

# Humanités Numériques

## Cours 8 : Sonder un grand corpus

Loïc Grobol

2021-11-09

### Google Books

- Collection multilingue de livres de 1500 à 2008
- Numérisation de plus de 5 millions de livres grâce à la reconnaissance optique de caractères (OCR)
- 500 milliards de mots cf. J. Michel et al. (2011). “Quantitative Analysis of Culture Using Millions of Digitized Books”, Science, vol. 331
- Nombre de mots par langues :
  - anglais : 361 milliards
  - français : 45 milliards
  - espagnol : 45 milliards
  - allemand : 37 milliards

### Mots ?

Extraction des « mots » en fonction :

- des caractères blancs
- de la ponctuation, des tirets et des apostrophes

Beaucoup de bruit :

- erreurs de reconnaissance automatique
- typos
- mots qui contiennent des chiffres, abréviations, etc.
- utilisation d’autres langues dans un livre majoritairement en anglais

### Un mot ou un “gram” ?

**token** : chaîne de caractères délimitée par des espaces

- 1-gramme (ou unigramme) : 1 token
- 2-gramme (ou bigramme) : une suite de 2 tokens
- 3-gramme (ou trigramme) : une suite de 3 tokens
- n-gramme : une suite de  $n$  tokens

---

Exemple :

Quand le mystère est trop impressionnant, on n'ose pas désobéir

- **Unigrammes** : Quand / le / mystère / est / trop / impressionnant / on / n'ose / pas / désobéir
  - **Bigrammes** : Quand le / le mystère / mystère est / est trop / trop impressionnant / impressionnant on / on n'ose / n'ose pas / pas désobéir
  - **Trigrammes** : Quand le mystère / le mystère est / mystère est trop / est trop impressionnant / trop impressionnant on / impressionnant on n'ose / on n'ose pas / n'ose pas désobéir
- 

Exercice : compter les bigrammes dans la phrase suivante :

Longtemps je me suis couché de bonne heure

---

1. Longtemps je
2. je me
3. me suis
4. suis couché
5. couché de
6. de bonne
7. bonne heure

## *Ngram Viewer*

Permet de faire des requêtes dans Google Books :

- n-grammes de mots
- n-grammes de catégories grammaticales (parties du discours)

Exemple : « vélo » vs « vélo » vs « bicyclette »

→ L'évolution des fréquences relatives des n-grammes dans le corpus

## Questions d'usage

Dit-on « autant pour moi » ou « au temps pour moi » ?

---

« par contre » ou « en revanche »

---

« *the United States are* » ou « *the United States is* »

- Anglais Américain
- Anglais Britannique

## Évolution des nominations

Ouvriers, travailleurs ou salariés

Et au pluriel

---

Féminisation des noms de métiers

et des fonctions

## Un proxy pour les LSHS

Le communisme

## Des trucs mécaniques

Les années absolues

et relatives

## Opérations

### Multiplier et diviser par un nombre

Utiliser les opérateurs \* et / :

ouvrier, travailleur, (prolétaire\*10)

(ouvrier/10), (travailleur/10), prolétaire

Parenthèses obligatoires !

### Additionner et soustraire

Sans surprise avec + et - :

(United States is-United States are)

‘(Bigfoot+Sasquatch), Yeti

## Catégories grammaticales

Avec `{CATEGORIE}`, par exemple `salarié_NOUN`

ouvrier, `salarié_NOUN`, `travailleur_NOUN`, `employé_NOUN`

---

Code	Catégorie
<i>NOUN</i>	nom
<i>VERB</i>	verbe
<i>ADJ</i>	adjectif
<i>ADV</i>	adverbe
<i>PRON</i>	pronom
<i>DET</i>	déterminant
<i>ADP</i>	adposition
<i>NUM</i>	chiffre
<i>CONJ</i>	conjonction
<i>PRT</i>	particule

Pour les linguistes : c'est un sous-ensemble des *Universal Part of Speech*.

## Le joker

Un autre usage de `*` : remplacer n'importe quel mot.

Exemple : `une colère *` donne les fréquences de trigrammes dont les deux premiers mots sont « une » et « colère »

## Les dépendances syntaxiques

`y => x` permet de savoir à quelle fréquence un mot `x` (« tigré », « roux » ou « noir ») modifie un mot `y` (« chat »)

`chat=>noir,chat=>roux,chat=>tigré`

Le concept de dépendance syntaxique est en fait plus vaste que ça : regardez par exemple `parle=>je,je parle,je * parle`.

## Autres fonctionnalités

Voir la doc

- `_INF` : formes flechées d'un verbe »> ex : `manger_INF`
- `*` avant une catégorie grammaticale : remplace n'importe quel mot de cette catégorie
- `*_ADJ` renvoie n'importe quel adjectif (les 10 plus fréquents sont renvoyés)
- `_START_` : début de la phrase
- `_END_` : fin de la phrase
- `_ROOT_` : racine de l'arbre de dépendance de la phrase

## À vous de jouer !

Faire la fiche d'exercices

## Remerciements

Pour les versions précédentes de ce cours que nous avons construit au fil des années, merci à Isabelle Tellier, Kim Gerdes, Serge Fleury, Yoann Dupont, Pablo Ruiz Fabo, Marine Delaborde et Mathilde Regnault.