Master Humanités Numériques

Machine Learning pour les données textuelles

Représenter les données textuelles

Julien Velcin
Laboratoire ERIC – Université Lyon 2
http://eric.univ-lyon2.fr/jvelcin

Représenter les données textuelles

Quelques définitions

Quelques définitions

- On appelle **document** un objet numérique qui véhicule un ensemble d'informations souvent structurées :

sans oublier les **méta-données** :
- auteur du document
- date de publication

etc.

- On appelle **corpus** un ensemble de documents. Le corpus est souvent associé à une structure (par ex. hyperliens, citations, etc.).

De multiples manières de représenter les données textuelles

- Comme une chaîne de caractères (*string*) :

  « Les humanités numériques peuvent être définies comme l'application du "savoir-faire des technologies de l'information [et de l'informatique/infosciences] aux questions de sciences humaines et sociales » (source : Wikipedia).

0	1	2	3	4	5	6	7	8	9
L	e	s		h	u	m	a	n	i

etc.

- Comme un sac de mots (*bag of words*) ou de termes :
- Comme une séquence de mots ou *tokens* :
- Comme un vecteur dans un espace vectoriel :
- Comme une matrice, un arbre, un graphe, etc. (non présenté dans le du cours)

De multiples manières de représenter les données textuelles

- Comme une chaîne de caractères (*string*) :
- Comme un sac de mots (*bag of words*) ou de termes :

- Comme une séquence de mots ou *tokens* :
- Comme un vecteur dans un espace vectoriel :
- Comme une matrice, un arbre, un graphe, etc.
  (non présenté dans le du cours)

De multiples manières de représenter les données textuelles

- Comme une chaîne de caractères (*string*) :
- Comme un sac de mots (*bag of words*) ou de termes :
- Comme une séquence de mots ou *tokens* :

0	1	2	3	4	5	6	7	
Les	humanités	numériques	peuvent	être	définies	comme	l’	etc.

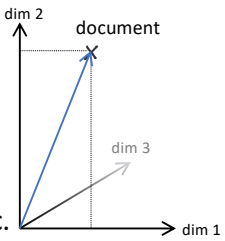
un token

- Comme un vecteur dans un espace vectoriel :
- Comme une matrice, un arbre, un graphe, etc.
  (non présenté dans le du cours)

De multiples manières de représenter les données textuelles

- Comme une chaîne de caractères (*string*) :
- Comme un sac de mots (*bag of words*) ou de termes :
- Comme une séquence de mots ou *tokens* :
- Comme un vecteur dans un espace vectoriel :

Les dimensions de l’espace (dim1, dim2…) sont, par exemple, les mots possibles dans le vocabulaire (ici : humanités, numériques, et, savoir-faire, etc.)



- Comme une matrice, un arbre, un graphe, etc.
  (non présenté dans le du cours)

Représenter les données textuelles

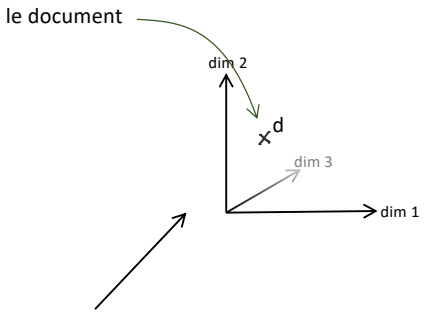
Motivation

Encoder le texte

- On **transforme** le texte (ou on le « projette », ou on le « plonge ») dans une représentation informatique qui cherche à capturer le **sens** du texte :

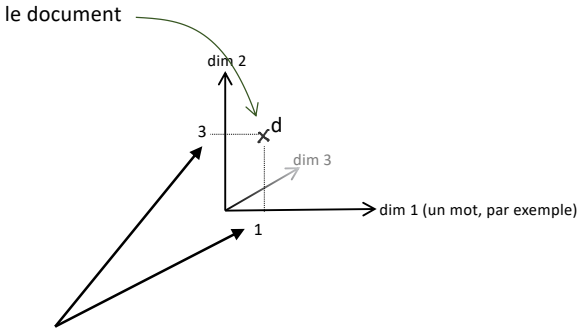
(1.0, 0.0, 0.2, -0.3…, 2.7)

Représentation d'un document



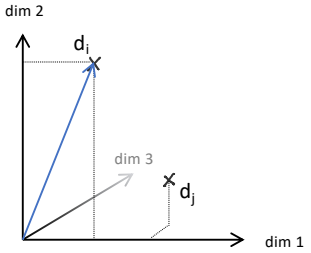
espace mathématique (le plus souvent vectoriel) multi-dimensionnel (ici à 3 dimensions pour illustrer)

Représentation d'un document



Les scores sont, par exemple, le nombre de fois où un mot est employé dans un document (TF)

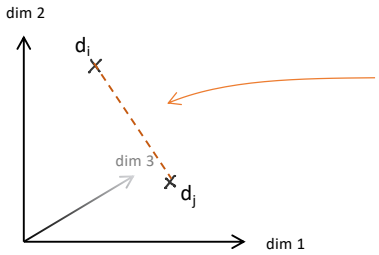
Représenter **d** dans un espace vectoriel (VSM de Salton)



Les axes peuvent être :
- des mots

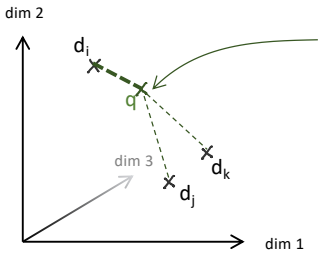
mais aussi :
- des thématiques
- des variables latentes

Comparer dans un espace vectoriel



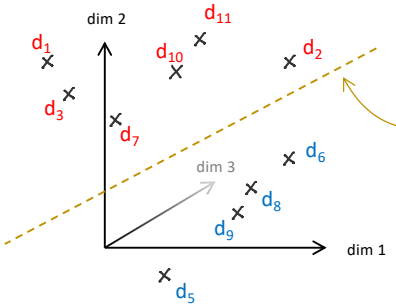
Différentes sortes de **mesures** :
- distance euclidienne
- produit scalaire
- cosinus
...

Comparer dans un espace vectoriel



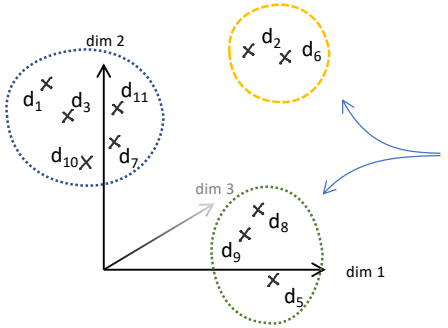
On peut considérer une **requête q** comme un pseudo-document et donc trouver les documents les plus proches

Classer dans un espace vectoriel



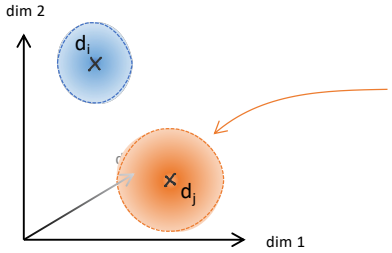
On peut chercher la **frontière** entre les classes dans cet espace (ici séparation linéaire)

Catégoriser dans un espace vectoriel (*clustering*)



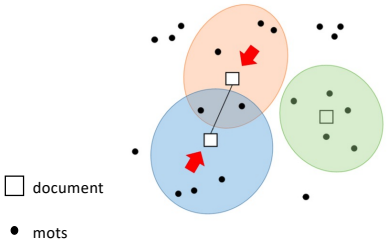
On peut chercher des **catégories** de manière non supervisée (*clustering*)

Aller plus loin : prise en compte de l'incertitude



On peut représenter un objet (document, auteur...) comme une **distribution de probabilité** dans l'espace

Aller plus loin : prise en compte de méta-données



On rapproche les représentations de documents qui sont **connectés** (relation de citation, même auteur...)

Document Network Projection in Pretrained Word Embedding Space
A. Gourru, J. Velcin, J. Jacques and A. Guille. ECIR 2020.
https://github.com/AntoineGourru/DNEmbedding

Représenter les données textuelles

Représentations classiques

Représentation classique : la matrice Documents x Termes (1)

- On transforme la chaîne de caractères pour construire un tableau où on compte les mots :

I love holidays. Sunbathing, swimming… I cannot imagine being away from the sea during holidays. Going to the mountain is not the same. I do not know…. I think the mountain is better for winter holidays and the sea for the summer ones.

word	*Frequency*
I	4
love	1
holidays	3
…	
sea	2
for	2
the	6
summer	1
ones	1

Term Frequency (TF) = nombre d'occurrences (le plus souvent)

L'ordre des mots ne compte pas ! cf. hypothèse du « sac de mots »

Représentation classique : la matrice Documents x Termes (2)

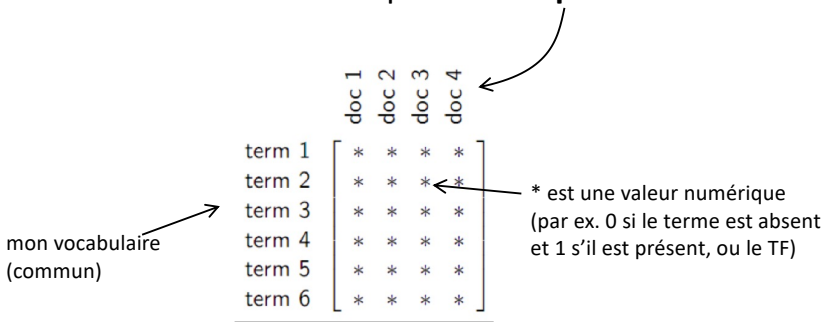
Cela possède bien sûr des limitations :

« Mary asked Fred out »

word	*Frequency*
Mary	1
asked	1
Fred	1
out	1

Représentation classique : la matrice Documents x Termes (3)

- On calcule cette matrice pour un **corpus** :



Représentation classique : la matrice Documents x Termes (4)

Voilà un exemple :

Docs	amp	brexit	euref	leav	remain	strongerin	vote	voteleav
738102860454498304	2	1	1	2	0	0	1	1
739933062281187329	0	0	1	2	2	0	1	0
745289444006170624	0	0	0	1	1	0	4	0
745501761289355264	0	0	0	0	7	0	0	0
745621915516149760	0	1	1	1	1	0	2	0
745649059231215616	1	0	0	1	1	1	2	0
745875415839965184	2	0	1	0	1	0	2	0
745922585494429697	1	0	1	0	1	1	2	0
745973624142725120	2	0	0	1	1	1	1	0
746108821479821312	0	0	1	0	4	0	1	0

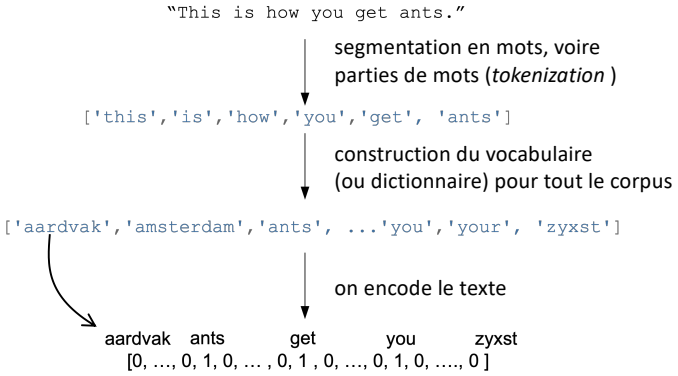
…et un autre avec une matrice Termes x Documents

	D1	D2	D3	D4	D5	D6	D7	D8	D9
measur	1	0	0	2	1	0	1	0	0
effici	1	0	1	0	0	0	1	0	0
machin	1	0	0	0	0	0	0	0	0
factori	1	0	0	0	0	0	0	0	0
system	1	0	0	0	0	0	0	0	0
input	1	0	0	2	0	0	1	0	1
output	1	1	0	2	0	0	1	0	1
averag	0	1	0	0	0	0	1	0	0
cost	0	1	1	0	0	0	0	0	0
resourc	0	1	0	0	0	0	0	0	0
consum	0	1	0	0	0	0	0	1	0
econom	0	0	0	1	0	0	0	0	0
labor	0	0	0	1	0	0	0	0	0
revenu	0	0	0	1	0	0	0	0	0
gdp	0	0	0	1	1	0	0	0	0
predict	0	0	0	0	1	0	0	0	0
futur	0	0	0	0	1	0	0	0	0
growth	0	0	0	0	1	0	0	0	0
gain	0	0	0	0	0	1	0	0	0
accomplish	0	0	0	0	0	1	0	0	0
energi	0	0	0	0	0	0	0	1	0
produc	0	0	0	0	0	0	0	1	0
food	0	0	0	0	0	0	0	1	0

On observe que c'est le TF qui est utilisé ici (d'autres pondérations sont possibles)

Une chaîne de traitement

- Voilà typiquement la chaîne qu'on applique :



Schémas de pondération

- Le score qu'on attribue à un mot (ou partie de mot, ou terme) pour un document peut varier :
  - présence ou absence : 0 ou 1
  - nombre d'occurrences (**TF**) : valeur entière (0, 1, 2, 3…)
  - …normalisé par la longueur du texte : TF / #mots
  - **TFxIDF** : prise en compte de la rareté des mots

    Score TFxIDF(terme t, document d) = $TF(t,d) \times idf(t)$

    où $idf(t) = \log N / df(t)$, N = nombre de documents

    et df(t) = nombre de documents contenant t
  - **OKAPI BM25** : variante de TFxIDF basé sur un modèle probabiliste de pertinence

Limitations du modèle classique

- Les matrices qui encodent l'information d'un corpus sont très grandes et **creuses** (*sparse*)
- La similarité entre deux documents (calculée par ex. avec une mesure de cosinus, cf. slide ultérieur) se base sur une correspondance exacte
  (par ex. « bateau » et « bateaux » sont des mots aussi différents que « bateau » et « poisson »)
- En conséquence, deux textes *similaires* en sens mais employant des termes *différents* seront considérés comme éloignés

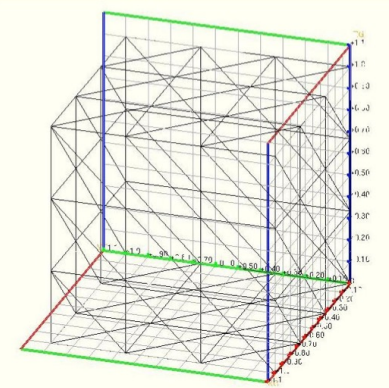
Quelques prétraitements standards

- Les prétraitements permettent souvent de réduire la taille du vocabulaire et de rendre les traitements aval plus robustes aux mots choisis
- Quelques prétraitements :
  - mise en minuscule
  - suppression de la ponctuation, des nombres…
  - suppression des mots-outils (*stopwords*)
  - suppression des mots trop fréquents ou trop rares (par ex. les *hapax* qui n'apparaissent qu'une fois)
  - racinisation (*stemming*) et lemmatisation

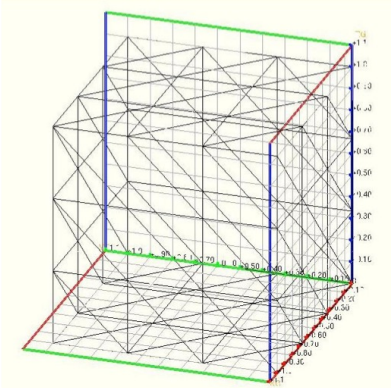
Comparer des textes (1)

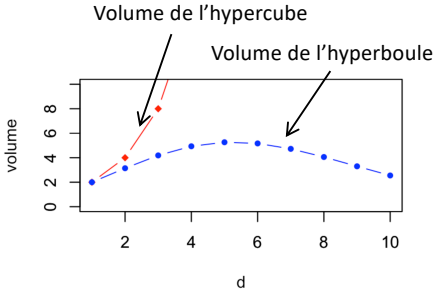
- Avec l'approche classique, les distances usuelles (ex. euclidienne) ne sont pas adaptées.
- Dans les espaces à **beaucoup de dimensions** :
  - Pourquoi les banquiers n'ont jamais de lingots sphériques ?
  - Pourquoi les marchands d'oranges occupent beaucoup de place pour empiler peu d'oranges ?
- Voir la partie « curiosités du calcul » de http://www.brouty.fr/Maths/sphere.html
- En lien avec ce qu'on appelle la « malédiction de la dimension » (*curse of dimensionality*)
- Richard E. Bellman (1920-1984): les hypervolumes sont presque **vides** !

Comparer des textes (2)



Comparer des textes (2)





Un volume avec dim=d a besoin de 10^d données pour peupler équitablement l'espace.

Produit scalaire et cosinus

- Produit scalaire : $\mathbf{x}.\mathbf{y} = \sum_{i=0}^{i=n} x_i y_i = x_0 * y_0 + x_1 * y_1 \ldots + x_n * y_n$

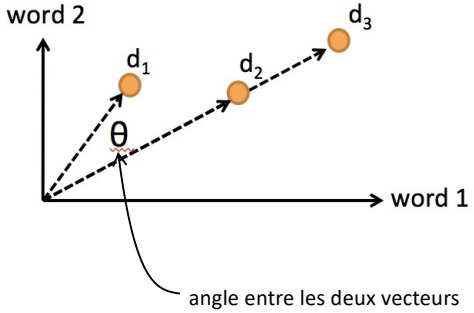
	mot1	mot2	mot3	mot4	mot5	mot6	...	motn
d_1	0	2	0	0	2	0		0
d_2	1	3	1	0	1	0		1

- Mesure du cosinus : $cosine(\mathbf{x},\mathbf{y}) = \underbrace{\frac{\mathbf{x}.\mathbf{y}}{\|\mathbf{x}\|_2.\|\mathbf{y}\|_2}}$

on normalise le produit scalaire

Avec des vecteurs positifs ou nuls (c'est le cas avec TF), la mesure s'étend de 0 (documents sans mots communs) et 1 (similarité maximum)

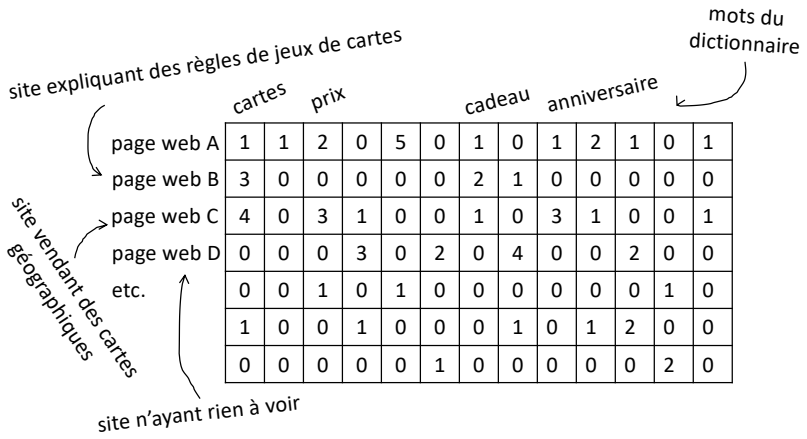
Cosinus : interprétation géométrique



Algorithme d'un moteur de recherche classique

- Indexer la BD pour obtenir une matrice **X** de format Documents x Termes
- Représenter la requête **q** dans le même espace (ie avec le même vocabulaire)
- Pour chaque document **d** de **X** :
  - calculer la similarité cos(d,q)
- Trier les documents en fonction de la similarité
- Afficher les **K** premiers documents de la liste

Illustration



page web A	1	1	2	0	5	0	1	0	1	2	1	0	1
page web B	3	0	0	0	0	0	2	1	0	0	0	0	0
page web C	4	0	3	1	0	0	1	0	3	1	0	0	1
page web D	0	0	0	3	0	2	0	4	0	0	2	0	0
etc.	0	0	1	0	1	0	0	0	0	0	0	1	0
	1	0	0	1	0	0	0	1	0	1	2	0	0
	0	0	0	0	0	1	0	0	0	0	0	2	0

Représenter les données textuelles

Représentation du sens des mots et représentations denses

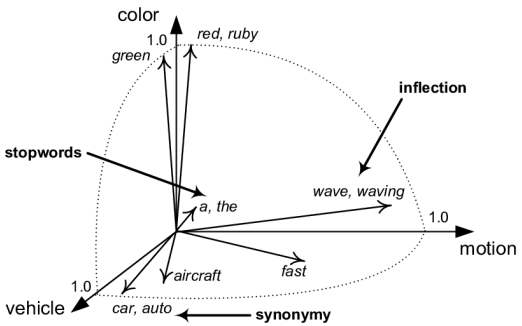
Retour sur la limitation du modèle classique

- Prenons des mots du Petit Larousse illustré de 2007 avec leur numéro dans l'ordre d'apparition :

Mot	N°
BAVARDE	8955
BAVARDER	8956
BAVAROIS	8957
ECOUTER	27959
ENTENDRE	30439

- Qu'observons-nous ?

Coder le sens des mots…



…permet de capturer le sens

- Document n°1 :
  d1 : « Il n'arrivait pas à se faire entendre. »

  ~~N° du mot : 30439~~ ➡ (0.3, 0, -1.2, 0, 0, 0.1…)

- Document n°2
  d2 : « Il fallait bien écouter dans ce cours. »

  ~~N° du mot : 27959~~ ➡ (0.28, 0, -1.1, 0, 0.1, 0…)

…permet de capturer le sens

- Document n°1 :
  d1 : « Il n'arrivait pas à se faire entendre. »

  ~~N° du mot : 30439~~ ➡ (0.3, 0, -1.2, 0, 0, 0.1…)

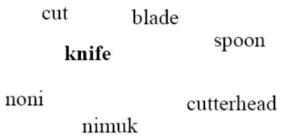
- Document n°2
  d2 : « Il fallait bien écouter dans ce cours. »

  ~~N° du mot : 27959~~ ➡ (0.28, 0, -1.1, 0, 0.1, 0…)

On peut calculer une distance ou une similarité relative au sens des mots entre ces vecteurs

Sémantique distributionnelle

- Deux linguistes sont souvent cités :
  Harris (1954) : Des mots apparaissant dans des contextes similaires ont des sens proches
  Firth (1957) : *« You shall know a word by the company it keeps »*



Par exemple

A bottle of ***tesgüino*** is on the table
Everybody likes ***tesgüino***
Tesgüino makes you drunk
We make ***tesgüino*** out of corn.

(tiré du cours de D. Jurafksy à Stanford)

Pouvez-vous deviner ce qu'est le « tesgüino » ?

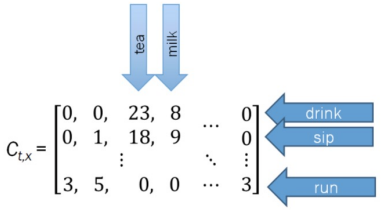
Encoder le sens des mots

- Une approche simple : prendre comme contexte les documents où apparaissent les mots :

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

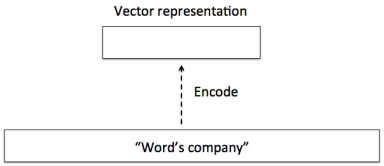
Figure 15.1 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

- ou bien partir de la matrice de co-occurrences :



Limitations de cette approche

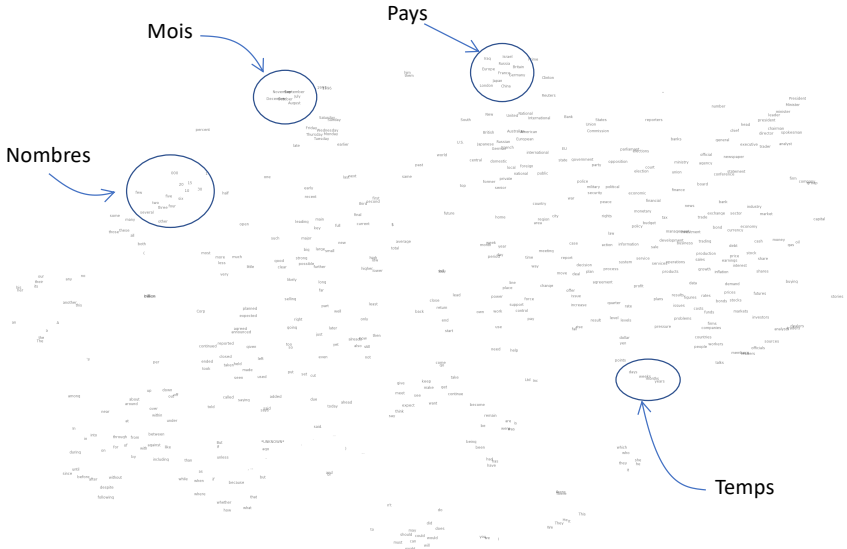
- Elles ressemblent à celles de la représentation classique des documents à partir des mots :
  - vecteurs très grands et **creux** (beaucoup de 0)
  - correspondance **exacte** entre les mots du contexte (par ex. on ne prend pas en compte les synonymes)
- La solution consiste à calculer des représentations denses avec des dimensions plus informatives



= contexte du mot

Word embedding

- L'objectif du plongement de mots (*word embedding*) est donc de calculer une **représentation dense** du sens des mots dans un espace vectoriel
- Dans cet espace, les mots proches au sens *géométrique* seront aussi proches dans leur *sémantique*



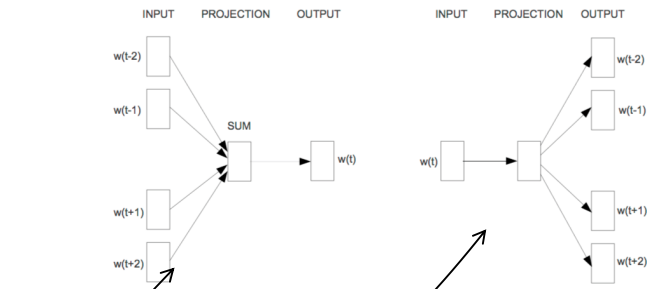
Représenter les données textuelles

Apprendre des représentations de mots

Apprendre des représentations statiques

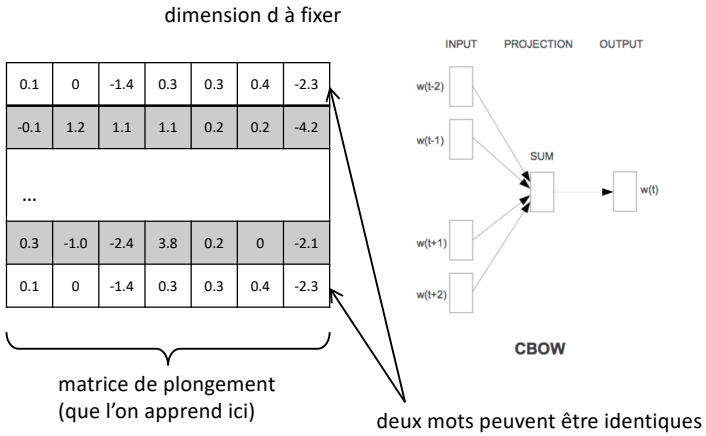
- Différentes approches qui font l'hypothèse d'un vecteur *unique* pour chaque mot du vocabulaire :
  - Word2Vec (Mikolov et al., 2013)
  - FastText (Bojanowski et al., 2017)
  - Glove (Pennington et al., 2014)
- L'arrivée du Transformer (Vaswani et al., 2017) change la donne en permettant de construire des représentations *contextuelles* (cf. cours suivant)

Word2Vec (Mikolov et al., 2013)



réseaux de neurones à 2 couches

Les vecteurs sont des paramètres



Modèles de langue

- Le type de tâche traitée avec ces architectures (par ex. prédire le mot masqué du milieu) est fortement liée à ce qu'on appelle un **modèle de langue** :

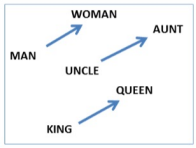
$$p(w_0, w_1, w_2 \dots w_n) = p(w_0) * p(w_1|w_0) * p(w_2|w_0, w_1) * p(w_3|w_0, w_1, w_2) \dots$$

1er mot 2ème mot

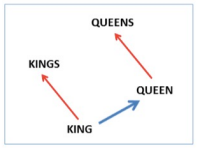
- Pour calculer ces probabilités, il faut réussir à **prédire** un mot à partir de son contexte, comme les mots qui le précèdent

Observations sur l'espace ainsi construit

- Des régularités se retrouvent dans l'espace :



- Cela permet par ex. de résoudre des analogies :



Des résultats surprenants

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Mikolov, T., Yih, W. T., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. Proceedings of HLT-NAACL, pp. 746-751.

Conclusion

- Les plongements de mots sont devenus incontournables en TAL

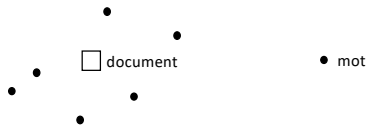
  "The use of word representations… has become a key "secret sauce" for the success of many NLP systems in recent years, across tasks including named entity recognition, part-of-speech tagging, parsing, and semantic role labeling." (Luong et al., 2013)

- Ils répondent aux problèmes des représentations creuses : taille réduite, correspondance approchée, capture de la sémantique
- Ils se couplent naturellement avec les architectures de réseaux de neurones profonds (*deep learning*)

Luong, T., Socher, R., & Manning, C. D. (2013). Better Word Representations with Recursive Neural Networks for Morphology. Proceedings of CoNLL, pp. 104-113.

Et pour les documents ?

- Solution naïve : prendre le centre d'inertie du nuage de points que sont les mots dans l'espace



- Apprendre des représentations contextuelles des mots et des documents (cf. cours suivant)