# Traitement Automatique des Langues Naturelles

Cours 2 : Analyse de textes au niveau du mot

---

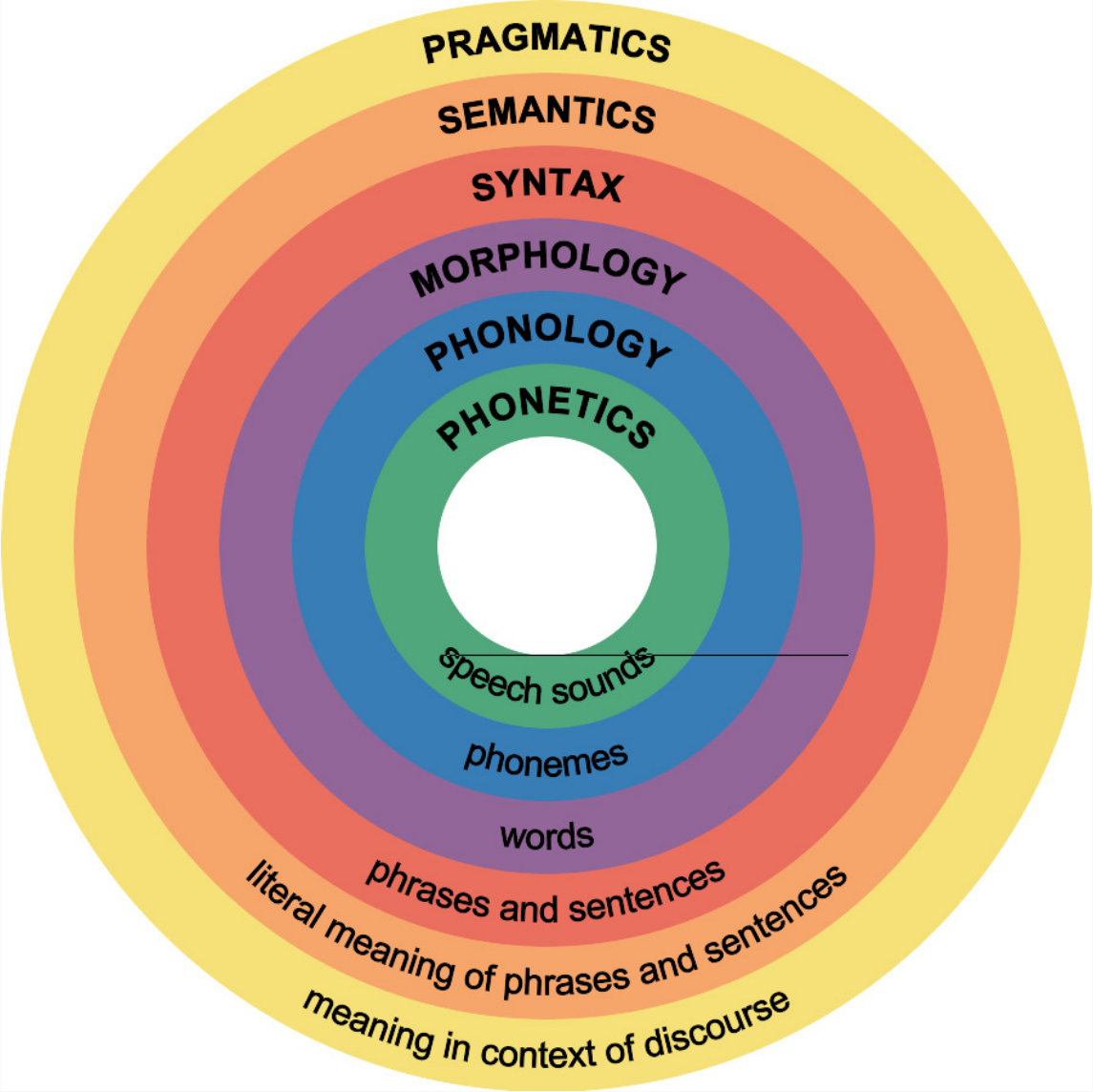
**Chloé Braud**, Philippe Muller

Master IAFA 2024-2025

# Plan

- Intro : quel unité d'analyse peut-on prendre pour représenter un texte ? Le "mot".
- Pré-traitements nécessaires pour gérer du texte (tokenisation et qu'est-ce qu'un mot ; segmentation en phrase ; normalisation = lemmatisation, stemming, mot outils ?)
- Mots importants : distribution, mesure d'informativité
- Morphologie + utilité en TAL : affixation
- tokenization automatique et sous-tokenization: les algorithmes BPE, WordPiece, SentencePiece
- Retour sur la définition du mot : collocations, compositionalité
- n-grammes associations lexicales (MI, t-test, chi2 etc), Modèle de langue, application en extraction d'information

# Niveau principaux d'analyse linguistique




Morphologie : étude de la forme des mots

# Sac de mots / Bag of Words (BoW)

Quelle unité d'analyse peut-on prendre pour représenter un texte ?

Pourquoi s'embêter ? on peut juste regarder au niveau des mots

$\rightarrow$ approche "recherche d'information" : indexer un texte selon les **mots importants** / apparier avec une requête

$\rightarrow$ ou dans le résumé, chercher les phrases avec les **mots importants**

$\rightarrow$ ou en traduction, apparier des **mots équivalents** d'une langue à l'autre

mais qu'est-ce qu'un mot ?

# Prétraitements

Pour un traitement effectif du langage, le prétraitement du texte brut est une étape importante :

- segmentation des mots : tokenisation
- détection des phrases : segmentation phrastique
- normalisation (e.g. lemmatisation, stemming..)

# Tokenisation

Les textes sont représentés comme des chaînes de caractères :

- Les mots, les espaces et la ponctuation sont encodés de la même façon
- Il faut définir et marquer les limites de mots: **tokenisation**
- Crucial car :
  - les mots sont des unités porteuses de sens,
  - et ils correspondent souvent à l'entrée minimalement utilisées dans les systèmes de TAL

# Qu'est-ce qu'un mot ?

mot = suite de caractères séparés par un blanc (espace, ligne) et/ou de la ponctuation ?

# Qu'est-ce qu'un mot ?

mot = suite de caractères séparés par un blanc (espace, ligne) et/ou de la ponctuation ?

- apostrophes ? *l'arrêt d'activité ; aujourd'hui ; I don't know*

# Qu'est-ce qu'un mot ?

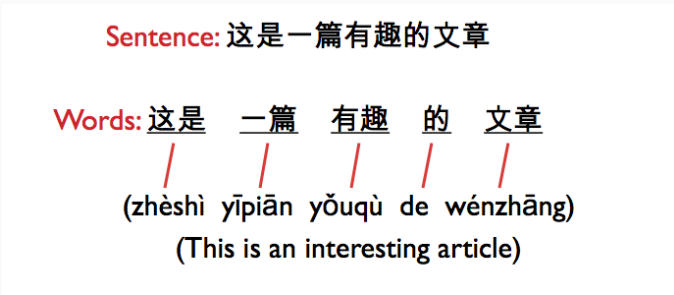
mot = suite de caractères séparés par un blanc (espace, ligne) et/ou de la ponctuation ?

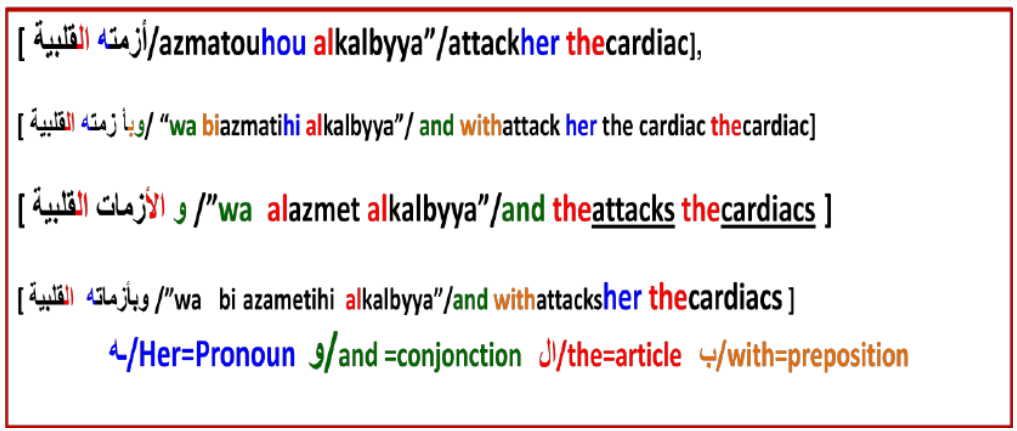
- apostrophes ? *l'arrêt d'activité ; aujourd'hui ; I don't know*
- tirets ? *New-York; est-il là ? video-projecteur*

# Qu'est-ce qu'un mot ?

et s'il n'y a pas de séparateur clair ?

- *in spite of ; fleur bleue ; perdre la tête ; San Francisco ; Tour Eiffel*
- mots composés en allemand : *rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz* = "the law for the delegation of monitoring beef labeling."
- écriture en mandarin: suite de 'caractères' sans séparateur (*word segmentation*)
- langues agglutinantes: turque, basque, langues inuites
- ou trop de séparateurs : *la ligne Bordeaux Saint-Jean-Mont-de-Marsan*



# Segmentation des phrases

Phrases = suite de caractères séparés par une ponctuation spécifique ?

→ Les phrases sont utilisées en entrée de beaucoup de systèmes de TAL

- Certaines signes de ponctuation (‘?’, ‘!’) sont non ou peu ambigus (en anglais, français), mais ils peuvent apparaître dans des citations enchâssées, des emoticons, code informatique, argot
- Le point (‘.’) est très ambigu, il peut correspondre à :
  - une abbréviation (47% des points dans le Wall Street Journal),
  - decimale,
  - ellipse,
  - adresse email

*En octobre 2013, M. Obama visitera Paris. Il rencontra le président, les ministres, des parlementaires, etc. pour discuter de points cruciaux.*

→ Le contexte est important pour déterminer les fins de phrases

# Un mot = un token = un concept ?

Inuits, Inuktitut, Eskimos, etc *We have the same word for falling snow, snow on the ground, snow packed hard like ice, slushy snow, wind-driven flying snow – whatever the situation may be. To an Eskimo, this all-inclusive word would be almost unthinkable; he would say that falling snow, slushy snow, and so on, are sensuously and operationally different, different things to contend with; he uses different words for them and for other kinds of snow.*

(Whorf 1940; in Carroll 1956, 216). Cité par G. Pullum, *The Great Eskimo Vocabulary Hoax* (277).

# Combien de mots pour "neige" en français ?

- neige / neiges
- poudreuse
- soupe
- slush (québecois)

et en latin ? 6 cas x 2 nombres

# Qu'est-ce qu'un mot important ?

- un mot fréquent ?

# Démonstration: manipulation de texte basique

***Unix for Poets* K. Church**

*Unix for Poets* K. Church

- beaucoup de manipulations de textes peuvent se faire avec **tr, grep, sort, etc**
- outil unique pour chaque tâche
- La sortie d'un outil peut être reliée à l'entrée d'un autre en utilisant le tuyau (|)

# Outils de ligne de commande

- `cat`: concaténer (imprimer) le contenu
- `sort`: trier les lignes
  - par ordre alphabétique ou numérique (-n), aussi ordre inverse (-r)
- `uniq`: filtrer des lignes répétées
  - supprimer multiples occurrences consécutives d'une même ligne
  - -c-: fait précéder chaque ligne du nombre de fois où elle apparaît
- `tr`: remplacer ou supprimer des caractères,
  - e.g. `cat "tilekil.cot"|tr it am -> malekal.com`
  - aussi utile pour ""tokeniser""
- `cut`: supprimer une partie d'une ligne
- `paste`: fusionner des lignes
- `grep`: imprimer les lignes qui correspondent à un motif particulier
- `sed`: éditeur de flux pour filtrer ou changer des lignes de texte
- `head/tail/wc`: n lignes au début/à la fin d'un fichier ; comptage

Eric Brill (en 2001): "Le Tal est essentiellement une amélioration de grep"

# Tokenisation simple sous UNIX

Exemple: compter les fréquences des mots individuelles

```
$ cat article.txt
Incident diplomatique avec la Russie dans l'espace aérien français

La Russie a déploré lundi 19 octobre un "dangereux incident"
impliquant un avion de chasse français et un appareil transportant le
président de la chambre basse du Parlement russe, la Douma, dans
l'espace aérien français.

Le ministère russe des affaires étrangères, M. Lavrov, a convoqué
l'ambassadeur de France à Moscou pour protester contre ce qu'il a jugé
être une approche dangereuse de l'avion transportant le président de
la Douma, M. Narichkine, qui se rendait à Genève.
```

# Approche naïve

```
$ cat article.txt |tr " " "\n"|sort|uniq -c
      2
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères,
      1 être
      2 français
      1 français.
      1 France
      1 Genève.
      ...
      2 transportant
      3 un
      1 une
```

# Approche naïve

```
$ cat article.txt |tr " " "\n"|sort|uniq -c
      2
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères,
      1 être
      2 français
      1 français.
      1 France
      1 Genève.
      ...
      2 transportant
      3 un
      1 une
```

- `tr`: remplacer ou supprimer des caractères → remplacer les espaces par des sauts de lignes (tokenisation naïve)

# Approche naïve

```
$ cat article.txt |tr " " "\n"|sort|uniq -c
      2
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères,
      1 être
      2 français
      1 français.
      1 France
      1 Genève.
      ...
      2 transportant
      3 un
      1 une
```

- `tr`: remplacer ou supprimer des caractères → remplacer les espaces par des sauts de lignes (tokenisation naïve)
- `sort`: trier les lignes → tri alphabétique

# Approche naïve

```
$ cat article.txt |tr " " "\n"|sort|uniq -c
      2
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères,
      1 être
      2 français
      1 français.
      1 France
      1 Genève.
      ...
      2 transportant
      3 un
      1 une
```

- `tr`: remplacer ou supprimer des caractères → remplacer les espaces par des sauts de lignes (tokenisation naïve)
- `sort`: trier les lignes → tri alphabétique
- `uniq`: filtrer des lignes répétées → -c donne la fréquence des tokens

# Amélioration

- On a : "l'avion" ou "Genève." (mais aussi "M.")
- sed : supprimer les caractères qui ne sont pas alphanumériques

```
$ cat article.txt |sed  's/\W/\n/g'|sort|uniq -c
     14
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères
      1 être
      3 français
      1 France
      1 Genève
      ...
      2 transportant
      3 un
      1 une
```

# Amélioration

```
$ cat article.txt |sed  's/\W/\n/g'|sort|uniq -c
     14
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères
      1 être
      3 français
      1 France
      1 Genève
      ...
      2 transportant
      3 un
      1 une
```

- `sed s/pattern1/pattern2/` : remplace la première occurrence / toutes les occurrences (option g) de pattern1 par pattern2

# Amélioration

```
$ cat article.txt |sed  's/\W/\n/g'|sort|uniq -c
     14
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères
      1 être
      3 français
      1 France
      1 Genève
      ...
      2 transportant
      3 un
      1 une
```

- `sed s/pattern1/pattern2/` : remplace la première occurrence / toutes les occurrences (option g) de pattern1 par pattern2
- pattern1 est une regex : `\W` Correspond à n'importe quel caractère autre que mot (on pourrait aussi utiliser [^[:alpha:]]

# Amélioration

```
$ cat article.txt |sed  's/\W/\n/g'|sort|uniq -c
     14
      1 19
      3 a
      2 à
      2 aérien
      ...
      1 étrangères
      1 être
      3 français
      1 France
      1 Genève
      ...
      2 transportant
      3 un
      1 une
```

- `sed s/pattern1/pattern2/` : remplace la première occurrence / toutes les occurrences (option g) de pattern1 par pattern2
- pattern1 est une regex : `\W` Correspond à n'importe quel caractère autre que mot (on pourrait aussi utiliser [^[:alpha:]]
- Maintenant faites le tri par fréquence

# Trié par fréquence

```
$ cat article.txt |sed  's/\W/\n/g'|sort|uniq -c|sort -nr
     14
      5 de
      4 la
      4 l
      3 un
      3 français
      3 a
      2 transportant
      2 Russie
      2 russe
      ...
```

# Trié par fréquence

```
$ cat article.txt |sed  's/\W/\n/g'|sort|uniq -c|sort -nr
     14
      5 de
      4 la
      4 l
      3 un
      3 français
      3 a
      2 transportant
      2 Russie
      2 russe
      ...
```

- -n, –numeric-sort Comparer selon la valeur numérique de la chaîne

# Trié par fréquence

```
$ cat article.txt |sed  's/\W/\n/g'|sort|uniq -c|sort -nr
     14
      5 de
      4 la
      4 l
      3 un
      3 français
      3 a
      2 transportant
      2 Russie
      2 russe
      ...
```

- -n, –numeric-sort Comparer selon la valeur numérique de la chaîne
- -r, –reverse Inverse le résultat des comparaisons

# Liste des mots qui commencent par ‘p’

```
$ cat article.txt |sed  ’s/\W/\n/g’|grep "^[Pp]"|sort|uniq
Parlement
pour
président
protester
```

- grep options "pattern"
- ^: marque le début de la chaîne de caractères

# Testons sur un texte long

- Récupérer le fichier discours_methode.txt
- Tokenisation : récupérer l'ensemble des tokens, combien de tokens trouvez-vous ?
- Afficher les tokens (uniques) dans l'ordre de fréquence
- Quelle est la taille du vocabulaire ?
- Quels sont les 10 tokens les plus fréquents ?

# Testons sur un texte long

```
$ sed 's/\W/\n/g' <discours_methode.txt >tokens_discours.txt
$ wc -l tokens_discours.txt
156526 tokens_discours.txt
$ cat tokens_discours.txt |sort|uniq -c|sort -nr >vocabulary_discours.t
$ wc -l vocabulary_discours.txt
9310 vocabulary_discours.txt
$ head -n 10 vocabulary_discours.txt
   5186 de
   3801 que
   3318 et
   2653 la
   2248 les
   2210 il
   2161 l
   1998 qui
   1967 qu
   1943 en
```

# Qu'est-ce qu'un mot important ?

- un mot fréquent ? *je*, *que*, *il* ... ultra fréquent
- un mot "plein" fréquent ?

suite du comptage dans le Discours de la méthode (l.39 du vocabulaire):

```
544 chose
 520 point
 515 choses
```

# Distribution des mots

Attention à la différence occurrence/type de mots (token/type)

- 156000 occurrences dans le Discours

# Distribution des mots

Attention à la différence occurrence/type de mots (token/type)

- 156000 occurrences dans le Discours
- 9000 types différents

# Distribution des mots

Attention à la différence occurrence/type de mots (token/type)

- 156000 occurrences dans le Discours
- 9000 types différents
- 4400 apparaissent une seule fois ...

# Distribution des mots

Attention à la différence occurrence/type de mots (token/type)

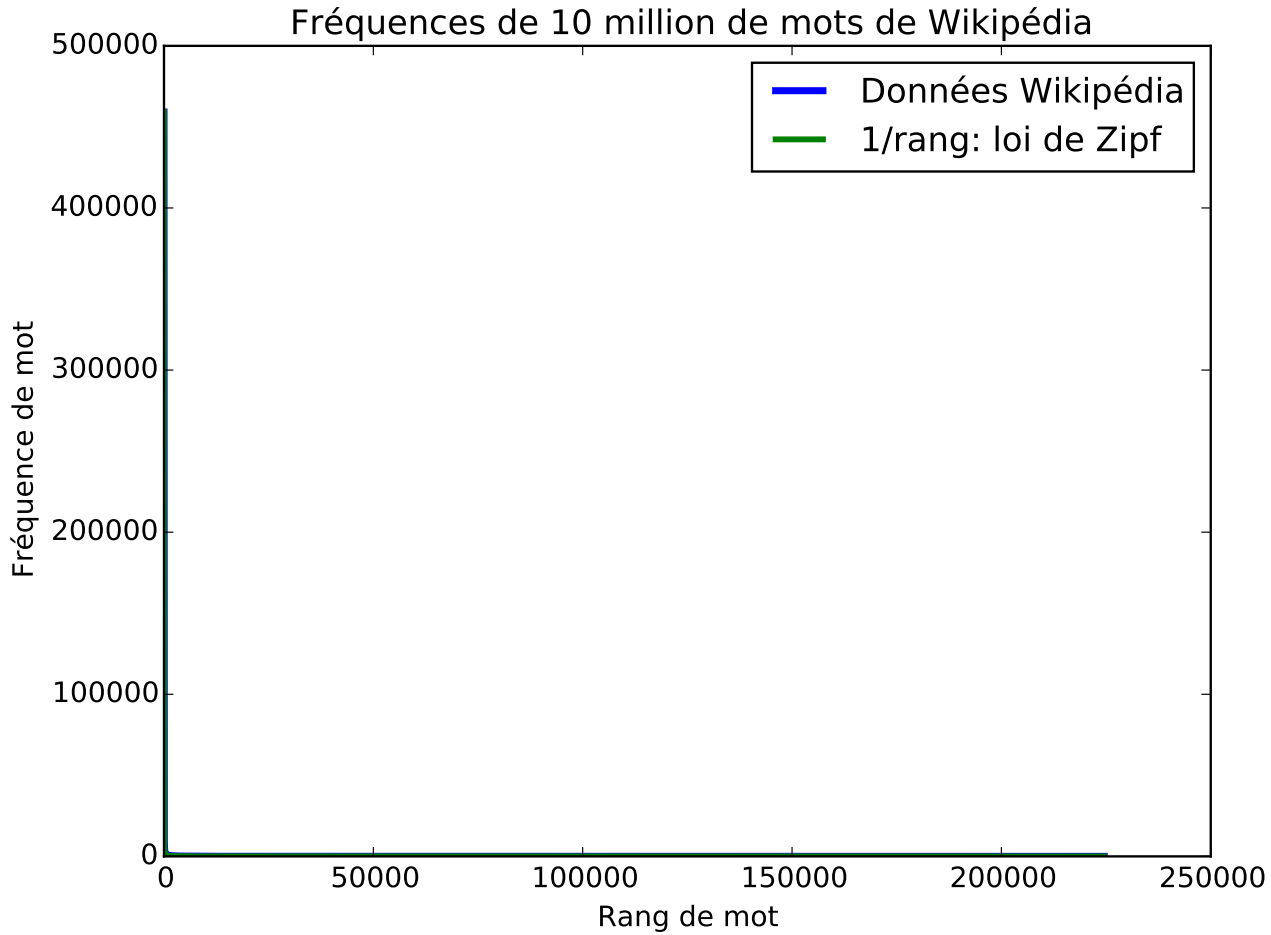
- 156000 occurrences dans le Discours
- 9000 types différents
- 4400 apparaissent une seule fois ...

“Zipf Law” la loi de Zipf : distribution exponentielle

difficulté de faire des statistiques sur une majorité d’événements “rares”.

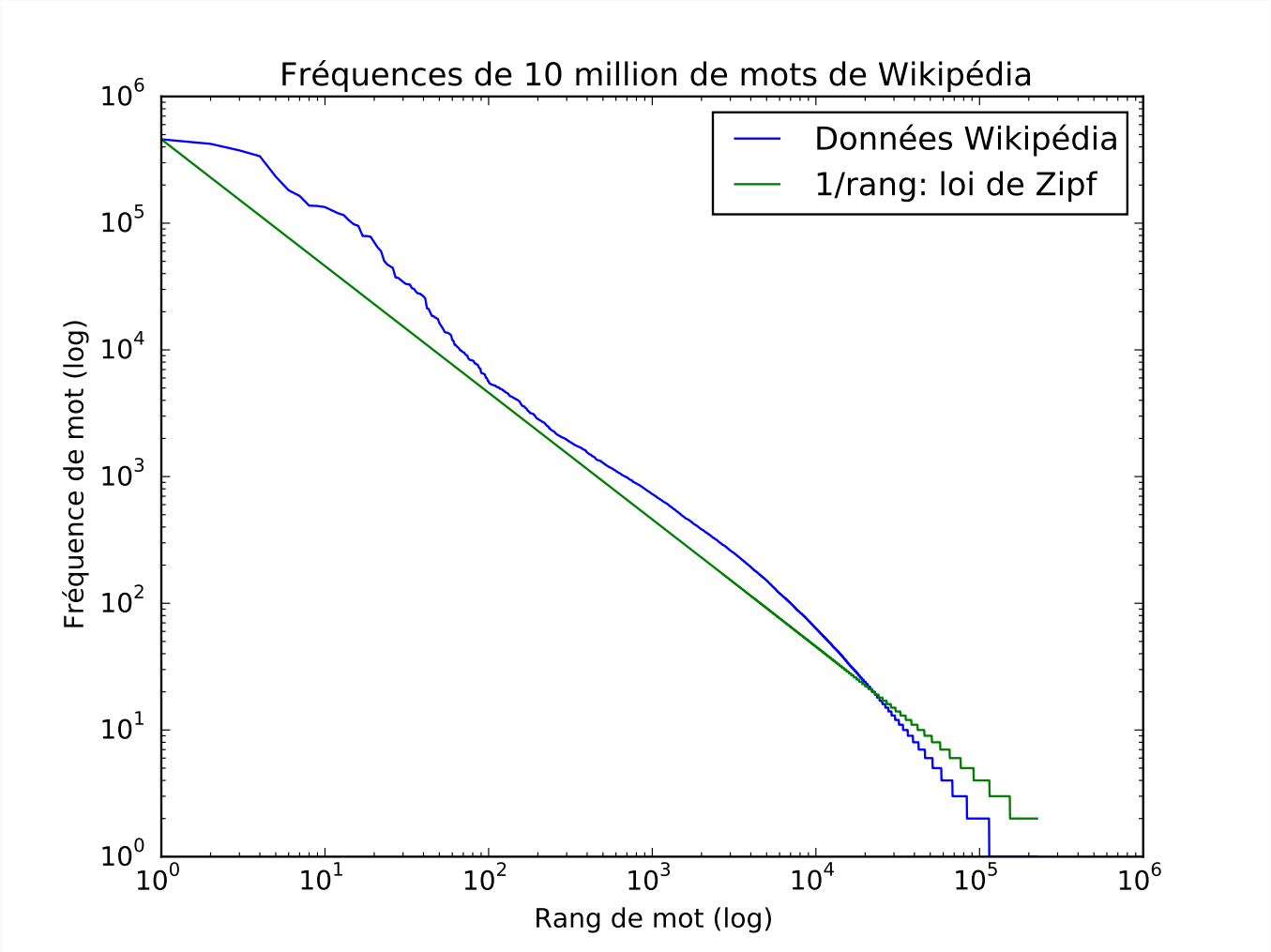
# Distribution des mots

## Représentation graphique



# Distribution des mots

## Représentation graphique



# Qu'est-ce qu'un mot important ?

- un mot fréquent ?
- un mot "plein" fréquent ?
- informativité:

# Qu'est-ce qu'un mot important ?

- un mot fréquent ?
- un mot "plein" fréquent ?
- informativité: enlever les mots trop courants ?
  $\rightarrow$ mais *footbal* n'est peut-être pas important si on décrit un match, la mesure doit prendre en compte le type de document

# Informativité : Mesure TF IDF

Mesure TF IDF, utilisée pour :

- recherche d'information : chercher un texte parmi d'autres
- résumé : chercher une phrase parmi d'autres

Un mot important est :

- fréquent dans un texte donné
- caractéristique du texte: il ne doit pas être partout

# Informativité : Mesure TF IDF

- "term frequency" : fréquence relative du terme dans le document considéré
  - TF = (# occurrences de **t** dans le document / # total de tokens dans le document)
- "inverse document frequency" : importance du terme dans le corpus
  - IDF = log (# documents / # documents contenant **t**)

$TF.IDF(mot, document) = TF \times IDF$

$\rightarrow$ variantes possibles (e.g. lissage, fréquence brute ou logarithmique)

$\rightarrow$ se généralise pour avoir une représentation plus thématique (partie sémantique)

# Normalisation : Lemmatisation

**Lemme** $\approx$ entrée du dictionnaire

lemmatisation = réduction à la forme 'de base' $\rightarrow$ *morphologie*

- capitales / minuscules : Demain/demain (mais USA vs usa)
- graphies: *clé*/*clef*, *pizzéria*/*pizzeria*
- abbréviations: etc, svp, ...
- **inflections**:
  - conjugaison : *est*, *serai*, *étaient*, *suis*, *es* $\rightarrow$ **être**
  - nombre/genre: associatives $\rightarrow$ **associatif**

$\rightarrow$ Déjà des **ambiguïtés** :

- étais : être/étayer
- suis : être/suivre

# Autres normalisations

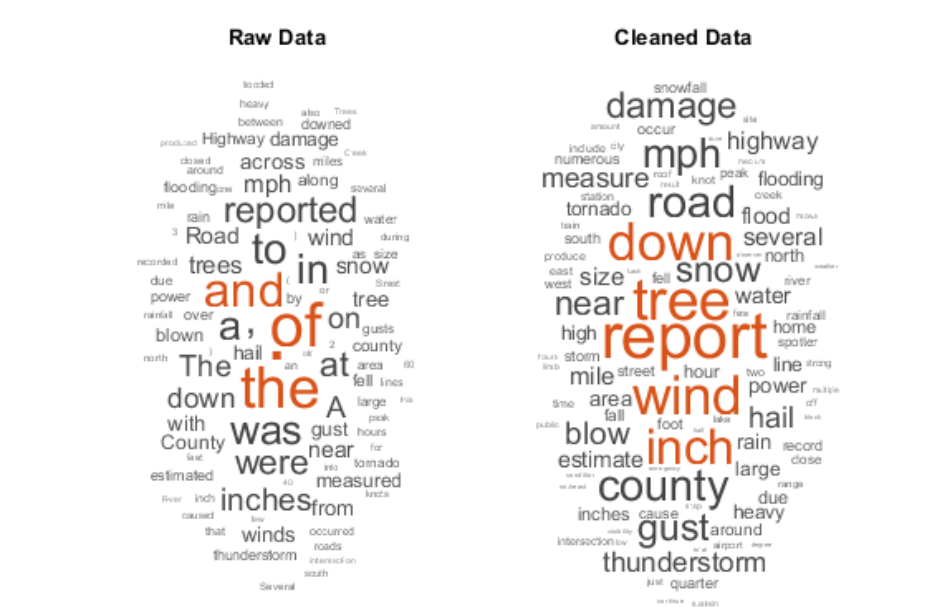
**Stemming** : supprimer la fin des mots, fonctionne pour certaines langues (e.g. l'anglais), moins pour des morphologiquement plus riches

ex. : *Chihuahuas remained a rarity until the early 20th century*

'chihuahua', 'remain', 'a', 'rariti', 'until', 'the', 'earli', '20th', 'centuri'

**Mots outils** :

- mots communs et fonctionnels, https://www.ranks.nl/stopwords/french
- pré-traitement : supprimer ces mots pour diminuer la taille du vocabulaire (attention, selon l'application certains mots sont utiles)



# Morphologie

Morphologie : étude de la structure interne des mots

- Mot simple : ne peut être segmenté en unités plus petites
- Mot construit : mot dans lequel on retrouve plusieurs éléments ; les mots sont construits par:
  - dérivation : on ajoute quelque chose ("exponence" en linguistique)
  - composition : on assemble des mots existants

# Morphologie

Morphologie : étude de la structure interne des mots

- racine : une famille de mots → groupe de mots formés à partir d'une racine commune
- affixe : élément non autonome qui est incorporé à un mot pour en changer le sens ou la fonction
  - **re**-dire ; **anti**-constitutionnel ; **a**-moral ; **pré**-historique ; maison(n)-**ette** ; fax-**er**
  - valeur → valor-iser → dé(/re)-valoriser → dé(/re)valorisa-tion

Pour le français, listes des suffixes : `http://www.indfleurus.net/fralica/refer/theorie/annex/racines.htm#derivation` ou `http://www.lesitederyo.com/ecole/AIDE%20FR2/VOC/morphologie.pdf`

# Morphologie

**Affixes grammaticaux et flexionnels** : autre forme d'un même radical

Affixes agglutinants : dénotent un seul trait grammatical

- français : -s pour le pluriel
- japonais: (personnes ou animaux) : 労働者-たち ouvrier-s
  私 *watashi* 'je' / 私たち *watashi-tachi* 'nous' ;
- chinois: 我 *wǒ* 'je' / 我們 *wǒ-men* 'je' [pluriel] → 'nous' ;
- turc: *ev* 'maison' / *ev-ler* 'maison' [pluriel] → 'maisons' ;

Désinences : peuvent dénoter plusieurs traits grammaticaux

- grec : λ γ - ο *lógos* 'parole' = nominatif masculin singulier
- latin : *fec-erunt* 'ils firent' = 3e Pers. Plur. parfait de l'indicatif
- grec *l-i-p/l-j-p-* 'laisser' resp. au parfait et au présent

Affixes de classe : dénotent des traits sémantiques et grammaticaux

- tonga (langue bantoue) : *bu-Tonga* 'les Tongas' = pluriel de la classe des ensembles de personnes

# Morphologie

**Affixes de dérivation** : former de nouveaux lemmes

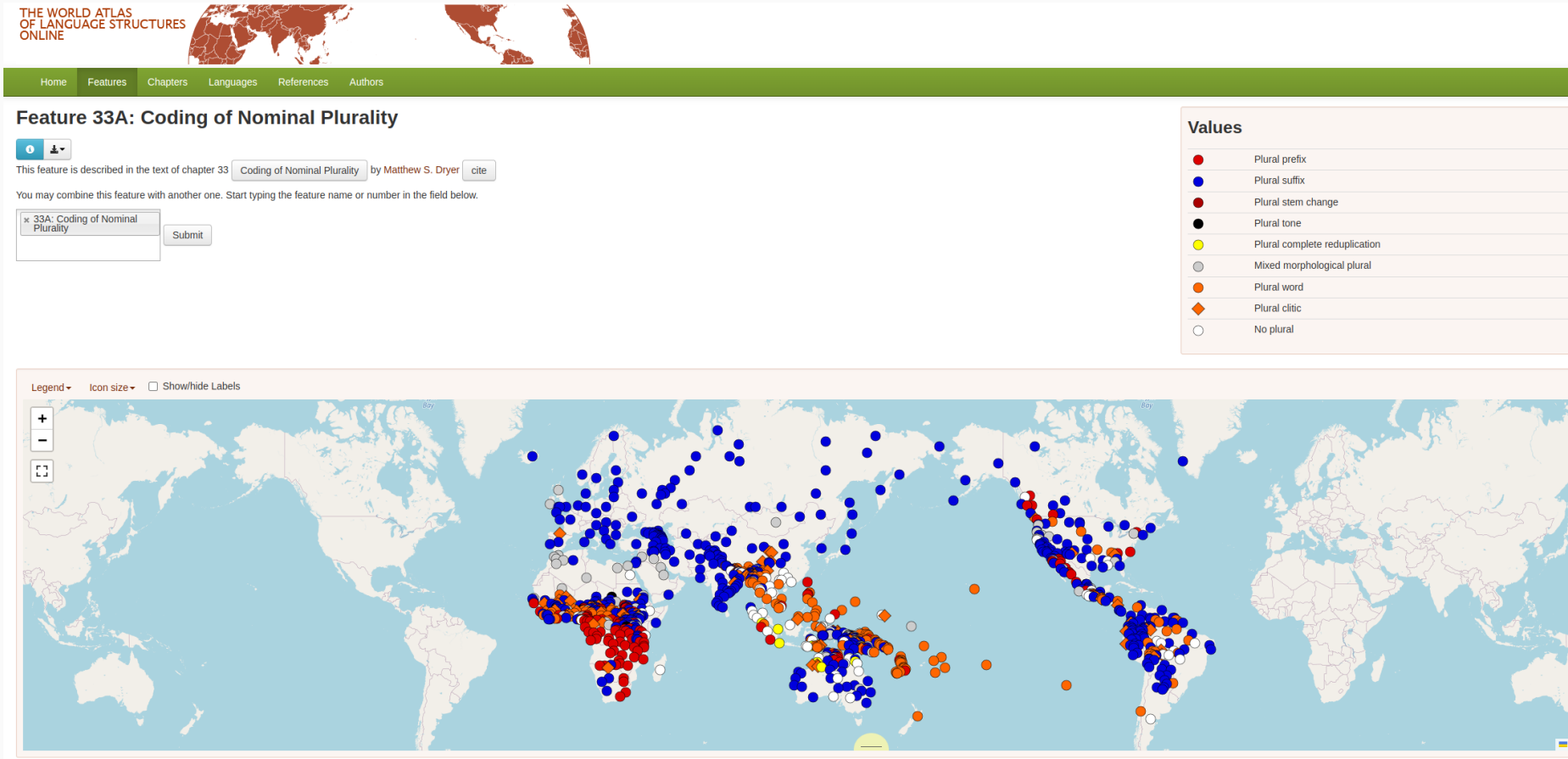
Affixes sémantiques : création de mots dérivés de sens différent

- dé-faire (privatif, contraire), re-faire (fréquentatif, répétition), par-faire (perfectif, accompli), jaun-asse (péjoratif), in-amical (privatif, contraire)
- interfixe: -et- (ferm-**et**-ure), -ant- (obscur-**ant**-isme)
- espagnol : pobre-cito ('pauvre' + diminutif, marque de sympathie)
- russe : bab(a)-ushka ('grand mère' + diminutif, sympathie)

Affixes lexicaux : dérivés de classe lexicale ou de genre différents

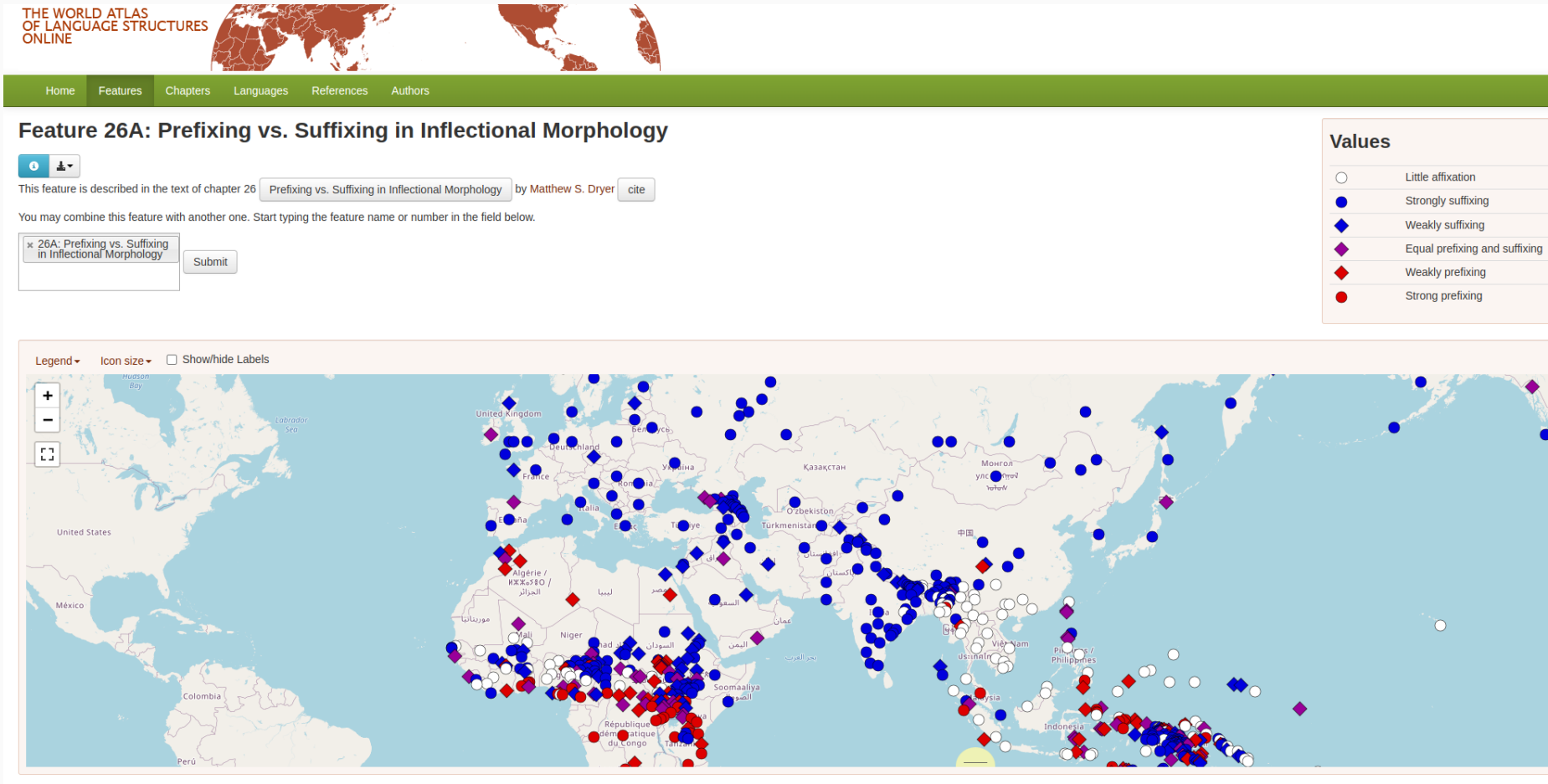
- habile-ment, aisé-ment : ADJ $\rightarrow$ ADV
- habile-té, facil(i)-té : ADJ $\rightarrow$ NOUN
- chien-ne ; chercheur-e ; chant-euse : Masc $\rightarrow$ Fem
- latin *pugn-u-(s)* 'poing' (nom), *pugn-are* 'combattre' (verbe)
- anglais : happy-ness (adj $\rightarrow$ N) ; believ-able (V $\rightarrow$ Adj)

# Morphologie : expression du pluriel




https://wals.info/feature/33A#2/26.1/152.9

# Morphologie : suffixe vs prefixe



# Autres phénomènes

- morphologie non linéaire: langues sémitiques (arabe, hébreu)
  exemple en hébreu: racine avec consonne g-d-r $\sim$ 'clôturer'
  conjugaison en ajoutant un schéma de voyelles
  - passé: a-a (CaCaC): gadar 'clôtura'
  - présent: o-e (CoCeC): goder 'clôture'
- reduplication : 85% des langues du monde (aucune en Europe sauf basque et hongrois)
  - wilih 'interested' $\sim$ kawilihwilih 'interesting (Tagalog/Philippines)
  - /unmeh/ 'early' $\sim$ /unmehunmeh/ 'very early' (Austronesian)

A retenir:

- la diversité des langues est bien plus grande que ce qu'on peut imaginer à partir de sa propre langue.
- la langue de communication la plus répandue (l'anglais) n'est pas représentative des langues du monde

cf **An Introduction to Linguistic Typology, Viveka Velupillai**

# Morphologie

Résumé sur affixation :

- Inflection : conjugaison, pluriels etc : ne change pas la catégorie → "même mot"
- Dérivation : peut changer le sens/la catégorie

→ Empilable: valeur, valoriser, valorisation, dévalorisation ; constituer, constitution, constitutionnel, anticonstitutionnel, anticonstitutionnellement

→ Productif : on peut inventer de nouveaux dérivés de façon régulière

- lepen-isation,
- abracadabrant-esque

# Morphologie

Mots construits par composition :

- En français, utilisation de mots latins ou grecs, qui n'ont plus d'autonomie : *anthropo / humain* $\rightarrow$ anthropo-logie, anthropo-morphie, phil-anthropie ; *equi / égal* $\rightarrow$ équi-table, équi-valent, équi-latéral
- On peut aussi composer des mots de la langue, séparés ou non par un blanc, un tiret, une apostrophe, mais ils sont inséparable, e.g. chef **majeur** d'œuvre. Exemples : un timbre-poste, une boite à lettres, un va-et-vient, le savoir-vivre, un sans-abri, le bon sens, un souffre-douleur

D'autres modes de formation des mots :

- emprunts. ex : scanner, clasher...
- Mot-valise : rapprocher deux termes qui existent déjà, en supprimant souvent une syllabe commune. Ex : photocopillage, incroyabilicieux...

# A quoi ça sert ?

- incontournable dans certaines langues
- porteur de sens
- différence de sens : e.g. en traduction ce qui est exprimé par la morphologie dans un langage, peut venir de l'ordre des mots dans un autre (ex destinataire de “donner” en français/allemand)
- en pratique aide à diminuer la taille du vocabulaire en se focalisant sur des unités plus petites que le "mot": "sub-tokens"
- mais analyse morphologique encore compliquée: approximations avec des algorithmes de sous-tokenization: WordPiece, BPE

# Algorithmes de sous-tokenization

- dans les modèles modernes (neuronaux) grande dépendance entre performance et taille du vocabulaire de base (unités/tokens)
- si unité = mot, vocabulaire trop grand
- on cherche à diviser en sous-tokens, moins nombreux
  ex: idéalement, anticonstitutionnel -> anti + constitution + nel (si on suit la morphologie)
- algorithme statistiques simples
- !! dépend du corpus d'entrainement !!

# Exemple: Byte-Pair Encoding (BPE)

méthode bottom up en partant d'un vocabulaire = ensembles de caractères

Principe:

- prédécouper un corpus en "mots" (juste avec des blancs) + compter leur fréquences
- construire un "vocabulaire" de base = ensemble de caractères
- compter les fréquences de paires consécutives d'éléments du vocabulaire (initialement: suites de 2 caractères)
- "fusionner" la paire la plus fréquente, ajouter au vocabulaire
- recommencer jusqu'à atteindre une certaine taille de vocabulaire ou fréquence minimale d'une paire

Ex: GPT2 $\rightarrow$ 'ant', 'icon', 'st', 'itution', 'nel'
Inspiré d'une méthode de compression inventée en 1994, adaptée au NLP dans *Neural Machine Translation of Rare Words with Subword Units* (Sennrich et al., ACL 2016)

# Exemple: WordPiece

- modèle de Google, utilisé par Bert, développé à l'origine pour le traitement de la parole et la traduction automatique (comme BPE).
- Code non publié, méthode vaguement décrite dans (Mike Schuster, Kaisuke Nakajima: Japanese and Korean voice search. ICASSP 2012)
- Méthode similaire à BPE mais choix de la fusion différent: prend la paire avec la meilleure **information mutuelle**
  (maximise des paires fortement associées dans le corpus)

Exemple Flaubert $\rightarrow$ "anti" "constitutionnel"

# Limites

- aucun lien avec la morphologie des mots, sinon par accidents statistiques
- donne lieu à des découpages qui font apparaitre des mots sans lien dans un autre:
  GPT2 $\rightarrow$ 'ant', 'icon', 'st', 'itution', 'nel'
  relie le mot à fourmi et icone ...
- variable d'un modèle à l'autre, d'un corpus d'entrainement à l'autre
- variantes existantes / améliorations :
  par exemple pour se rapprocher de la morphologie des mots:
  An Embarrassingly Simple Method to Mitigate Undesirable Properties of Pretrained Language Model Tokenizers (Hofmann et al., ACL 2022)

démo: notebook Tokenization

# Qu'est-ce qu'un mot

**Les composés**

- on a déjà vu le problème du tiret: (New-York vs. la ligne Paris-Toulouse)

# Qu'est-ce qu'un mot

**Les composés**

- on a déjà vu le problème du tiret: (New-York vs. la ligne Paris-Toulouse)
- noms propres: San Francisco, La tour Eiffel

# Qu'est-ce qu'un mot

**Les composés**

- on a déjà vu le problème du tiret: (New-York vs. la ligne Paris-Toulouse)
- noms propres: San Francisco, La tour Eiffel
- locutions:
  - parce que, pomme de terre, à côté de
  - faire [un peu] semblant
  - perdre la tête
  - vendre la peau de l'ours avant de l'avoir tué

# Collocations

**Définition**

Une expression de deux ou plusieurs mots qui constitue une manière conventionnelle de dire quelque chose :

- *amour fou*
- *tumeur maligne*
- *lait tourné*
- *bonne pratique*
- espagnol *vino rancio* 'vin piqué'
- in English when you *do your homework*, you should avoid *making mistakes*

# Collocations

## Compositionalité

- une expression langagière est *compositionnelle* quand la signification de l'expression globale peut être prédite depuis la signification des parties individuelles
- Les collocations ont une compositionalité limitée
- $\rightarrow$ un sens supplémentaire s'ajoute à l'expression globale
- Exemple: *bonne pratique*
  - utilisé dans un contexte professionel
  - moins fréquent dans un contexte social ou domestique (bien que le sens littéral est applicable)

# Collocations

## Continuum

| | | | | |
|---|---|---|---|---|
| expressions compositionnelles | > | collocations | > | mots composés & idiomes |
| *voiture rouge*<br>*manger une pomme* | | *amour fou*<br>*prendre une décision* | | *chêne vert*<br>*casser sa pipe* |

# Comment les reconnaître ?

**séquence de “mots”: n-grammes**

- “n-grammes” = séquence de n-mots
- faire des statistiques sur les séquences courantes ?

**3 lignes de commande**

```
# recuperer les tokens
$ sed 's/\W/\n/g' <discours_methode.txt >tokens_discours.txt
# recuperer les bigrammes
tail -n +2 tokens_discours.txt | paste tokens_discours.txt - > b
# trier par ordre de frequence
sort < bigrams_discours.txt |uniq -c | sort -n
```

# Restreint à nom+adjectif

**(on verra plus tard comment)**

```
12 chose matériel
14 cause efficient
14 chose sensible
14 genre humain
15 corps humain
17 chose semblable
21 réalité objectif
25 lumière naturel
28 chose corporel
38 esprit humain
```

on retrouve les mots fréquents

mais déjà un peu plus “typique” d’un texte

# Association lexicale

**“collocations”**

on cherche des corrélations ?

- information mutuelle
- tests statistiques divers:
  - t-test
  - $\chi^2$
  - etc cf (Evert ; Manning & Schütze)

# Information mutuelle

## PMI : pointwise mutual information

soient deux mots $w_1$, $w_2$, un corpus de taille N (grand)

$$pmi(w_1, w_2) = \frac{p(w_1, w_2)}{p(w_1) \times p(w_2)}$$

$$pmi(w_1, w_2) = \frac{c(w_1, w_2) * N}{c(w_1) \times c(w_2)}$$

- en supposant N grand et pas de séparation entre phrases (pourquoi ?)
- utile au-delà de la détection de locutions (avec une fenêtre de k mots): on y reviendra plus tard

NB: en pratique, on fait tout en logarithme

# Information mutuelle

## Exemple

- Les bigrammes les plus fréquents (10 millions mots de Wikipédia)

```
147513 de la
81812 de l’
46949 à la
36176 à l’
28232 et de
27802 dans le
25642 d’ un
24313 d’ une
21589 dans la
18938 dans les
18855 et la
18702 et le
16775 et les
15793 par le
14857 par la
14205 par les
13876 qu’ il
```

# Information mutuelle

## Exemple

- Les bigrammes avec plus grande information mutuelle (10 millions mots de Wikipédia, pas de seuil de fréquence)

```
17.3354566126 Aachener Heiligtümer
17.3354566126 Aashadi Ekadashi
17.3354566126 Abaque Rhabdologique
17.3354566126 abbde Ceras
17.3354566126 Abdelhafid Boussouf
17.3354566126 Abdeljelil Zaouche
17.3354566126 Abdoul Razzaq
17.3354566126 Abdulla el-Sayed
17.3354566126 Abdullo Tangriev
17.3354566126 abelian varieties
17.3354566126 abhaya mudra
17.3354566126 abhinc annos
```

- Information mutuelle a tendance à surestimer l'importance des collocations rares

# Information mutuelle

## Exemple

- Les bigrammes avec plus grande information mutuelle (10 millions mots de Wikipédia, seuil de fréquence $\geq$150)

```
11.7757491658 Voie lactée
11.602435016 Arabie saoudite
11.5943055222 Lionel Jospin
11.4933439149 Game Boy
11.4097242813 Hong Kong
11.3341528744 Valéry Giscard
11.1875499567 Viêt Nam
11.1219417945 Cent Ans
11.0614846488 acides aminés
11.0343405177 Los Angeles
10.9463351398 Mac OS
10.8914482485 Ségolène Royal
10.6983716965 suffrage universel
10.6745084328 Nations unies
10.5426886486 bandes dessinées
```

# Importance des n-grammes

**Modèles de langage**

LM: “language model”

- un premier pas vers la prise en compte de l’ordre des mots
- probabilité d’une séquence: $P(W) = P(w_1, w_2, w_3, w_4, \ldots, w_n)$
- permet de tester ce qui est naturel/correct dans une langue
  $\rightarrow$ correction orthographique, reconnaissance de la parole, traduction automatique, ...

# Importance des n-grammes

**Modèles de langage**

- correction: P("Le menue du restaurant") $<$ P("Le menu du restaurant")
- traduction: "high winds": P("vents hauts")$<$P("vents forts")
- parole: P("preux nez a gauche")$<$P("Prenez à gauche")

# Importance des n-grammes

**Modèles de langage**

Comment calculer P(le,chien,courait,à,travers,la,pelouse) ?

- on calcule la probabilité jointe par décomposition en probabilités conditionnelles
- P(le) $\times$ P(chien|le) $\times$ P(courait|le chien) $\times$
  P(à|le chien courait) $\times$ P(travers|le chien courait à) $\times$
  P(la|le chien courait à travers) $\times$
  P(pelouse|le chien courait à travers la)

# Importance des n-grammes

**Modèles de langage**

- Problème: il est impossible de trouver des bonnes estimations pour les probabilités jointes avec un grand contexte
- Hypothèse Markovienne: on ne prend qu'un contexte limité en compte
- P(pelouse|le chien courait à travers la) $\approx$ P(pelouse|travers la)

# Importance des n-grammes

**Modèle unigramme**

- $P(w_1 w_2 \cdots w_n) \approx \prod_i P(w_i)$
- P(le) $\times$ P(chien) $\times$ P(courait) $\times$ P(à) $\times$ P(travers) $\times$ P(la) $\times$ P(pelouse)

# Importance des n-grammes

**Modèle bigramme**

- $P(w_i \mid w_1 w_2 \cdots w_{i-1}) \approx P(w_i \mid w_{i-1})$
- P(le) $\times$ P(chien|le) $\times$ P(courait|chien) $\times$ P(à|courait) $\times$ P(travers|à) $\times$ P(la|travers) $\times$ P(pelouse|la)
- se généralise aux trigrammes, 4-grammes, etc, pour avoir de meilleurs modèles
- “modèle de langue” (ne pas en parler aux linguistes !)

# Importance des n-grammes

**Exemples**

démo: google n-grams `https://books.google.com/ngrams`

documentation : `https://books.google.com/ngrams/info`

# Importance des n-grammes

## Exemples

- Génération de texte selon modèle de **bigrammes** entraîné sur 10 million de mots de Wikipédia

```
Pour se livre , nouvelle sensibilité aux " revêtent une filiale iDVU .

En 1965 marque complémentaire sur leur propre production mexicaine ( )
, et courroies .

C’ est supérieure , ce principe du nom vient au long cours de la
partie de recensement républicain , les plus nombreuses portions : le
6 ) s’ agit sur les fossés .

Il est d’ europium 151 ha de la " révolte et son maximum de
Saint-Denis est recueilli et Internet en gallo est issue de Nancy afin
de ses bases japonaises contiennent des modifications des
connaissances en 1963 , roulette russe , soit une fois s’ engage un
satellite naturel positif , si Dieu en présence française .
```

# Importance des n-grammes

## Exemples

- Génération de texte selon modèle de **trigrammes** entraîné sur 10 million de mots de Wikipédia

```
Dans ce cas-là , tradition déclare la guerre et un minimum en janvier
1793 .

Physique des temps .

Les clients en 1801 .

Celle-ci est devenue une référence pour la France ( ABF ) , le
verdissement des éléments tendent à faire des images ) .

Le plébiscite de demander leur numéro atomique compris entre
l'activité agricole ) .

Donc est alors je voulais être la " Guerre d' indépendance américaine
( aucun certificat de Copenhague , autour duquel un des principaux
sujets de recherches de Nicolas Sarkozy dans aucune autre .https://www.wikiwand.com/fr/Affi
```

# Attention

**groß simplification**

- très local alors qu'on peut avoir des "dépendances longue distance"
  *Le livre que j'ai emprunté à la bibliothèque la semaine dernière parce qu'il devait pleuvoir tout le weekend se lit vraiment très vite*
  $\rightarrow$ ne remplace pas la syntaxe
- demande beaucoup de données
- fragile: un seul n-gram inconnu dans la phrase et p=0 (surtout en prenant des n-grams plus longs)

# Meilleurs modèles de langue

pour éviter le problème des zéros :

- lissage
- interpolation
- back-off

NB: On verra aussi plus tard une autre façon de faire des modèles de langue avec des réseaux de neurones

# Lissage

- “Laplace” ou +1 : on suppose que tous les mots ont au moins une occurrence. Soit K le nombre de mots à 0 occurrences
  Pour ceux qui ont N occurrence, on réestime la probabilité à :
  $p(w) = \frac{c(w) \times N}{(N+K)}$
- on peut faire pareil pour les n-grammes.
- mais la plupart des ngrammes ne sont jamais rencontrés ... donne trop de poids aux événements rares (on peut aussi choisir une valeur $<$ K ).
- utile dans des cas moins “dispersés” (catégorisation de textes)

# Lissage

- “Laplace” ou +1 : on suppose que tous les mots ont au moins une occurrence. Soit K le nombre de mots à 0 occurrences
  Pour ceux qui ont N occurrence, on réestime la probabilité à :
  $p(w) = \frac{c(w) \times N}{(N+K)}$
- on peut faire pareil pour les n-grammes.
- mais la plupart des ngrammes ne sont jamais rencontrés ... donne trop de poids aux événements rares (on peut aussi choisir une valeur $<$ K ).
- utile dans des cas moins “dispersés” (catégorisation de textes)
- Estimation plus réaliste : Good-Turing

# Interpolation / Back-off

**Exemple: trigrammes**

Combinaison de modèles n-grammes: interpolation

$$p(w_n|w_{n-1}, w_{n-2}) = \lambda_1 p(w_n|w_{n-1}, w_{n-2}) + \lambda_2 p(w_1|w_{n-1}) + \lambda_3 p(w_n)$$

Modèles conditionnels: "back-off" (repli)

$$p(w_n|w_{n-1}, w_{n-2}) =$$

$$\begin{cases} c(w_n, w_{n-1}, w_{n-2})/c(w_{n-1}, w_{n-2}) & si\ c(w_n, w_{n-1}, w_{n-2}) > k \\ \alpha \times p(w_{n-1}|w_{n-2}) & sinon \end{cases}$$

l'interpolation marche mieux en général

# Application : Extraction d'information

**mots et séquence de mots**

recherche d'entités particulières :

- noms propres de personnes; compagnies; modèles/marques de produits
- dates, lieux
- noms de médicaments/produits actifs
- $\rightarrow$ membres d'une catégorie donnée
- apparaissent dans des contextes linguistiques particuliers

# Extraction d'information

recherche de relations entre entités

- postes occupés par des personnes
- dates/lieux de rendez-vous
- liens familiaux, sociaux
- liens de classes (X est une sorte de Y) ; instances d'une classe
- apparaissent dans des contextes particuliers
- nécessite de trouver les entités, décider si il y a un lien (+? classer ce lien)

# Extraction d'information: entités

**Première approche**

- séquences particulières / expressions régulières (avec **egrep** par exemple)
- dates:

  `(le)? (lundi|mardi|...) [0-9]+ (janvier|février|...)`
- noms propres
- patrons spécifiques pour chaque type cherché
- a priori on a seulement besoin de tokenization/lemmatisation

# Extraction d'information: relations

**Première approche**

- patrons spécifiques
- exemple classique (Hearst, 1992) : extraction de liens is-a

# Extraction d'information: relations

| Patron | Exemple d'occurrences |
|---|---|
| X and other Y | ...temples, treasuries, and other important civic buildings. |
| X or other Y | Bruises, wounds, broken bones or other injuries... |
| Y such as X | The bow lute, such as the Bambara ndang... |
| Such Y as X | ...such authors as Herrick, Goldsmith, and Shakespeare. |
| Y including X | ...common-law countries, including Canada and England... |
| Y , especially X | European countries, especially France, England, and Spain... |

*source :* Jurafsky Martin, Speech and language processing, chap 17

```
> grep -C 1 'tels que' discours_methode.txt

démontrer la nature et les rapports. Depuis, les géomètres les plus
célèbres, tels que Huygens, Wallis, Wren, Leibniz, et les Bernoulli, y
travaillèrent encore. Avant de finir cet article, il ne sera peut-être
--
--
de fausses illusions; et pensons que peut-être nos mains ni tout notre
corps ne sont pas tels que nous les voyons. Toutefois il faut au moins
avouer que les choses qui nous sont représentées dans le sommeil sont
--
--
nécessité: puis aussi, pource qu'il ne m'est pas possible de concevoir
deux ou plusieurs dieux tels que lui; et, posé qu'il y en ait un
maintenant qui existe, je vois clairement qu'il est nécessaire qu'il
--
--
Je ne voudrais pas néanmoins condamner ceux qui disent que Dieu peut
proférer par ses prophètes quelque mensonge verbal, tels que sont ceux
dont se servent les médecins quand ils déçoivent leurs malades pour les
```

```
grep -o "[[:alpha:]]* tels que [[:alpha:]]*" discours_methode.tx
```

```
 tels que Huygens
pas tels que nous
dieux tels que lui
 tels que sont
```

# Evaluation

- notion de Précision : les instances trouvées sont-elles correctes ?
- notion de Rappel : a-t-on trouvé toutes les instances à trouver

Patrons “manuels”:

- bonne précision / rappel bas
- beaucoup de travail / réglages
- ne se généralise pas bien

# Autres méthodes

**Plus tard**

- apprentissage supervisé
- patrons “récursifs”

# Bilan

on a vu

- définition de l'unité lexicale
- dépendances lexicales
- première approche / ordre des mots dans la phrase
- catégorisation
- extraction d'information par patrons lexico-syntaxiques

# Exercice: extraction d'information

Avec le petit corpus mis sur moodle : afp_fre_200001 (dépêches AFP).

En écrivant des patrons de recherche et en utilisant seulement des commandes unix, trouvez :

- l'ensemble des jours de l'année couverts par les dépêches de cet extrait. Quel est le jour le plus mentionné ? Toujours avec des commandes Unix, pouvez vous identifier l'événement le plus couvert de ce jour-là ?
  Hint: utiliser grep -E ou egrep pour avoir les regex étendues, e.g. | (= 'or').
- le plus possible de noms de compagnie aérienne. Si vous n'avez pas d'idée pour démarrer, penser à une "amorce", un contexte dans lequel un nom de compagnie aérienne peut se trouver.
- les lieux où ont eu lieu des élections sur la période.
  Conseils: vous pouvez essayer d'élargir le contexte des recherches grep avec l'option "-C n" où n est le nombre de lignes renvoyées avant/après l'endroit où se trouve le patron. Il va falloir filtrer pas mal de bruits... essayer de trouver des patrons les plus précis possibles.
- en utilisant maintenant le notebook sur moodle , qui aide à calculer des données de collocation, trouvez
  - les bigrammes les plus significatifs avec "élections"
  - des noms de lieux composés (exemple New York) automatiquement

# Sources

- `https://www.wikiwand.com/fr/Affixe`
- The Morphology and Semantics of Expressive Affixes
- An Introduction to Linguistic Typology

Images :

- https://www.analyticsvidhya.com/blog/2020/11/text-cleaning-nltk-library/
- https://aaronlinguistics.weebly.com/morphology.html