrir l'espace et estimer les modèles

Réduction de dimension

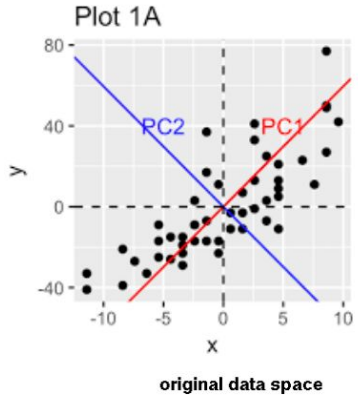
Intuition : en réalité, les points n'occupent pas uniformément tout l'espace



- les points sont représentés en 3 dimensions (Figures a. et b.)
- mais en réalité, ils sont disposés sur un sous-espace de dimension 2 (Figure c.)

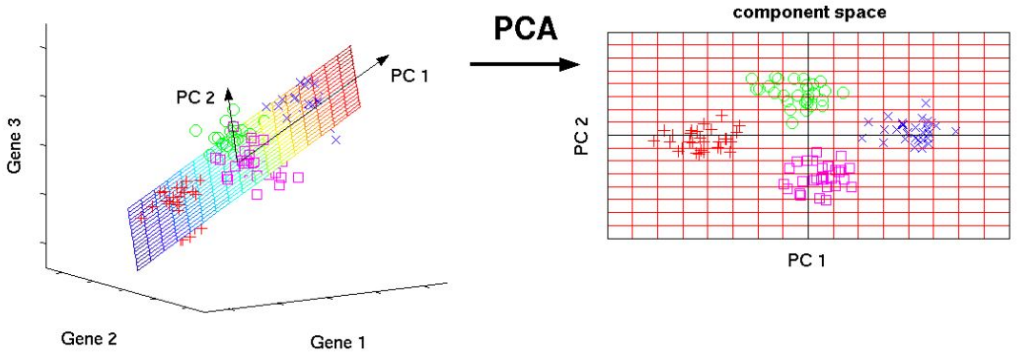
On peut donc conserver les relations entre les points tout en réduisant la dimension de l'espace de représentation

Réduction de dimension



Analyse en composantes principales (PCA)

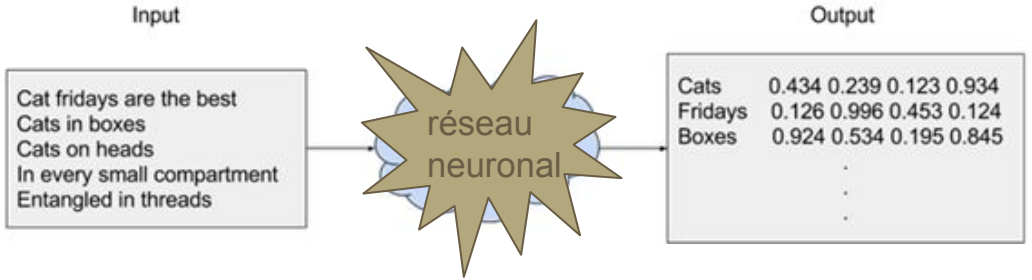
- Recherche des axes des composantes principales
- Sélection des composantes
- Projection des données sur les axes des composantes



LSI/SVD

Word embeddings (« plongements de mots »)

- Cette représentation permet de représenter chaque mot d'un dictionnaire par un **vecteur de nombres réels**.
- Les mots avec des contextes similaires possèdent des vecteurs qui sont relativement proches.
- Cette technique est basée sur l'hypothèse qui veut que les mots apparaissant dans des contextes similaires ont des significations apparentées :
  - « chien » et « chat » (animaux domestiques)
  - « samedi » et « dimanche » (jours dans une semaine)
  - « vin rouge », « bière » (boissor



Word embeddings (« plongements de mots »)

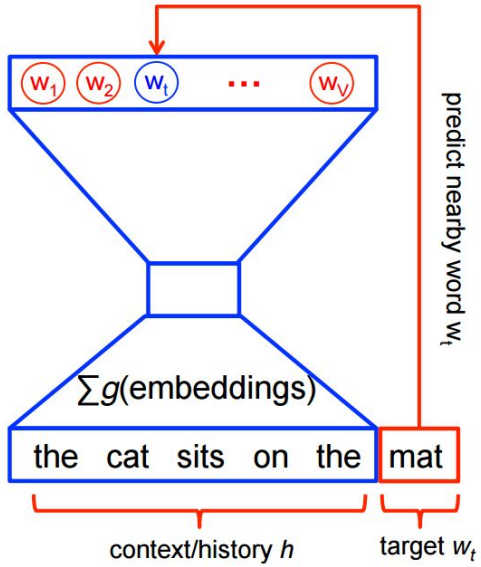
tous les mots du vocabulaire |V|=50,000

probabilités → .. p(autre mot|mat) **p(the|mat)** p(autre mot|mat)..

fonction d'activation pour normaliser la sortie d'un réseau en une distribution de probabilité sur des classes de sortie prédites → Softmax classifier

la plupart des calculs sont ici $W_x V_{the} + b$ → Hidden layer

→ Projection layer




```
Vocabulaire |V|=50,000

0   1 2  3    4   5   .. 50,000
the a in with cat dog ..
```

0 4 9 7 867 67

The cat sits on the mat

```
the → 0.34 0.4 0.11 0.5 0.89
cat → 0.6 0.23 0.8 0.87 0.21
dog → 0.77 0.21 0.09 0.29 0.05
...
```

Word Embeddings: GloVe (Global Vectors for Word Representation)

- Prise en compte des co-occurrences des mots pour la création de représentations vectorielles
- **Objectif** : trouver une représentation vectorielle qui conserve les ratios de fréquence de co-occurrence

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

$$P_{ij} = P(j|i) = \frac{X_{ij}}{Xi}$$

Word Embeddings: GloVe (Global Vectors for Word Representation)

- **Apprentissage** : minimisation de `J`

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

- `V` : taille du vocabulaire
- `f` : fonction de pondération
- b_i : permet de prendre en compte les différences de fréquence X_i
- W_i : représentation vectorielle

Recherche des représentation vectorielles des mots qui approchent le mieux $\log(X_{ij})$

- Optimisation par descente de gradient

Word Embeddings:

A Neural Probabilistic Language Model
Bengio et al., 2003

Modélisation statistique de la langue
`(running, walking)`, we could naturally generalize (i.e. transfer probability mass) from

The cat is walking in the bedroom

to

A dog was running in a room

and likewise to

The cat is running in a room
A dog is walking in a bedroom
The dog was walking in the room
...

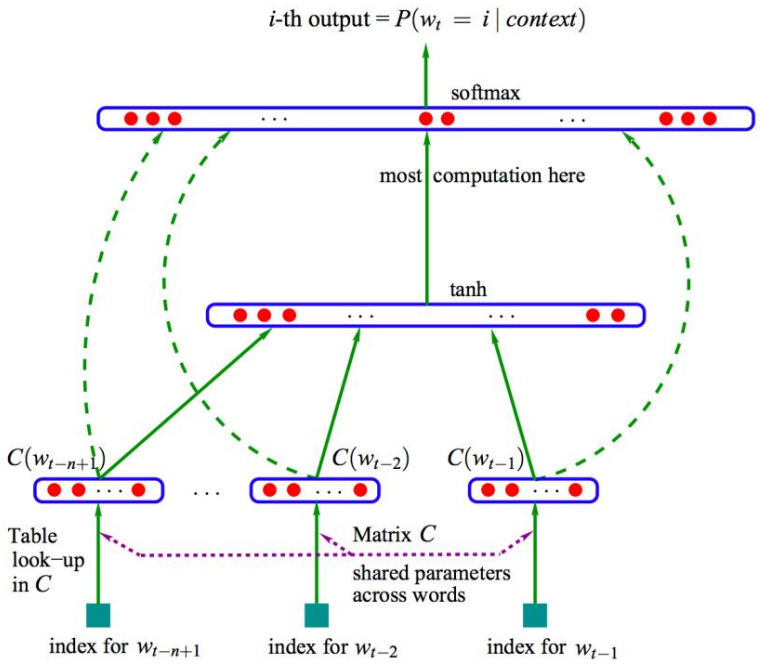
1.1 Fighting the Curse of Dimensionality with Distributed Representations

In a nutshell, the idea of the proposed approach can be summarized as follows:

1. associate with each word in the vocabulary a distributed *word feature vector* (a real-valued vector in $\mathbb{R}^m$),
2. express the joint *probability function* of word sequences in terms of the feature vectors of these words in the sequence, and
3. learn simultaneously the *word feature vectors* and the parameters of that *probability function*.

Word Embeddings:

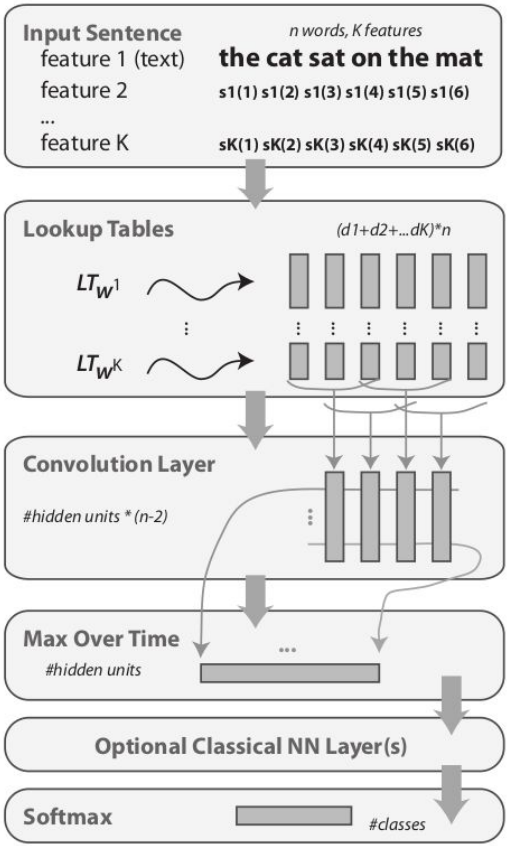
A Neural Probabilistic Language Model
Bengio et al., 2003



Word Embeddings:

A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, ICML 2008

Ronan Collobert et Jason Weston

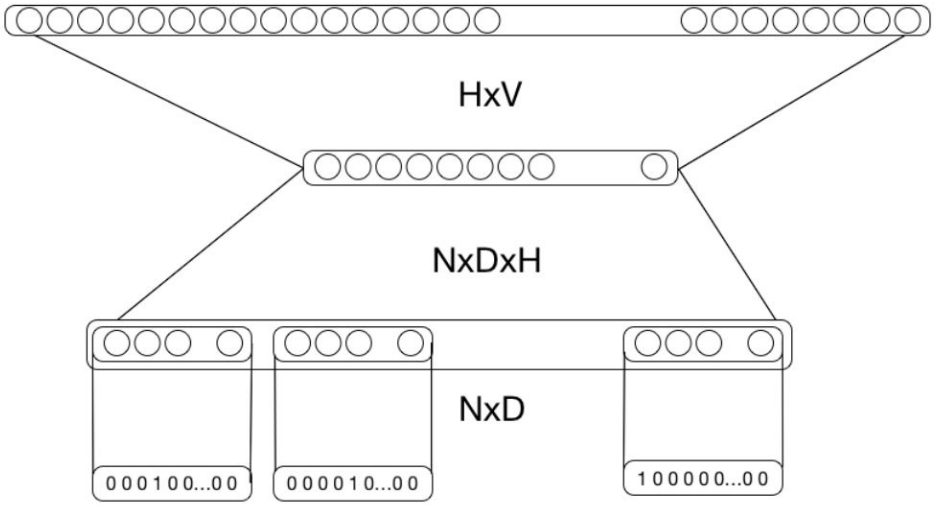


- Apprentissage des représentations (deep learning versus shallow features)
- Entrainement end-to-end versus features engineering + classifieur
- Pré-entraînement non supervisé (modèle de langue)
- Apprentissage supervisé mutlitâche

2018 International Conference on Machine Learning (ICML) “Test of Time Award”

Word Embeddings:

A Neural Probabilistic Language Model
Bengio et al., 2003



V = 100 000
D = 50 à 200
H = 500 à 1000

- HxV est énorme mais il existe des techniques d'accélération
- NxDxH reste problématique

Word Embeddings: Word2Vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

1 Introduction

Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data. An example is the popular N-gram model used for statistical language modeling - today, it is possible to train N-grams on virtually all available data (trillions of words [3]).

However, the simple techniques are at their limits in many tasks. For example, the amount of relevant in-domain data for automatic speech recognition is limited - the performance is usually dominated by the size of high quality transcribed speech data (often just millions of words). In machine translation, the existing corpora for many languages contain only a few billions of words or less. Thus, there are situations where simple scaling up of the basic techniques will not result in any significant progress, and we have to focus on more advanced techniques.

With progress of machine learning techniques in recent years, it has become possible to train more complex models on much larger data set, and they typically outperform the simple models. Probably the most successful concept is to use distributed representations of words [10]. For example, neural network based language models significantly outperform N-gram models [1, 27, 17].

1.1 Goals of the Paper

The main goal of this paper is to introduce techniques that can be used for learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary. As far as we know, none of the previously proposed architectures has been successfully trained on more

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Kai Chen
Greg Corrado
Jeffrey Dean

Proceedings of Workshop at ICLR, 2013.

Word Embeddings: Word2Vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

1 Introduction

Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data. An example is the popular N-gram model used for statistical language modeling - today, it is possible to train N-grams on virtually all available data (trillions of words [3]).

However, the simple techniques are at their limits in many tasks. For example, the amount of relevant in-domain data for automatic speech recognition is limited - the performance is usually dominated by the size of high quality transcribed speech data (often just millions of words). In machine translation, the existing corpora for many languages contain only a few billions of words or less. Thus, there are situations where simple scaling up of the basic techniques will not result in any significant progress, and we have to focus on more advanced techniques.

With progress of machine learning techniques in recent years, it has become possible to train more complex models on much larger data set, and they typically outperform the simple models. Probably the most successful concept is to use distributed representations of words [10]. For example, neural network based language models significantly outperform N-gram models [1, 27, 17].

1.1 Goals of the Paper

The main goal of this paper is to introduce techniques that can be used for learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary. As far as we know, none of the previously proposed architectures has been successfully trained on more

Objectif : étant donné un mot w_t dans un corpus de taille T, prédire les mots w_c qui peuvent apparaître dans son contexte :

$$p(w_c \mid w_t) = \frac{e^{s(w_t,\, w_c)}}{\sum_{j=1}^{W} e^{s(w_t,\, j)}}$$

Si s est une fonction de similarité entre mots, la probabilité d'un mot w_c conditionnellement à un autre mot w_t peut être calculée par :

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p(w_c \mid w_t)$$

Word Embeddings: Word2Vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

1 Introduction

Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data. An example is the popular N-gram model used for statistical language modeling - today, it is possible to train N-grams on virtually all available data (trillions of words [3]).

However, the simple techniques are at their limits in many tasks. For example, the amount of relevant in-domain data for automatic speech recognition is limited - the performance is usually dominated by the size of high quality transcribed speech data (often just millions of words). In machine translation, the existing corpora for many languages contain only a few billions of words or less. Thus, there are situations where simple scaling up of the basic techniques will not result in any significant progress, and we have to focus on more advanced techniques.

With progress of machine learning techniques in recent years, it has become possible to train more complex models on much larger data set, and they typically outperform the simple models. Probably the most successful concept is to use distributed representations of words [10]. For example, neural network based language models significantly outperform N-gram models [1, 27, 17].

1.1 Goals of the Paper

The main goal of this paper is to introduce techniques that can be used for learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary. As far as we know, none of the previously proposed architectures has been successfully trained on more

Le **problème** avec cette fonction objectif est le terme de normalisation : il nécessite de calculer s sur tous les mots

$$p(w_c \mid w_t) = \frac{e^{s(w_t,\, w_c)}}{\sum_{j=1}^{W} e^{s(w_t,\, j)}}$$

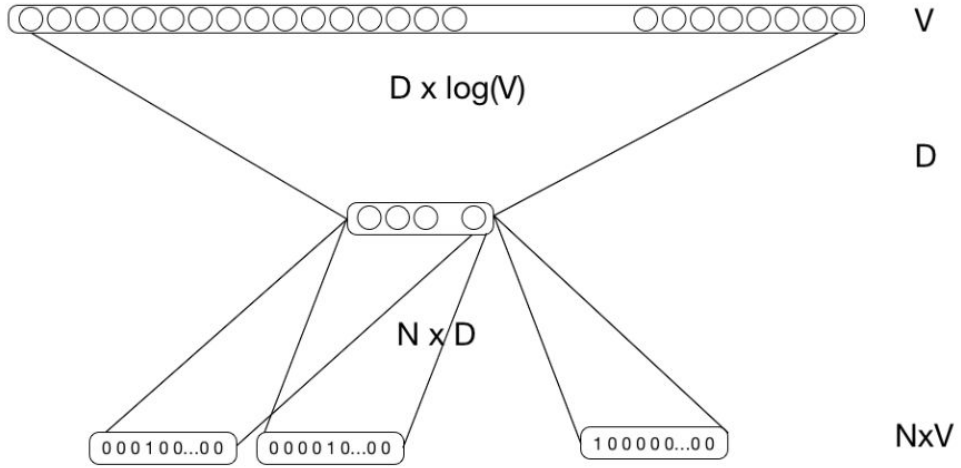
Word2Vec remplace donc cette fonction objectif par une tâche de classification : prédire si oui ou non un mot apparaît dans un contexte d'un mot donné.

$$\log\left(1 + e^{-s(w_t,\, w_c)}\right) + \sum_{n \in \mathcal{N}_{t,c}} \log\left(1 + e^{s(w_t,\, n)}\right)$$

Word Embeddings: Word2Vec

Approche 1 : Continuous Bag of word (CBOW)

- Supprimer la couche cachée
- Sommer les contextes

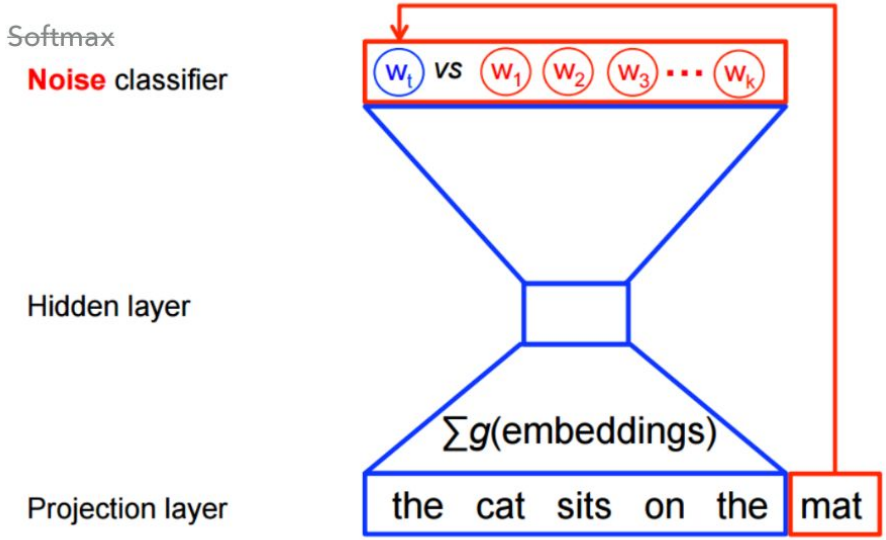


Prédiction du mot courant en fonction du contexte droit et gauche

Word Embeddings: Word2Vec

Approche 1 : Continuous Bag of word (CBOW); Paramètres du modèles :

- Exemples négatifs : nombre ?
- Embedding : taille ?
- Contexte : gauche/droite, taille ?

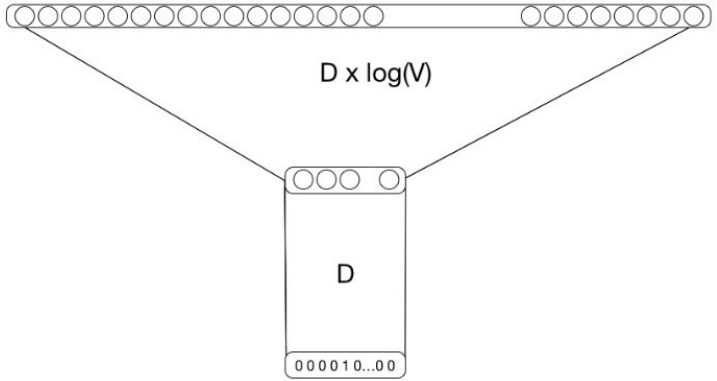


Prédiction du mot courant en fonction du contexte droit et gauche

Word Embeddings: Word2Vec

Approche 2 : Continuous Skip-Gram

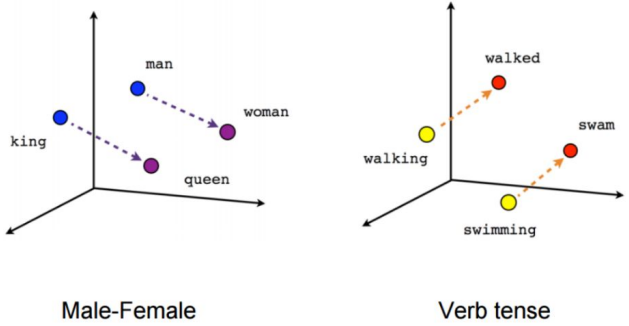
- Supprimer la couche cachée
- Prédire chaque mot du contexte à partir du mot courant



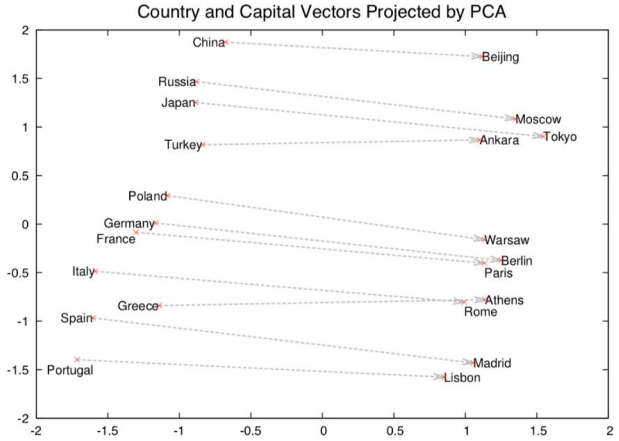
Prédiction pour C contextes

Word embeddings (« plongements de mots »)

- Relations sémantiques et géométriques



Male-Female Verb tense

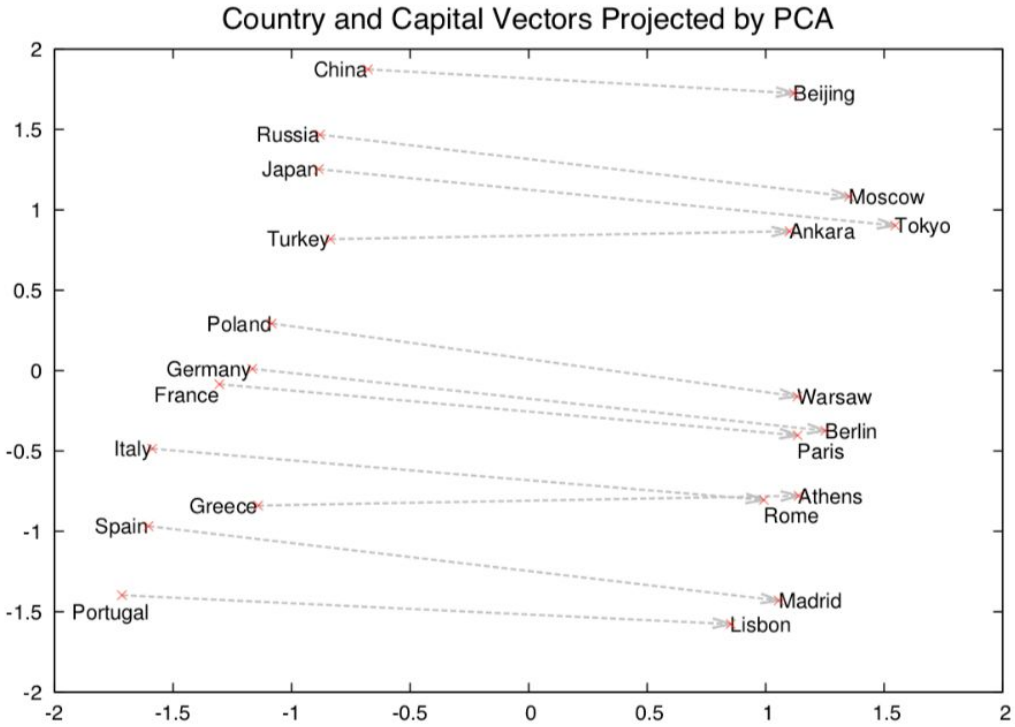


- Arithmétique vectorielle et sémantique

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

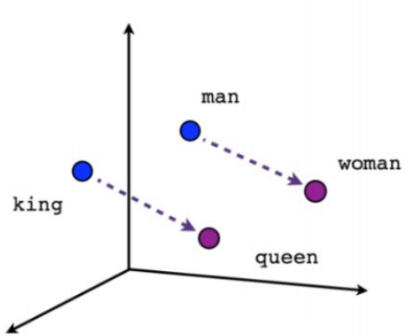
Word Embeddings: Word2Vec

Relations sémantiques et géométriques

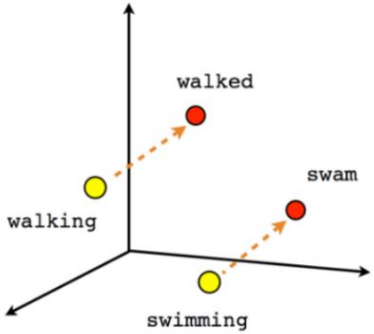


Word Embeddings: Word2Vec

Relations sémantiques et géométriques



Male-Female



Verb tense

Word Embeddings: Word2Vec

Arithmétique vectorielle et sémantique

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Word Embeddings: Word2Vec

Limites :

- Pas de représentation pour les mots inconnus :
  mots rares, nom propres, néologismes, fautes d'orthographe, argot, jargon, erreur de reconnaissance (OCR, parole)
- Pas de paramètres partagés pour les différentes formes fléchies d'un mot

```
mange / mangerai
cheval /chevaux
```

Word Embeddings: FastText

Enriching Word Vectors with Subword Information

Piotr Bojanowski* and Edouard Grave* and Armand Joulin and Tomas Mikolov
Facebook AI Research
{bojanowski,egrave,ajoulin,tmikolov}@fb.com

Abstract

Continuous word representations, trained on large unlabeled corpora are useful for many natural language processing tasks. Popular models that learn such representations ignore the morphology of words, by assigning a distinct vector to each word. This is a limitation, especially for languages with large vocabularies and many rare words. In this paper, we propose a new approach based on the skipgram model, where each word is represented as a bag of character n-grams. A vector representation is associated to each character n-gram; words being represented as the sum of these representations. Our method is fast, allowing to train models on large corpora quickly and allows us to compute word representations for words that did not appear in the training data. We evaluate our word representations on nine different languages, both on word similarity and analogy tasks. By comparing to recently proposed morphological word representations, we show that our vectors achieve state-of-the-art performance on these tasks.

1 Introduction

Learning continuous representations of words has a long history in natural language processing (Rumelhart et al., 1988). These representations are typically derived from large unlabeled corpora using co-occurrence statistics (Deerwester et al., 1990; Schütze, 1992; Lund and Burgess, 1996). A large body of work, known as distributional semantics, has studied the properties of these methods (Turney et al., 2010; Baroni and Lenci, 2010). In the neural network community, Collobert and Weston (2008) proposed to learn word embeddings using a feed-forward neural network, by predicting a word based on the two words on the left and two words on the right. More recently, Mikolov et al. (2013b) proposed simple log-bilinear models to learn continuous representations of words on very large corpora efficiently.

Most of these techniques represent each word of the vocabulary by a distinct vector, without parameter sharing. In particular, they ignore the internal structure of words, which is an important limitation for morphologically rich languages, such as Turkish or Finnish. For example, in French or Spanish, most verbs have more than forty different inflected forms, while the Finnish language has fifteen cases for nouns. These languages contain many word forms that occur rarely (or not at all) in the training corpus, making it difficult to learn good word representations. Because many word formations follow rules, it is possible to improve vector representations for morphologically rich languages by using character level information.

In this paper, we propose to learn representations for character n-grams, and to represent words as the sum of the n-gram vectors. Our main contribution is to introduce an extension of the continuous skipgram model (Mikolov et al., 2013b), which takes into account subword information. We evaluate this model on nine languages exhibiting different morphologies, showing the benefit of our approach.

*The two first authors contributed equally.

Enriching Word Vectors with Subword Information

Piotr Bojanowski
Edouard Grave
Armand Joulin
Tomas Mikolov

Transactions of the Association for Computational Linguistics, 2013

Word Embeddings: FastText

« Our main contribution is to introduce an extension of the continuous skip-gram model (Mikolov et at., 2013), which takes into account subword information»

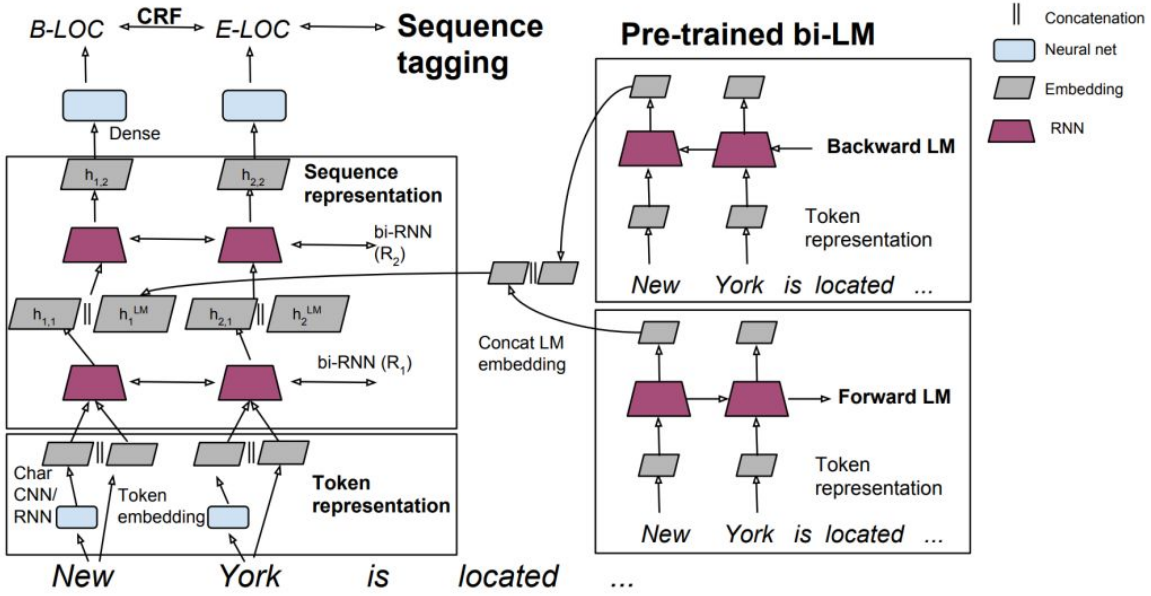
« Learn representations for character n-grams, and to represent words as the sum of the n-gram vectors.»

$$\sum_{t=1}^{T} \left[\sum_{c \in \mathcal{C}_t} \ell(s(w_t, w_c)) + \sum_{n \in \mathcal{N}_{t,c}} \ell(-s(w_t, n)) \right]$$

$$\ell : x \mapsto \log(1 + e^{-x})$$

$N_{t,c}$: ensemble d'exemples négatifs tirés du vocabulaire
C_t : mots du contexte du mot cible t

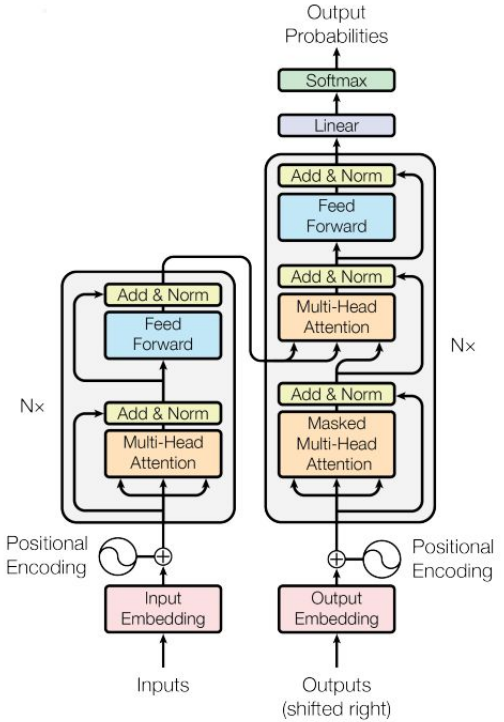
Word embeddings : modèles plus complexes



- Semi-supervised sequence tagging with bidirectional language models, Peters et al., ACL 2017
- Deep Contextualized word representation, Peters et al., NAACL 2018

Word embeddings : modèles plus complexes (+ des modèles d'attention)

Attention is all you need
Vaswani et al., NIPS 2017



→ Transformer

Word embeddings : problème de biais

Biais de position sur l'axe homme-femme

Man is to computer programmer as woman is to homemaker? Debiasing word embeddings

Bolukbasi et al., NIPS 2016

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

Biais d'analogie

Gender stereotype *she-he* analogies.

sewing-carpentry	register-nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	hairdresser-barber

Gender appropriate *she-he* analogies.

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Diviser les jeux de données

- Afin d'entraîner les modèles et évaluer la performance de ses modèles avec chaque représentation de mots, nous allons diviser les jeux de données en :
  - entraînement
  - développement (validation)
  - teste

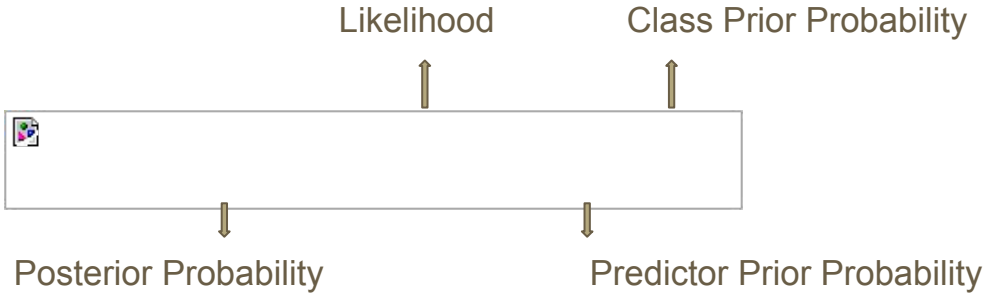
- Afin d'évaluer correctement la performance de chaque modèle, il est très important que les données d'entraînement et de teste soient différentes.

Approches de classification

- Actuellement, il y a plusieurs approches disponibles pour la classification de données.
- Certaines approches sont plus adaptées pour certains types et quantité de données.
- Par exemple, les réseaux neuronaux sont très populaires actuellement car ils ont battu la plupart de systèmes.
- Dans notre cours, nous irons analyser trois approches :
  - **Naïve Bayes**
  - **SVM**
  - **Logistic Regression**

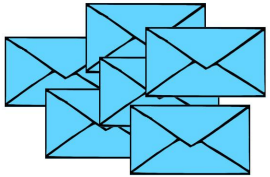
Naïve Bayes

- La classification naïve bayésienne est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance des hypothèses :

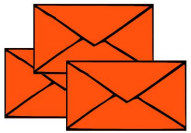


- Un classificateur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques.

Naïve Bayes – example filtrage de spam



6 NORMAL



3 SPAM

- Pré-traitement: on compte pas les mots vides (“a”, “in”, “the”, etc.)
- Données d'entraînement 6 NORMAL mails, 3 SPAM mails
- |V|=4, V={“hello”, “friend”, “book”, “money”}

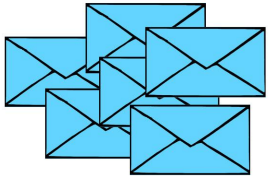
“hello” – 4 fois, “friend” – 3 fois, “book” – 2 fois, “money” – 1 fois, → 10 mots en total

P(“hello”|NORMAL)=4/10=0.40
P(“friend”|NORMAL)=3/10=0.30
P(“book”|NORMAL)=2/10=0.20;
P(“money”|NORMAL)=1/10=0.10

“hello” – 3 fois, “friend” – 2 fois, “book” – 0 fois, “money” – 4 fois, → 9 mots en total

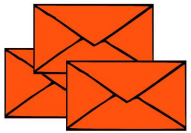
P(“hello”|SPAM)=3/9=0.33
P(“friend”|SPAM)=2/9=0.22
P(“book”|SPAM)=0
P(“money”|SPAM)=4/9=0.44

Naïve Bayes – example filtrage de spam



6 NORMAL

P(“hello”|N)=0,40
P(“friend”|N)=0,30
P(“book”|N)=0,20
P(“money”|N)=0,10



3 SPAM

P(“hello”|S)=0,33
P(“friend”|S)=0,22
P(“book”|S)=0
P(“money”|S)=0,44

Naïve = Nous supposons que chaque mot d'une phrase est indépendant des autres.

P(N|“hello friend”) = P(“hello friend”|N) x P(N)
P(“hello friend”)

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

P(“hello friend”|N) = P(“hello”|N) x P(“friend”|N)
= 0,40 x 0,30 = 0,12

P(N) = #Nombre N/#Nombre Total = 6/(6+3)=0,66

P(N|“hello friend”) = P(“hello friend”|N) x P(N)
=
= 0,12 x 0,66 = **0,07**
P(S|“hello friend”) = 0,33 x 0.22 x 0,33 = **0,02**

SVM : support vector machine

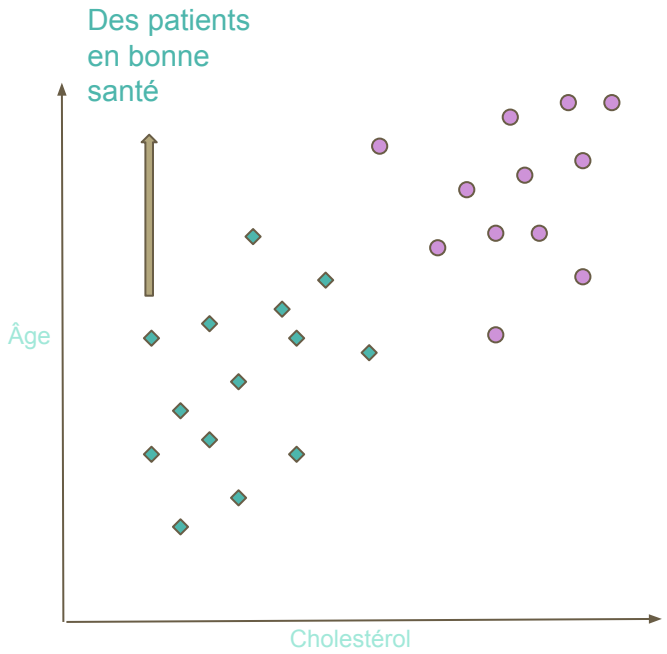
- Les SVMs sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de classification et de régression. Le but est de trouver un «hyperplan» qui pourrait séparer les données avec précision. Il pourrait y avoir de nombreux hyperplans de ce type → «hyperplan optimal» → **problème d'optimisation**
- Ces techniques reposent sur deux idées clés :
  - la notion de «marge» maximale : la distance entre la frontière de séparation («hyperplan») et les échantillons les plus proches («vecteurs support»)
  - la notion de fonction noyau («kernel trick») : les données qui ne sont pas linéairement séparables sont transformées pour être mappées dans un nouvel espace (2d → $\varphi((a, b)) = (a, b, a^2 + b^2)$ ← 3d, polynomial d=2)

- Les SVM **maximisent** la **marge** autour de l'**hyperplan de séparation**
- **Algorithme déterministe !**

SVM : support vector machine

SVM linéaire → Trouver une surface de décision linéaire («hyperplan») qui peut séparer les classes de patients et qui a la plus grande distance (c'est-à-dire le plus grand «écart» ou «marge») entre les patients à la frontière (c'est-à-dire les «vecteurs support»)

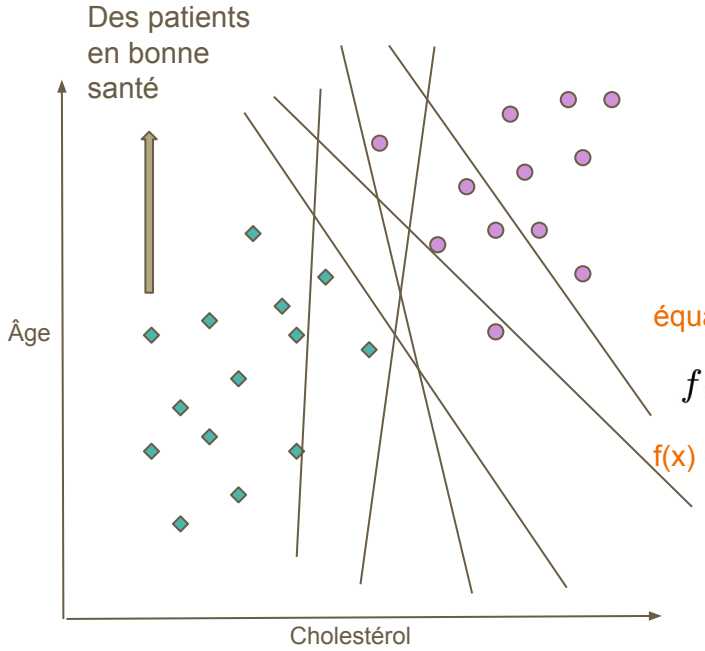
	Cholestérol	Âge	Classe
Patient 1	150	25	Pas malade
Patient 2	250	30	Malade
Patient 3	130	65	Malade
Patient 4 ...	350	45	Malade



SVM : support vector machine

SVM linéaire → Trouver une surface de décision linéaire («hyperplan») qui peut séparer les classes de patients et qui a la plus grande distance (c'est-à-dire le plus grand «écart» ou «marge») entre les patients à la frontière (c'est-à-dire les «vecteurs support»)

	Cholestérol	Âge	Classe
Patient 1	150	25	Pas malade
Patient 2	250	30	Malade
Patient 3	130	65	Malade
Patient 4 ...	350	45	Malade



équation d'une ligne

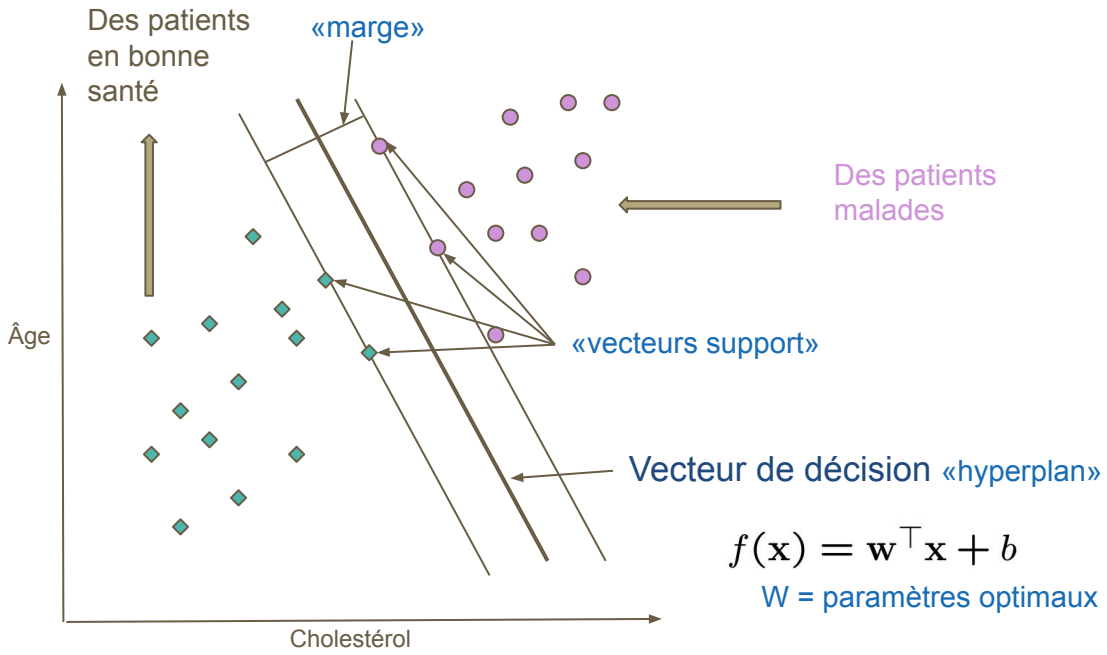
$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

f(x) = W1 x Âge + W2 x Cholestérol + b

SVM : support vector machine

SVM linéaire → Trouver une surface de décision linéaire («hyperplan») qui peut séparer les classes de patients et qui a la plus grande distance (c'est-à-dire le plus grand «écart» ou «marge») entre les patients à la frontière (c'est-à-dire les «vecteurs support»)

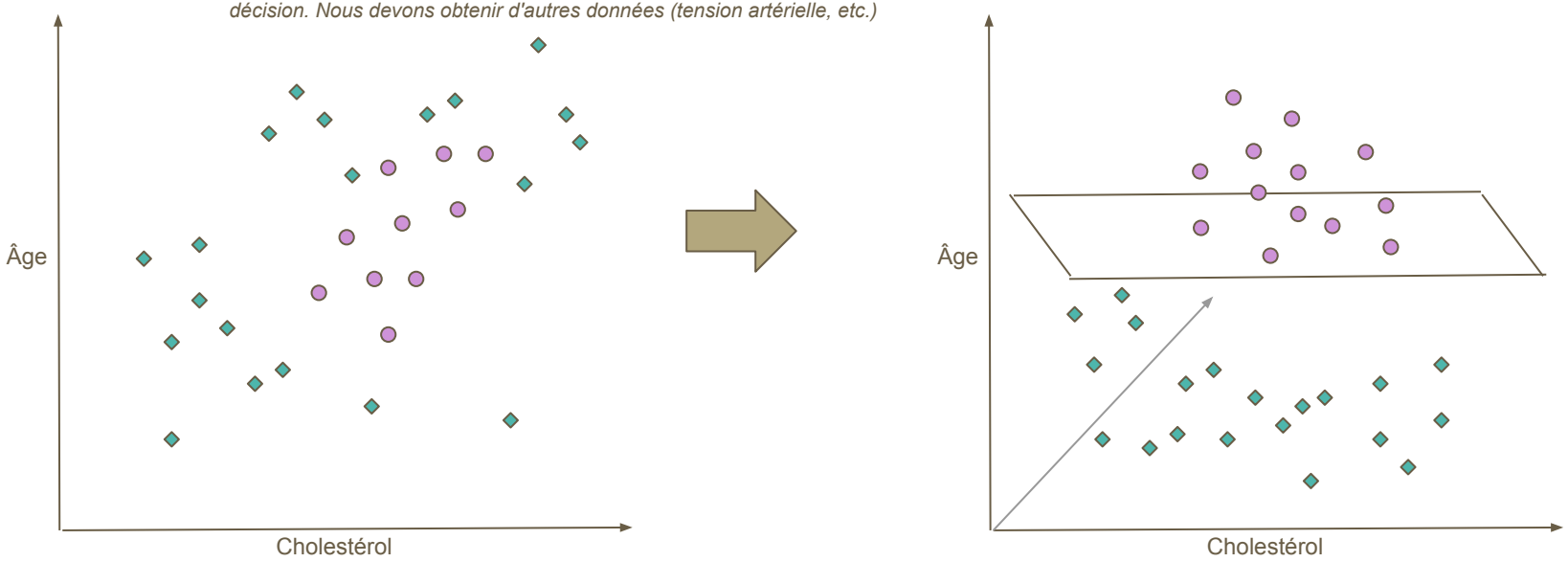
	Cholestérol	Âge	Classe
Patient 1	150	25	Pas malade
Patient 2	250	30	Malade
Patient 3	130	65	Malade
Patient 4 ...	350	45	Malade



SVM : support vector machine

Si une telle surface de **décision linéaire n'existe pas**, les données sont mappées dans un espace dimensionnel beaucoup plus élevé («espace de caractéristiques») où se trouve la surface de décision de séparation.
L'espace des caractéristiques est construit via une projection mathématique («kernel trick», kernel polynomial ou gaussien).

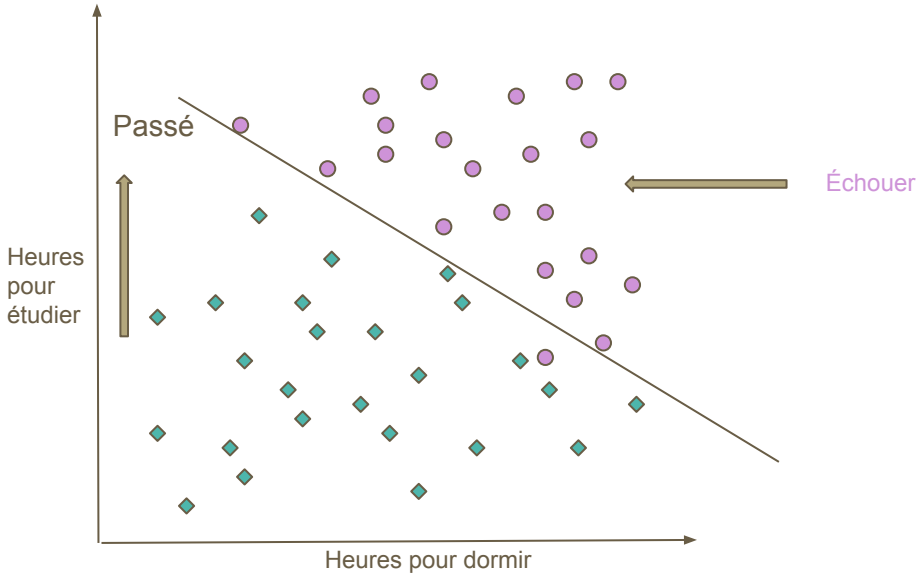
Cela peut également signifier que les caractéristiques ne sont pas suffisantes. L'âge et le cholestérol sont corrélés mais pas suffisants pour prendre une décision. Nous devons obtenir d'autres données (tension artérielle, etc.)



Régression logistique

- La **régression logistique** est un algorithme de classification qui transforme sa sortie à l'aide de la fonction **sigmoïde** logistique pour donner une valeur de probabilité pour les classes de sortie.
- ~ réseau neuronal à 1 couche
- **Algorithme statistique !**

	Heures pour dormir	Heures pour étudier	Classe
Student 1	8	7	Passé
Student 2	12	5	Échouer
Student 3	10	3	Échouer
Student 4 ...	9	8	Passé



Régression logistique

- La **régression logistique** à trouver les coefficients optimaux, de sorte que l'erreur globale (fonction de coût) soit minimisée, par descente de gradient. LR transforme sa sortie en utilisant la fonction sigmoïde logistique $\sigma = \frac{1}{1+e^{-z}}$ pour renvoyer une valeur de probabilité. $p \geq 0.5, class = 1$ $p < 0.5, class = 0$

Attribuer une probabilité à chaque résultat :

$$P(y = 1|x) = \sigma(w^T x + b)$$

S'entraîner pour maximiser les probabilités. LR une fonction de coût (perte) appelée Cross-Entropy

y = classe correcte, p = probabilité de la classe prédite $-(y \log(p) + (1 - y) \log(1 - p))$

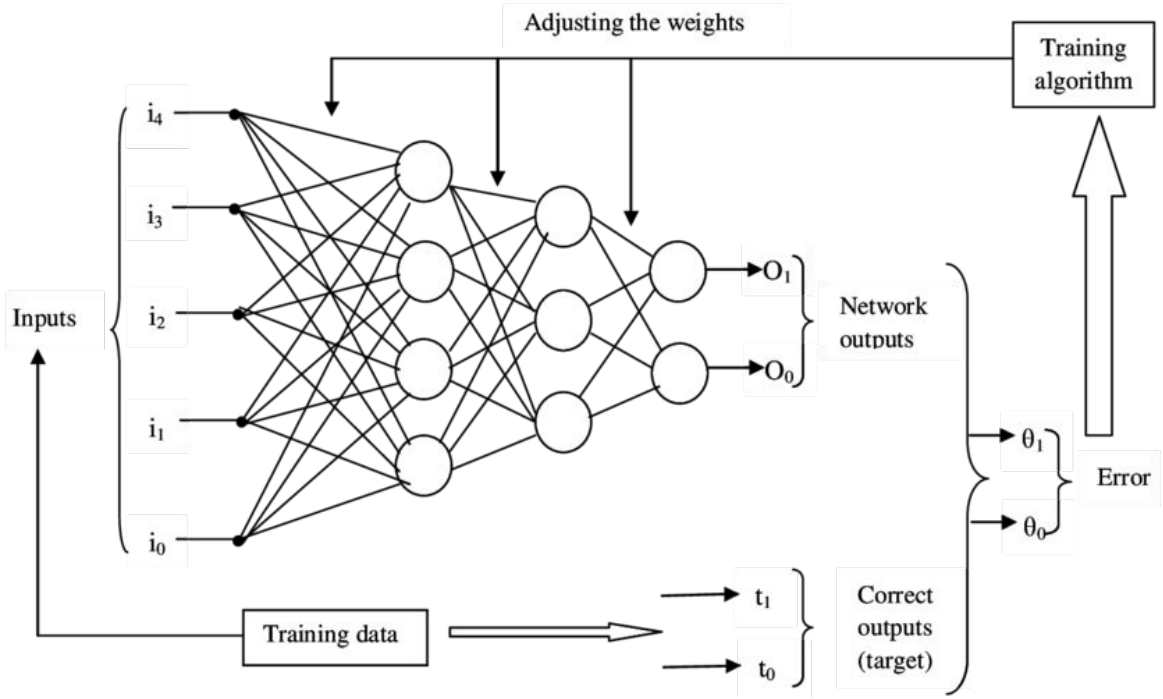
$$l(w) = -\sum_{n=1}^{N} \sigma(w^T x_n + b)^{y_n} (1 - \sigma(w^T x_n + b))^{(1-y_n)}$$

Pour minimiser les coûts (pertes ~ erreurs), nous utilisons la descente de gradient (Gradient Descent)

Descente de gradient (stochastique) connaît un très grand intérêt aujourd'hui, en particulier pour l'entraînement des réseaux de neurones profonds (deep learning).

Régression logistique

(descente de gradient)



Quand utiliser ces modèles

En fonction du nombre d'ensembles d'entraînement (données) / caractéristiques dont vous disposez, vous pouvez choisir d'utiliser la régression logistique, SVM ou Naïve Bayes.

Généralement, Naive Bayes est bon mais trop naïf et a de faibles performances.

Pour les autres algorithmes, considérons :

- `n` = nombre de caractéristiques
- `m` = nombre d'exemples (textes/images)

1. Si **`n`** est grand (1 à 10 000) et **`m`** est petit (10 à 1 000): utilisez la régression logistique ou SVM linéaire
2. Si **`n`** est petit (1 - 10 00) et **`m`** est intermédiaire (10 - 10 000): utilisez SVM avec noyau (gaussien, polynomial, etc.)
3. Si **`n`** est petit (1 - 10 00), **`m`** est grand (50 000 - 1 000 000 +): tout d'abord, ajoutez manuellement plus de caractéristiques, puis utilisez la régression logistique ou SVM linéaire
4. Si **`n`** est grand (1 à 10 000), m est grand (50 000 - 1 000 000 +): utiliser des réseaux de neurones

Il est généralement conseillé d'essayer d'abord d'utiliser la régression logistique pour voir comment fonctionne le modèle. S'il échoue, vous pouvez essayer d'utiliser SVM sans noyau (autrement appelé SVM avec un noyau linéaire). La régression logistique et SVM avec un noyau linéaire ont des performances similaires mais en fonction de vos fonctionnalités, l'une peut être plus efficace que l'autre.

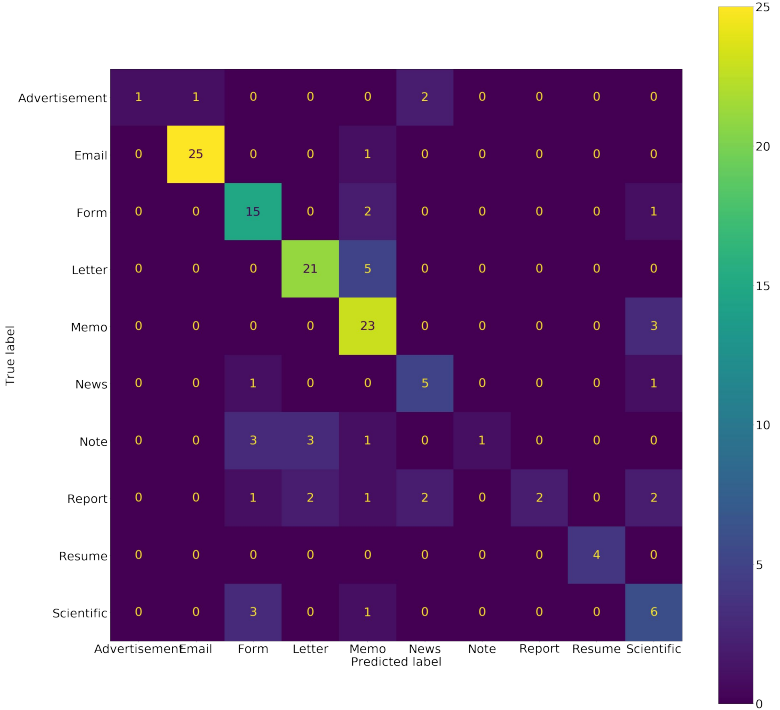
Analyse de la performance

- Après avoir entraîné notre modèle, nous pouvons faire la prédiction des classes des textes du jeux de données de test.
- La fonction f1-score calcule la performance d'une méthode à partir de l'analyse de la quantité de données qui ont été prédit correctement.

  - précision = la proportion des prédictions positifs était effectivement correcte
  - rappel = la proportion de résultats positifs réels a été identifiée correctement
  - F1 = la moyenne pondérée de la précision et du rappel (score final)

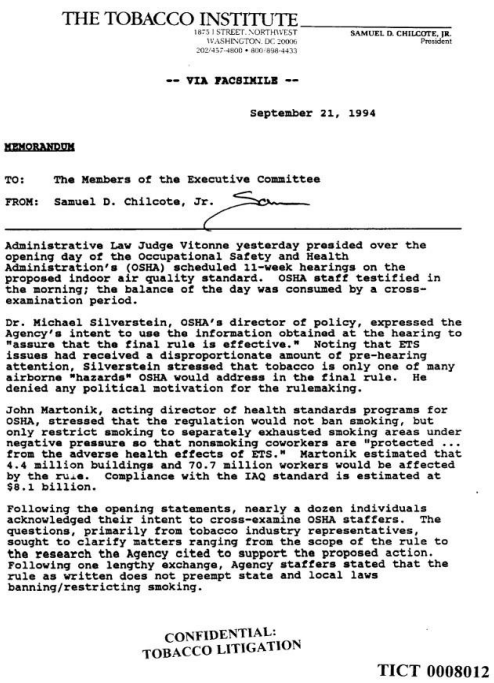
Matrice de confusion

- La matrice de confusion nous permettent de visualiser les résultats de la prédiction :
  - Vrai positifs
  - Faux positifs
  - Vrai négatifs
  - Faux négatifs



Jeux de données : Tobacco3482

Image



THE TOBACCO INSTITUTE

1875 I STREET, NORTHWEST
WASHINGTON, DC 20006
202/457-4800 • 800/898-4433

SAMUEL D. CHILCOTE, JR.
President

-- VIA FACSIMILE --

September 21, 1994

MEMORANDUM

TO: The Members of the Executive Committee

FROM: Samuel D. Chilcote, Jr.

Administrative Law Judge Vitonne yesterday presided over the opening day of the Occupational Safety and Health Administration's (OSHA) scheduled 11-week hearings on the proposed indoor air quality standard. OSHA staff testified in the morning; the balance of the day was consumed by a cross-examination period.

Dr. Michael Silverstein, OSHA's director of policy, expressed the Agency's intent to use the information obtained at the hearing to "assure that the final rule is effective." Noting that ETS issues had received a disproportionate amount of pre-hearing attention, Silverstein stressed that tobacco is only one of many airborne "hazards" OSHA would address in the final rule. He denied any political motivation for the rulemaking.

John Martonik, acting director of health standards programs for OSHA, stressed that the regulation would not ban smoking, but only restrict smoking to separately exhausted smoking areas under negative pressure so that nonsmoking coworkers are "protected ... from the adverse health effects of ETS." Martonik estimated that 4.4 million buildings and 70.7 million workers would be affected by the rule. Compliance with the IAQ standard is estimated at $8.1 billion.

Following the opening statements, nearly a dozen individuals acknowledged their intent to cross-examine OSHA staffers. The questions, primarily from tobacco industry representatives, sought to clarify matters ranging from the scope of the rule to the research the Agency cited to support the proposed action. Following one lengthy exchange, Agency staffers stated that the rule as written does not preempt state and local laws banning/restricting smoking.

CONFIDENTIAL:
TOBACCO LITIGATION

TICT 0008012

Texte

```
THE TOBACCO INSTITUTE

1875 | STREET, NORTHWEST SAMUEL D. CHILCOTE, JR.
WASHINGTON, DC 20006 President

202/457-4800 © 800/898-4433


@- VIA FACSIMILE --

September 21, 1994

TO: The Members of the Executive Committee


FROM: Samuel D. Chilcote, Jr.

Administrative Law Judge Vitonne yesterday presided over the
opening day of the Occupational Safety and Health
Administration's (OSHA) scheduled 111-week hearings on the
proposed indoor air quality standard. OSHA staff testified in

the morning; the balance of the day was consumed by a cross-
examination period.

Dr. Michael Silverstein, OSHA's director of policy, expressed the
Agency's intent to use the information obtained at the hearing to
"assure that the final rule is effective." Noting that ETS
issues had received a disproportionate amount of pre-hearing
attention, Silverstein stressed that tobacco is only one of many
airborne "hazards" OSHA would address in the final rule. He
denied any political motivation for the rulemaking.

John Martonik, acting director of health standards programs for
OSHA, stressed that the regulation would not ban smoking, but
only restrict smoking to separately exhausted smoking areas under
negative pressure so that nonsmoking coworkers are "protected ...
from the adverse health effects of ETS." Martonik estimated that
4.4 million buildings and 70.7 million workers would be affected

by the rule. Compliance with the IAQ standard is estimated at
$8.1 billion.

Following the opening statements, nearly a dozen individuals
acknowledged their intent to cross-examine OSHA staffers. The
questions, primarily from tobacco industry representatives,
sought to clarify matters ranging from the scope of the rule to
the research the Agency cited to support the proposed action.
Following one lengthy exchange, Agency staffers stated that the

rule as written does not preempt state and local laws
banning/restricting smoking.

CONFIDENTIAL:
TOBACCO LITIGATION

TICT 0008012
```

Le gouvernement américain a attaqué en justice cinq grands groupes américains du tabac pour avoir amassé d'importants bénéfices en mentant sur les dangers de la cigarette.

Dans ce procès 6 910 192 de documents ont été collectés et numérisés. Afin de faciliter l'exploitation de ces documents par les avocats, vous êtes en charge de mettre en place une classification automatique des types de documents: **Advertisement, Email, Form, Letter, Memo, News, Note, Report, Resume, Scientific.**

https://www.kaggle.com/competitions

Links

Machine learning Coursera famous courses, Andrew Ng, https://www.coursera.org/learn/machine-learning

Machine learning Coursera (on youtube), Andrew Ng, https://www.youtube.com/watch?v=PPLop4L2eGk

The most famous book on deep learning: https://www.deeplearningbook.org/ (Ian Goodfellow, Yoshua Bengio and Aaron Courville)

ML and DL People

Andrew Ng, Founder and CEO of Landing AI, Founder of deeplearning.ai.

Fei-Fei Li, Professor of Computer Science at Stanford University.

Andrej Karpathy, Senior Director of Artificial Intelligence at Tesla.

Demis Hassabis, Founder and CEO of DeepMind.

Ian Goodfellow, Director of Machine Learning at Apple.

Yann LeCun, Vice President and Chief AI Scientist at Facebook.

Jeremy P. Howard, Founding Researcher at fast.ai, Distinguished Research Scientist at the University of San Francisco.

Ruslan Salakhutdinov, Associate Professor at Carnegie Mellon University, Director of AI Research at Apple.

Geoffrey Hinton, Professor of Computer Science at the University of Toronto, VP and Engineering Fellow at Google

Rana el Kaliouby, CEO and Co-Founder of Affectiva.

Daphne Koller, Founder and CEO of insitro, Co-Founder of Coursera, Adjunct Professor of Computer Science and Pathology at Stanford.

Alex Smola, Director, Amazon Web Services.