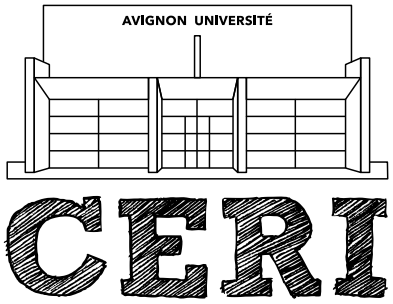


Centre d'Enseignement et de Recherche en Informatique

Partie 6
Compression d'index

Vincent Labatut

Laboratoire Informatique d'Avignon – LIA EA 4128

vincent.labatut@univ-avignon.fr

2019/20

M2 ILSEN

UE Ingénierie du document et de l'information

UCE3 Indexation & Recherche d'information

AVIGNON UNIVERSITÉ

Plan de la séance

1. Généralités et observations
2. Tailles du lexique vs. corpus
3. Fréquence des termes
4. Traitement du lexique
5. Traitement des postings

Section 1

Généralités et observations

AVIGNON
UNIVERSITÉ

Compression d'index
Intérêt de la compression

- Intérêt :
  - Volume de stockage réduit (typiquement : de 75%)
  - Accès plus rapide (plus d'informations mises en cache)
  - Transfert plus rapide
    - Temps transfert non-compressé > temps transfert compressé + temps décompression...
    - ...à condition d'utiliser des algos de compression rapides

Compression d'index
Nature de la compression

- Données compressibles :
  - Lexique
    - Compressé → pourrait tenir en mémoire
    - Traitement de requête plus rapide
  - Listes de postings
    - Réduire l'espace disque occupé
    - Réduire le temps d'accès aux listes de postings
    - Garder certains postings en mémoire (cache)
- Deux types de compressions
  - Avec perte : taux de compression plus élevé
    - Prétraitement effectué lors de l'indexation ≈ compression avec perte
  - Sans perte : méthodes présentées dans la suite
- Méthode de compression dépend des caractéristiques des données

Compression d'index
Description d'une collection

- Collection Reuters RCV1 :

	Termes (taille du lexique)			Postings (taille de l'index non-positionnel)			Tokens (taille de l'index positionnel)		
	Nbre (k)	**Δ%**	**T%**	**Nbre (k)**	**Δ%**	**T%**	**Nbre (k)**	**Δ%**	**T%**
Tout	484	–	–	109 971	–	–	197 879	–	–
Sans nombres	473	−2	−2	100 680	−8	−8	179 158	−9	−9
Minuscules	391	−17	−19	96 969	−3	−12	179 158	0	−9
30 mots-vides	391	0	−19	83 390	−14	−24	121 858	−31	−38
150 mots-vides	391	0	−19	67 002	−30	−39	94 517	−47	−52
Racinisation	322	−17	−33	63 812	−4	−42	94 517	0	−52

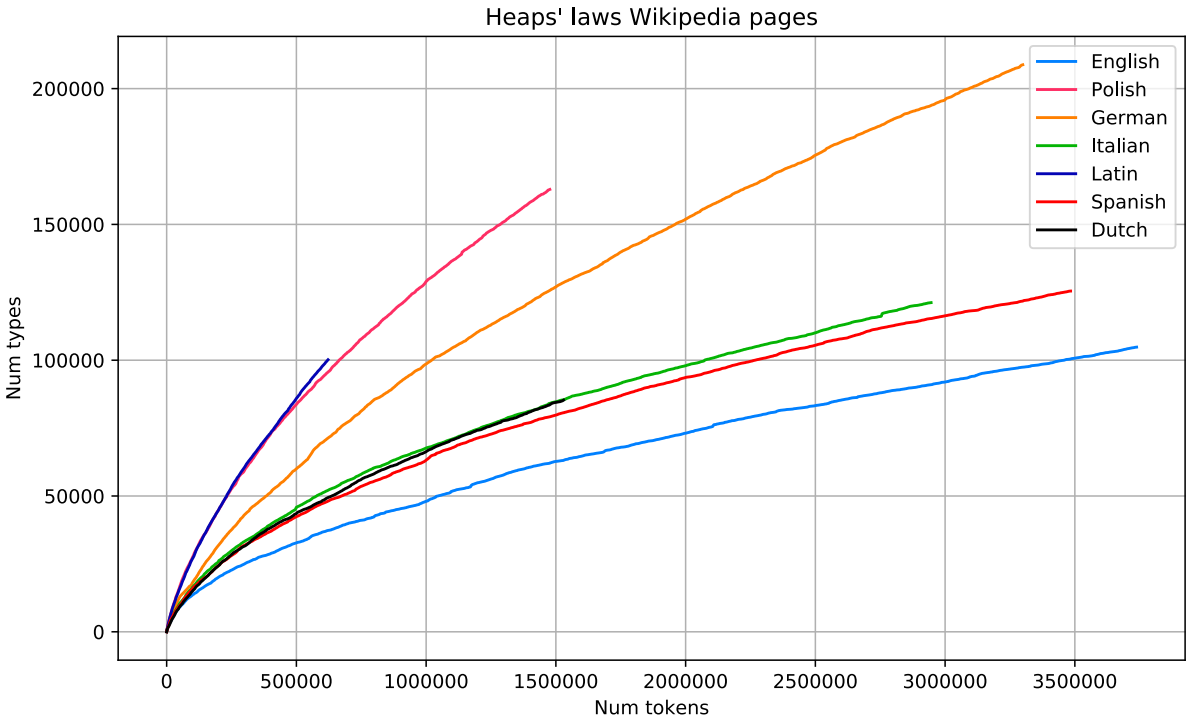
- Observations :
  - Nombre de *termes* affecté par casse et racinisation
  - Nombre de *postings* affecté par mots-vides
    - Règle des 30 : 30 mots les plus fréquents ≈ 30% des tokens
  - Réduction causée par 150 mots-vides pas maintenue après compression
  - En FR : gain encore plus important avec racinisation

Section 2

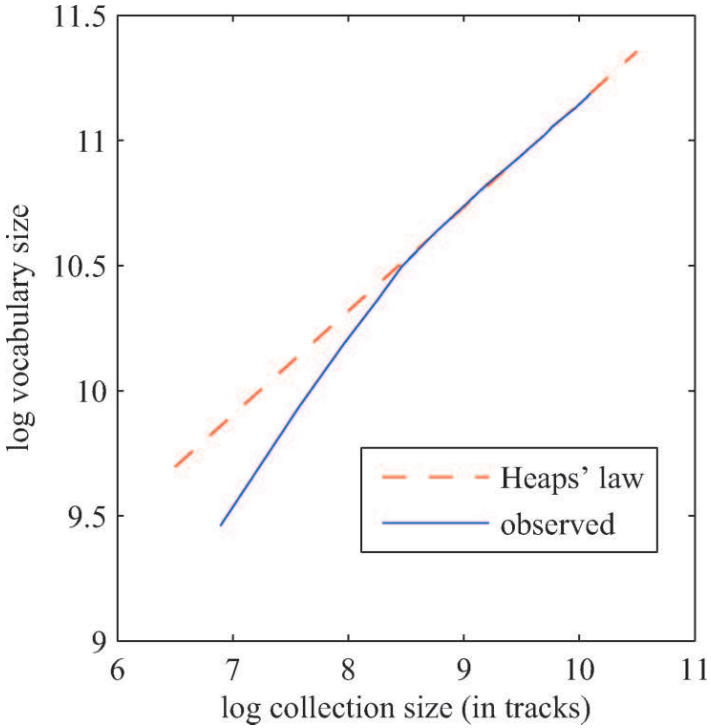
Tailles du lexique vs. corpus

AVIGNON
UNIVERSITÉ

Tailles du lexique vs. corpus
Exemples I

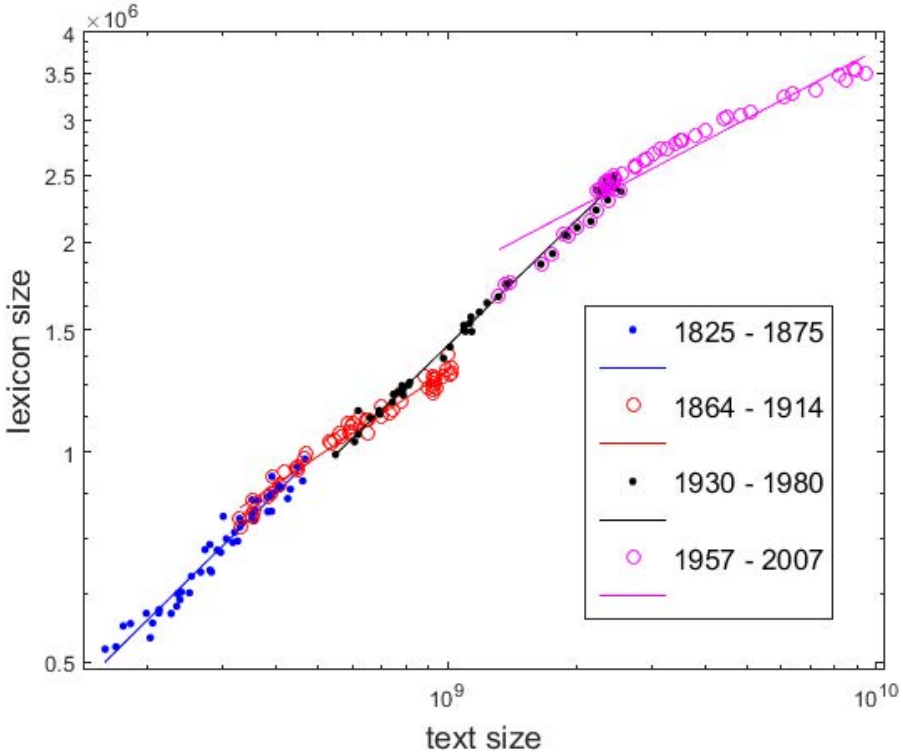


Comparaison entre plusieurs langues européennes [lien]

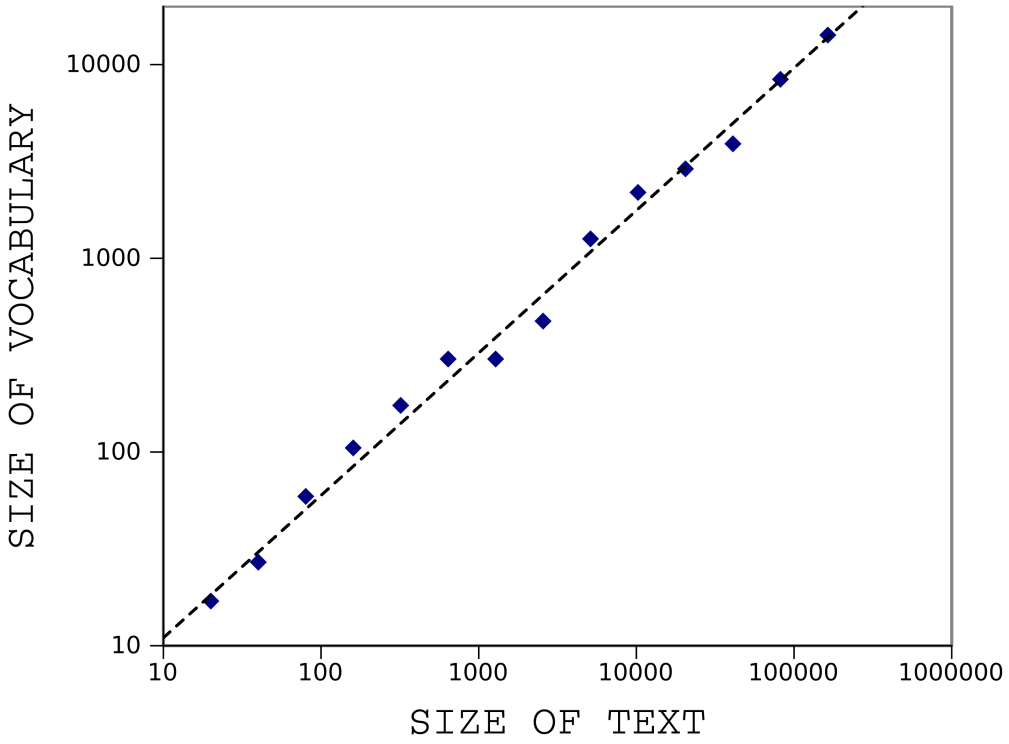


Tags des services musicaux Last.fm et MyStrands [LS08]

Tailles du lexique vs. corpus
Exemples II



Sur la base des n-grammes de Google Books [BLS16]



Dans *L'origine des espèces* (EN) de C. Darwin [ND15]

Tailles du lexique vs. corpus
Loi de Heaps

- Lexique d'un corpus : bien plus grand que dictionnaire de la langue
  - Noms propres (personnes, lieux...)
  - Codes, références (alphanumériques)
  - Entités scientifiques (molécules, gènes), jargon

Loi de Heaps

Loi **empirique** attribuée à H. S. Heaps [Hea78] et reliant la **taille du lexique** (T) à la **taille du corpus** (S). Formellement :

$$T = \alpha \times S^{\beta},$$

où α et β sont des constantes, avec généralement $30 \leq \alpha \leq 100$ et $\beta \approx 0{,}5$.

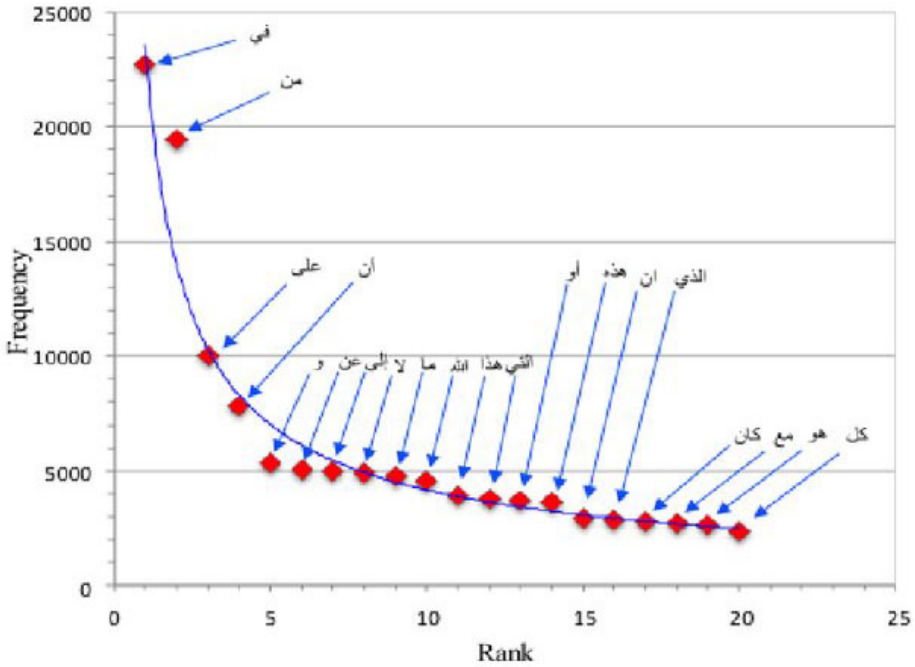
- Conséquences :
  - Taille du lexique croit avec taille de collection : pas de limite sup
  - Grand lexique pour grande collection
  - Mais cet accroissement est de moins en moins rapide

Section 3

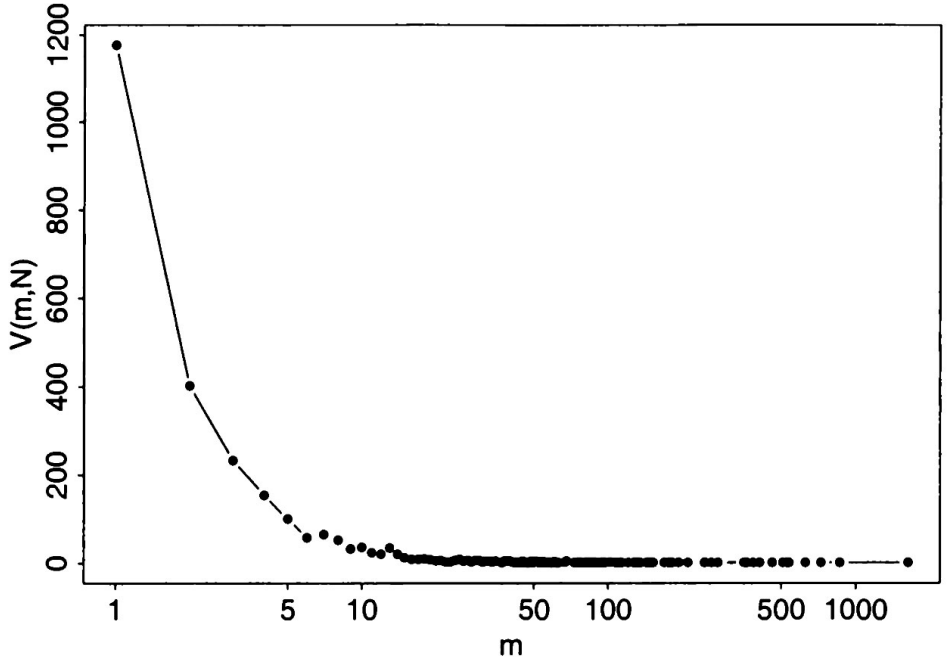
Fréquence des termes

AVIGNON
UNIVERSITÉ

Fréquence des termes
Exemples I

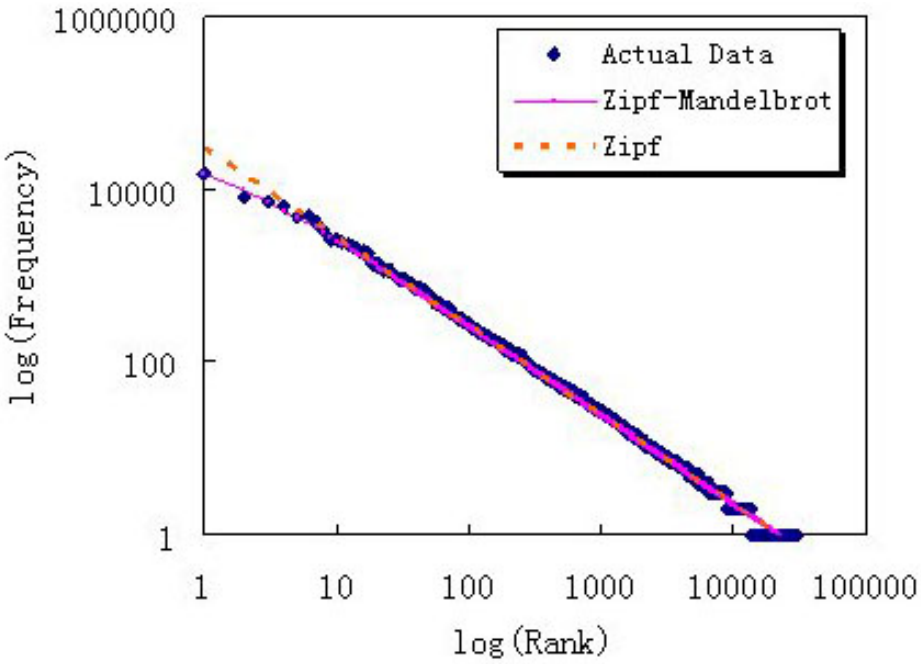


Dans la langue arabe (20 mots les plus fréquents) [MM16]

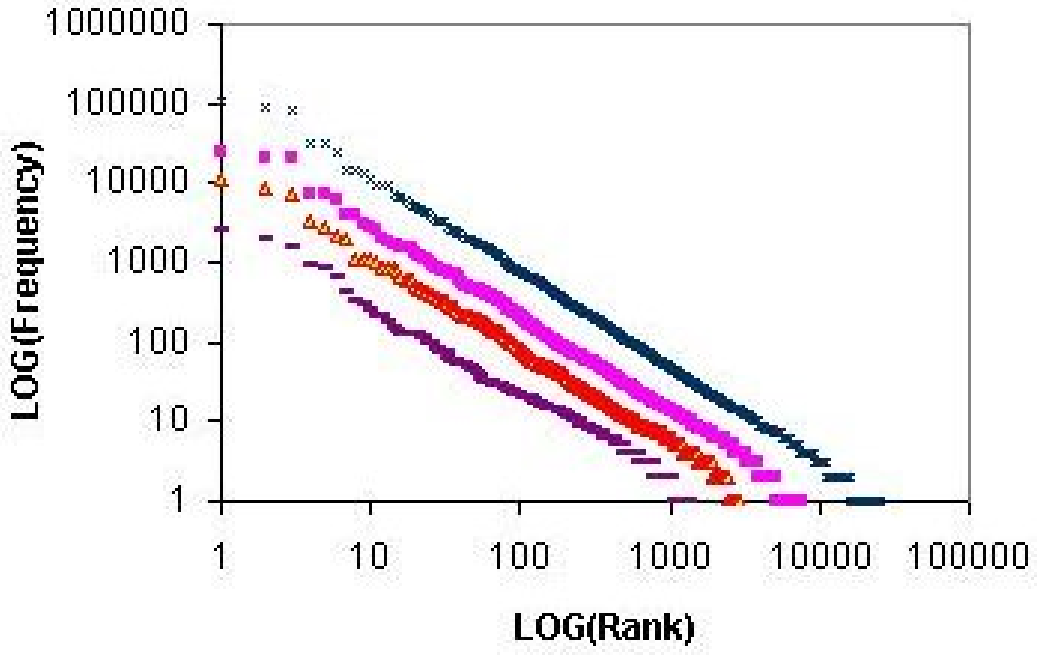


Dans *Alice au pays des merveilles* (EN) de L. Carroll [Baa01]

Fréquence des termes
Exemples II



Dans *Ulysse* (EN) de J. Joyce [Zha08]



Dans du code source Java [Zha08]

Fréquence des termes
Loi de Zipf

Loi de Zipf

Loi **empirique** attribuée à G. K. Zipf [Zip49] et reliant la **fréquence de collection** $cf(t)$ d'un terme t à son **rang** $r(t)$ dans la liste des termes classés par fréquence.
Formellement :

$$cf(t) = \alpha \times r(t)^{\beta},$$

où α et β sont des constantes, avec $\beta = -1$.

- Interprétation :
  - Le 2ème terme $(\alpha/2)$ le plus fréquent est 2 fois moins fréquent que le 1er (C)
  - Le 3ème terme $(\alpha/3)$ est 3 fois moins fréquent que le 1er
  - Etc.
- Conséquences :
  - Très peu de termes très fréquents
  - Très nombreux termes très peu fréquents

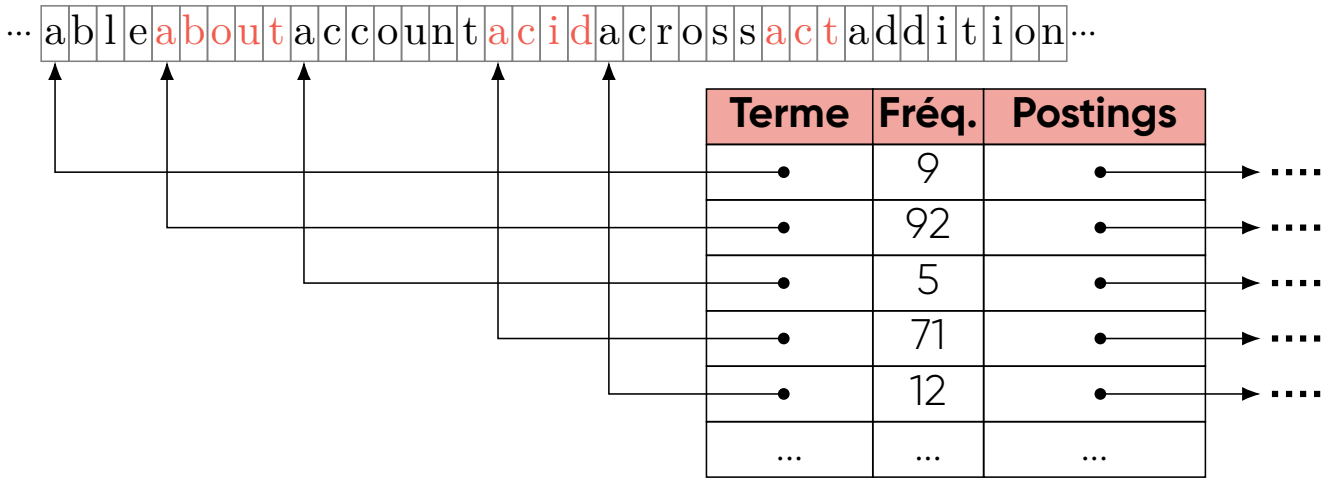
Section 4

Traitement du lexique

AVIGNON
UNIVERSITÉ

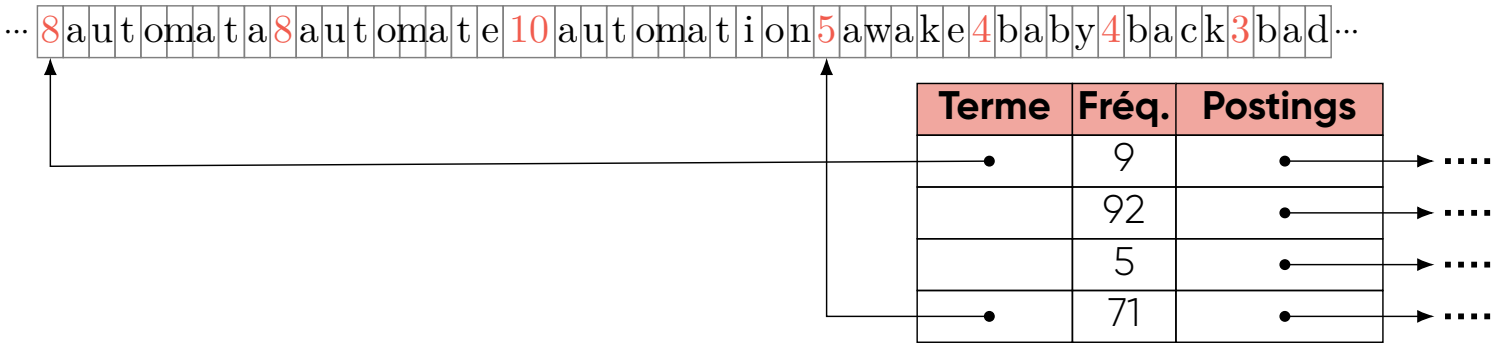
Traitement du lexique
Représentation linéaire

- Représentation linéaire
  - Termes = longue chaîne de caractères
  - Lexique = tableau d'entrées constituées de :
    - Un pointeur vers le terme
    - La fréquence de document
    - Un pointeur vers la liste de postings

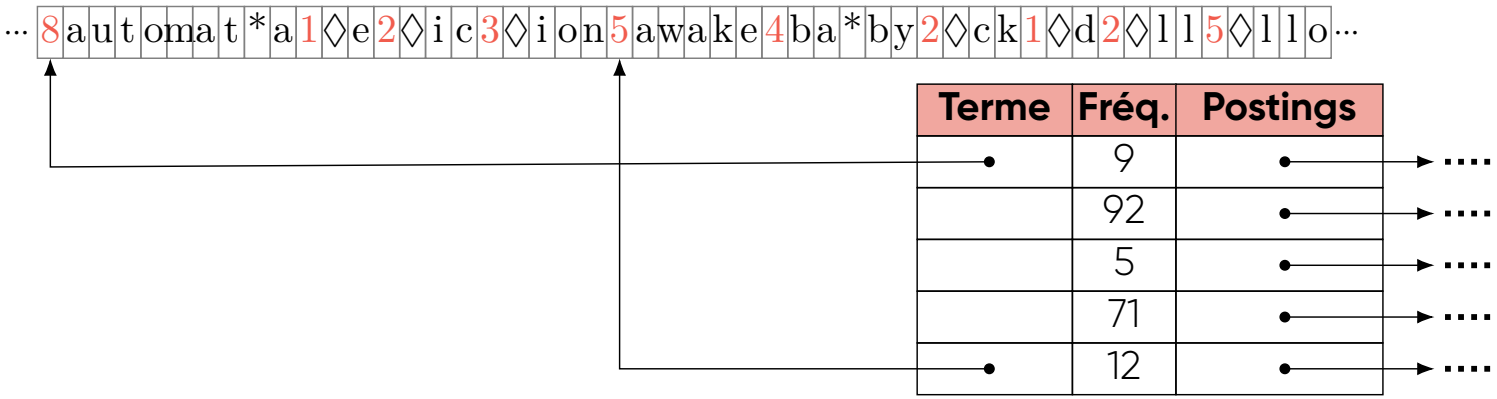


Traitement du lexique
Améliorations

- Améliorations :
  - Par bloc
    - Principe : accès un bloc à la fois (un peu moins rapide)



  - Avec mutualisation
    - Principe : ne pas répéter les préfixes communs



Section 5

Traitement des postings

AVIGNON
UNIVERSITÉ

Traitement des postings
Observations et approche basique

- Note : postings = entiers
- Approche basique :
  - Postings codés sur le nombre minimal de bits nécessaires
  - Ex. : Reuters = 800 000 documents
    - $\log_2 800\,000 = 19,6$ soit 20 bits
- Amélioration :
  - Observation : postings sont ordonnés
  - $\rightarrow$ Stocker les intervalles au lieu des postings eux-mêmes
  - Ex. : $\mathrm{Computer}$, 283154, 283159, 283202...
    - $\rightarrow$ $\mathrm{Computer}$, 283154, 5, 43...
- Valeurs des intervalles très variables
  - (en fonction de la fréquence du terme)
  - $\rightarrow$ Besoin d'un codage à taille variable

Traitement des postings
Codage à taille variable

- Codage à taille variable
  - On utilise seulement le nombre d'octets nécessaire
  - 7 bits utilisés pour coder la valeur
  - bit de continuation : **1 bit** (poids fort) indiquant si l'octet traité est terminal (1) ou pas (0)
  - Le **reste des bits** : zéros de remplissage
- Exemple :

DocId	Intervalle	Code
824	–	00000110
		10111000
829	5	10000101
215 406	214 577	00001101
		00001100
		10110001

- Note : on peut travailler avec des mots autres que l'octet (i.e. des paquets $\neq$ 8 bits)

Traitement des postings
Codage gamma

- Codage gamma :
  - Principe :
    - **Décalage** = codage binaire tronqué du 1 de poids fort
    - **Longueur** = longueur du décalage exprimé en codage unaire complété d'un 0 final de **séparation**
    - Code = concaténation de la longueur et du décalage
  - Exemple : $13 = (1101)_2$
    - Décalage : 101
    - Longueur : $3 \rightarrow 1110$
    - Code : 1110101
- Propriétés : pour une valeur v
  - Longueur du décalage : $\lceil \log_2 \ v \rceil - 1$
  - Longueur de la longueur : $\left(\lceil \log_2 \ v \rceil - 1\right) + 1 = \lceil \log_2 \ v \rceil$
  - Longueur du code : $2\lceil \log_2 \ v \rceil$
  - (Longueur optimale théorique : $\log_2 \ v$)

Section 6

Conclusion

AVIGNON
UNIVERSITÉ

Concepts abordés dans cette partie

- Représentation linéaire et ses améliorations
- Codage à taille variable
- Loi de Heaps
- Loi de Zipf
- Codage gamma

Lectures recommandées

[MRS08] *Introduction to Information Retrieval*, chapitre 5.

[BCC10] *Information Retrieval : Implementing and Evaluating Search Engines*, chapitre 6.

[BR11] *Modern Information Retrieval : The Concepts and Technology behind Search*, chapitre 9.

[AG13] *Recherche d'information - Applications, modèles et algorithmes*, chapitre 2.

[CMS15] *Search Engines : Information Retrieval in Practice*, chapitres 3 & 5.

Références bibliographiques I

[AG13] M.-R. Amini et É. Gaussier. *Recherche d'information - Applications, modèles et algorithmes.* Paris, FR : Eyrolles, 2013. url : https://www.eyrolles.com/Informatique/Livre/recherche-d-information-9782212673760/.

[Baa01] R. H. Baayen. *Word Frequency Distributions.* Dordrecht, NL : Springer, 2001. doi : 10.1007/978-94-010-0844-0.

[BR11] R. Baeza-Yates et B. Ribeiro-Neto. *Modern Information Retrieval : The Concepts and Technology behind Search.* 2nd Edition. Boston, USA : Addison Wesley Longman, 2011. url : http://people.ischool.berkeley.edu/~hearst/irbook/.

[BLS16] V. V. Bochkarev, E. Y. Lerner et A. V. Shevlyakova. « Verifying Heaps' law using Google Books Ngram data ». In : *arXiv* cs.CL (2016), p. 1612.09213. url : https://arxiv.org/abs/1612.09213.

Références bibliographiques II

[BCC10] S. Büttcher, C. L. A. Clarke et G. V. Cormack. *Information Retrieval : Implementing and Evaluating Search Engines*. Cambridge, USA : MIT Press, 2010. url : http://www.ir.uwaterloo.ca/book/.

[CMS15] W. B. Croft, D. Metzler et T. Strohman. *Search Engines : Information Retrieval in Practice*. Pearson, 2015. url : http://www.search-engines-book.com/.

[Hea78] H. S. Heaps. *Information Retrieval : Computational and Theoretical Aspects*. Academic Press, 1978. url : https://dl.acm.org/citation.cfm?id=539986.

[LS08] M. Levy et M. Sandler. « Learning Latent Semantic Models for Music from Social Tags ». In : *Journal of New Music Research* 37.2 (2008), p. 137–150. doi : 10.1080/09298210802479292.

[MRS08] C. D. Manning, P. Raghavan et H. Schütze. *Introduction to Information Retrieval*. New York, USA : Cambridge University Press, 2008. url : http://www-nlp.stanford.edu/IR-book/.

Références bibliographiques III

[MM16] A. Masrai et J. Milton. « How Different Is Arabic from Other Languages ? The Relationship between Word Frequency and Lexical Coverage ». In : *Journal of Applied Linguistics and Language Research* 3.1 (2016), p. 15–35. url : http://www.jallr.com/index.php/JALLR/article/view/213/pdf213.

[ND15] E. Najafi et A. H. Darooneh. « The Fractal Patterns of Words in a Text : A Method for Automatic Keyword Extraction ». In : *PLoS ONE* 10.6 (2015), e0130617. doi : 10.1371/journal.pone.0130617.

[Zha08] H. Zhang. « Exploring Regularity in Source Code : Software Science and Zipf's Law ». In : *15th Working Conference on Reverse Engineering*. Antwerp, BE, 2008. doi : 10.1109/WCRE.2008.37.

[Zip49] G. K. Zipf. *Human Behavior and the Principle of Least Effort - An Introduction to the Principle of Human Ecology.* Addison-Wesley, 1949. url : https://archive.org/details/in.ernet.dli.2015.90211.