Chap. 3: Indexation : Techniques de pondération et Statistiques sur le texte

Pondération des mots

- Comment caractériser les termes importants dans un document?
- …… Pondération des termes
- Idée sous jacente :
  - Les termes importants doivent avoir un poids fort

Approches de pondération

- Plusieurs approches :
  - **Tf, IDF** approche plus répandue
  - Pourvoir discriminatoire d'un terme
  - Modèle 2 poisson
  - Clumping model
  - Modèle de Langage

- Dépend aussi du modèle de RI.

tf.idf

- *tf* : Idée sous jacente : plus un terme est fréquent dans un document plus il est important dans la description de ce document
  - Exemple de *tf* :

$$tf = \begin{cases} freq(t,d) \\ 1+\log(freq(t,d)) \\ \dfrac{freq(t,d)}{\max_{\forall t' \in d}(t',d)} \\ \dfrac{freq(t,d)}{\sum_{\forall t' \in d} freq(t',d)} \end{cases}$$

Taille (longueur) du document

  - "Okapi tf" : K introduit pour tenir compte de la longueur des documents

$$tf / (K+tf)$$

$$tf = \frac{freq(t,d)}{k1.(1-b+b*\frac{dl}{avgdl})+freq(t,d)}$$

tf.idf

- IDF : (Inverse Document Frequency) la frequence du terme dans la collection

$$idf(t) = \begin{cases} \log\left(\frac{N}{n_t}\right) \\ \log(\frac{N - n_t}{n_t}) \end{cases}$$

avec
N : le nombre de documents de la collection,
n_t : le nombre de documents contenant le terme t

Tf.Idf

– *Quelques formules répandues en RI*

$$w(t,d) = tf * idf = \begin{cases} \dfrac{(1+\log(freq(t,d)) * \log\dfrac{N}{n_t}}{\sum_{\forall t' \in d} (1+\log(freq(t',d)) * \log\dfrac{N}{n_{t'}}} \\ \dfrac{freq(t,d)}{k1.(1-b+b*\dfrac{dl}{avgdl}) + freq(t,d)} * \log\dfrac{N-n_t}{n_t} \end{cases}$$

Facteur de normalisation

Tf.Idf

- Exploitation en RI
  - Retour (transp.→ , calcul score d'un document)
  - Soit une requête q(t1, t2) et document d(t1, t2, ..tn)
  - Calculer le score de document vis-à-vis de la requête

→ Faire la somme pondérée des termes de la requête apparaissant dans le document

$$score(q,d) = \sum_{t \in q} w(t,d)$$

Ce point sera détaillé dans le chapitre Modèles de RI

Quelques Statistiques sur le texte

- La fréquence d'apparition d'un terme dans une collection est un bon indicateur de l'importance de ce terme.

Quelques Statistiques sur le texte

- "Principle of Least Effort" (Zipf)
  - Il est plus simple pour un auteur (rédacteur d'un document) de répéter les mots que d'en chercher de nouveaux.

term frequency decreases rapidly
as a func2on of rank!

Exemple de mots fréquents

Frequent Word	Number of Occurrences	Percentage of Total
the	7,398,934	5.9
of	3,893,790	3.1
to	3,364,653	2.7
and	3,320,687	2.6
in	2,311,785	1.8
is	1,559,147	1.2
for	1,313,561	1.0
The	1,144,860	0.9
that	1,066,503	0.8
said	1,027,713	0.8

Artifact of InQuery's stemming technique

Frequencies from 336,310 documents in the 1GB TREC Volume 3 Corpus
125,720,891 total word occurrences; 508,209 unique words

Loi de Zipf

- Loi de Zipf

  Rang . Pr≈ constante

  - Pr = fréquence du terme de rang r/N
  - N nombre total d'occurrence
  - A ≈ 0.1

Exemple loi de Zipf

Word	Freq	r	Pr(%)	r*Pr
the	2,420,778	1	6.488	0.0649
of	1,045,733	2	2.803	0.0561
to	968,882	3	2.597	0.0779
a	892,429	4	2.392	0.0957
and	865,644	5	2.32	0.116
in	847,825	6	2.272	0.1363
said	504,593	7	1.352	0.0947
for	363,865	8	0.975	0.078
that	347,072	9	0.93	0.0837
was	293,027	10	0.785	0.0785
on	291,947	11	0.783	0.0861
he	250,919	12	0.673	0.0807
is	245,843	13	0.659	0.0857
with	223,846	14	0.6	0.084
at	210,064	15	0.563	0.0845
by	209,586	16	0.562	0.0899
it	195,621	17	0.524	0.0891
from	189,451	18	0.508	0.0914
as	181,714	19	0.487	0.0925
be	157,300	20	0.422	0.0843
were	153,913	21	0.413	0.0866
an	152,576	22	0.409	0.09
have	149,749	23	0.401	0.0923
his	142,285	24	0.381	0.0915
but	140,880	25	0.378	0.0944

Word	Freq	r	Pr(%)	r*Pr
has	136,007	26	0.365	0.0948
are	130,322	27	0.349	0.0943
not	127,493	28	0.342	0.0957
who	116,364	29	0.312	0.0904
they	111,024	30	0.298	0.0893
its	111,021	31	0.298	0.0922
had	103,943	32	0.279	0.0892
will	102,949	33	0.276	0.0911
would	99,503	34	0.267	0.0907
about	92,983	35	0.249	0.0872
i	92,005	36	0.247	0.0888
been	88,786	37	0.238	0.0881
this	87,286	38	0.234	0.0889
their	84,638	39	0.227	0.0885
new	83,449	40	0.224	0.0895
or	81,796	41	0.219	0.0899
which	80,385	42	0.215	0.0905
we	80,245	43	0.215	0.0925
more	76,388	44	0.205	0.0901
after	75,165	45	0.201	0.0907
us	72,045	46	0.193	0.0888
percent	71,956	47	0.193	0.0906
up	71,082	48	0.191	0.0915
one	70,266	49	0.188	0.0923
people	68,988	50	0.185	0.0925

Top 50 words from 84,678 Associated Press 1989 articles
(37,309,114 word occurrences, lowercased, punctuation removed, 266MB)

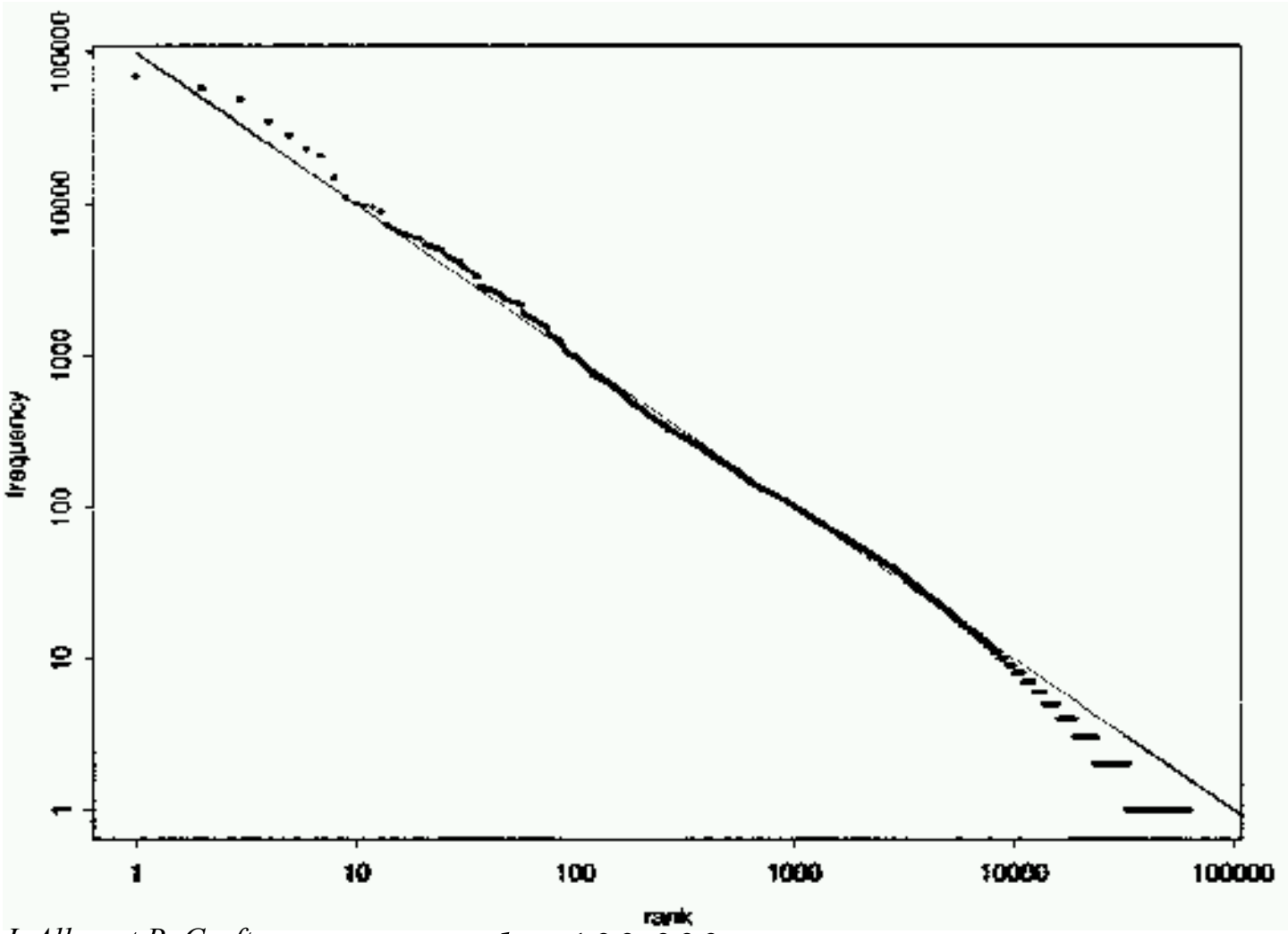
Est que les données suivent réellement la loi de Zipf ?

- Une loi de la forme $y = kx^c$ est appelée loi puissance.

- Est une loi de puissance $c = -1$
  - $r = (A*N)\cdot n^{-1}$ et $n = (A*N)\cdot r^{-1}$
  - A*N est une constante pour une collection donnée

- En passant à un logarithme.
  - $\log(n) = \log(A*Nr^{-1}) = \log(A*N) - 1\cdot\log(r)$

Exemple loi de Zipf le corpus Brown



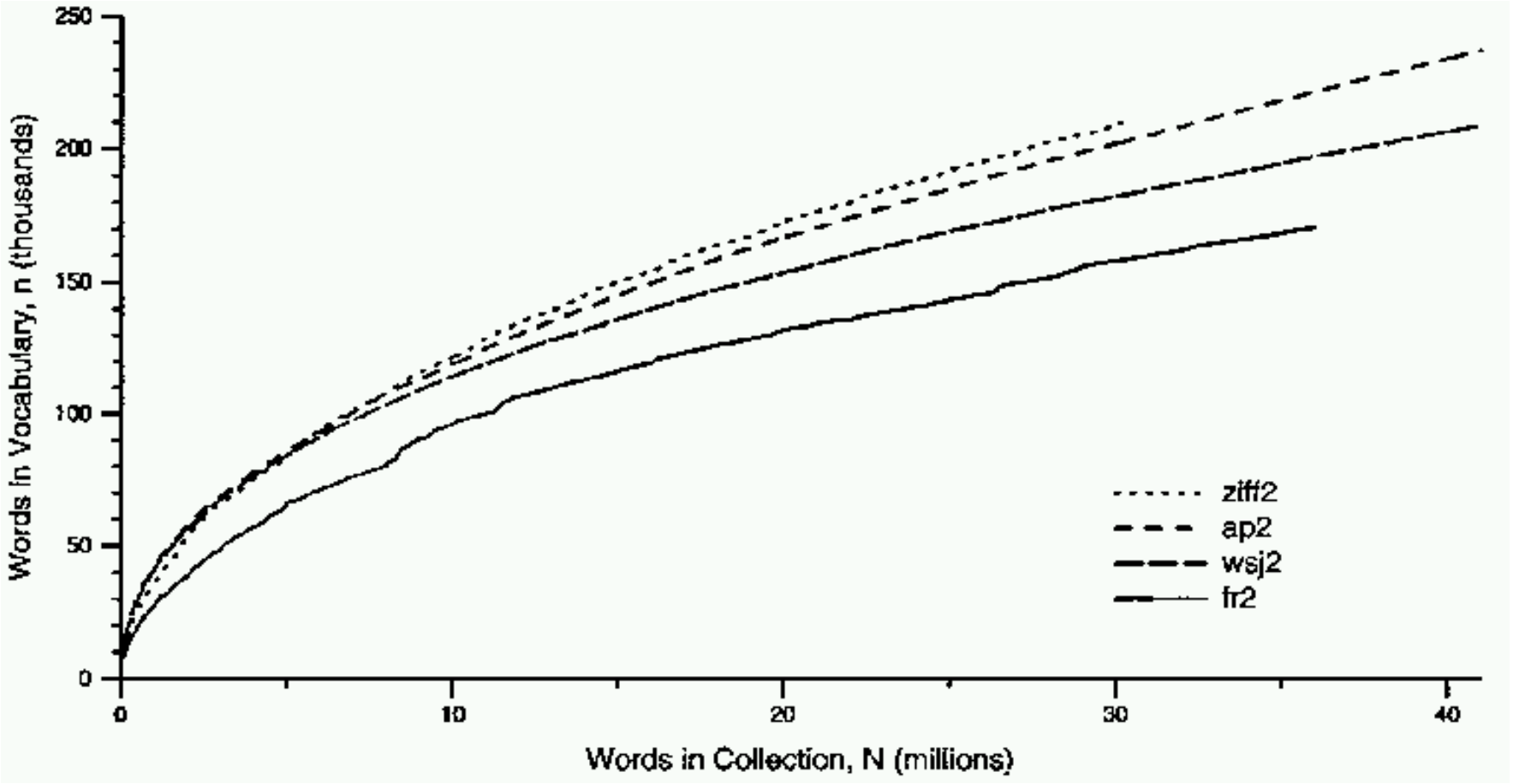
$$k = 100{,}000$$

Transparent J. Allan et B. Croft

Accroissement du vocabulaire de l'index (loi de Heap)

- La taille de l'index croît de manière logarithmique
  - L'index n'a pas borne supérieure (noms propres, erreurs de typos, etc.)
  - Mais, les nouveaux mots apparaissent moins fréquemment quand le vocabulaire croît.
- Considérons V la taille de l'index(en nombre de mots) et n le nombre de documents dans le corpus
  - $V = Kn^{\beta}$ $(0< \beta <1)$
  - Constantes typiques :
    - $K \approx 10-100$
    - $\beta \approx 0.4-0.6$ (approx. Racine carré de n)

Loi de Heap



Transparent J. Allan et B. Croft