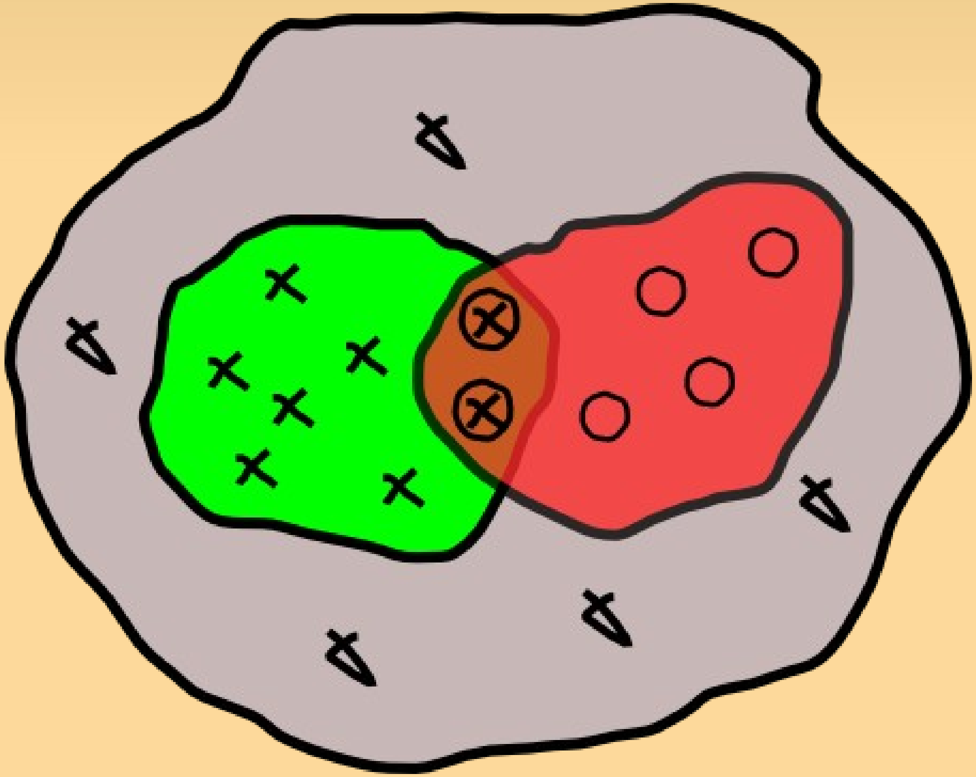
Quelques techniques d’extraction sémantique

Adrian Dimulescu
Telecom ParisTech
30 mars 2009

Qu’est-ce que le sens ?

Le sens est le contexte

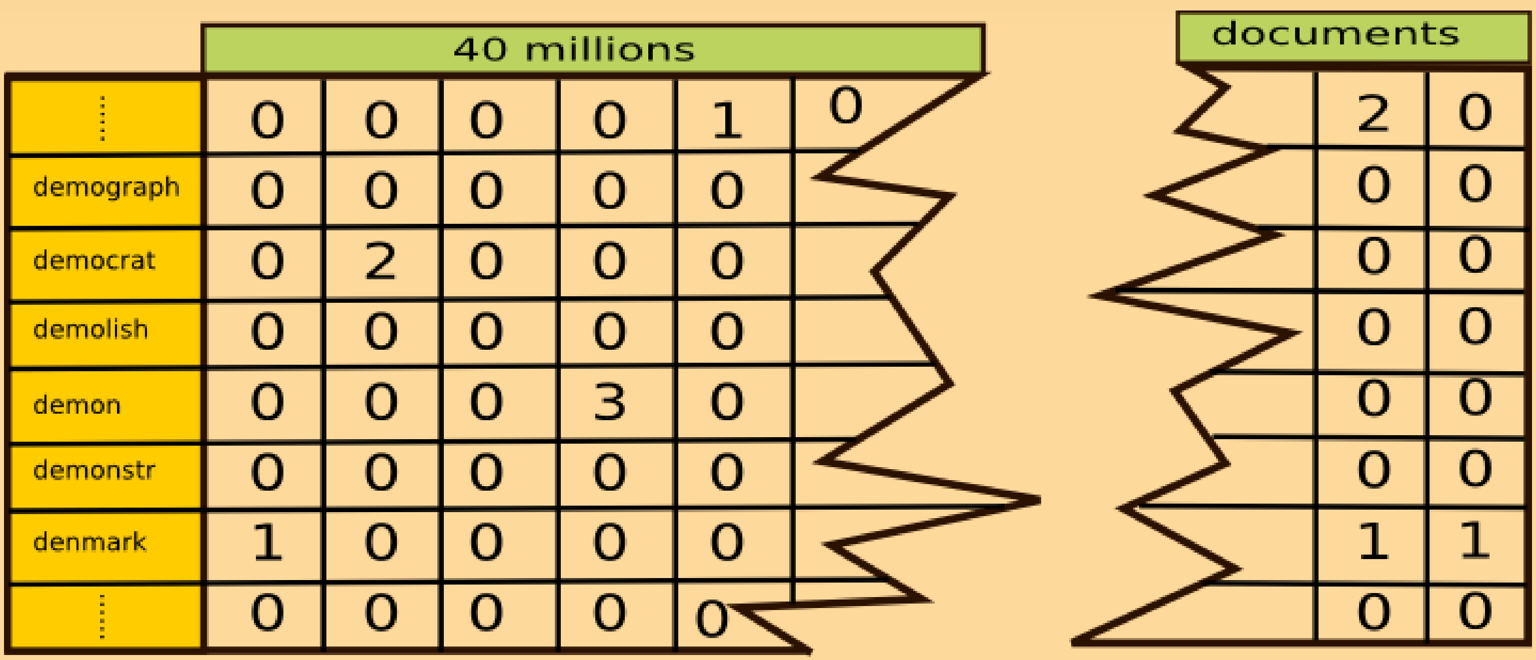


Plan

- Latent semantic analysis
- Complexité de Kolmogorov
- Distance informationnelle
- Quelques exemples

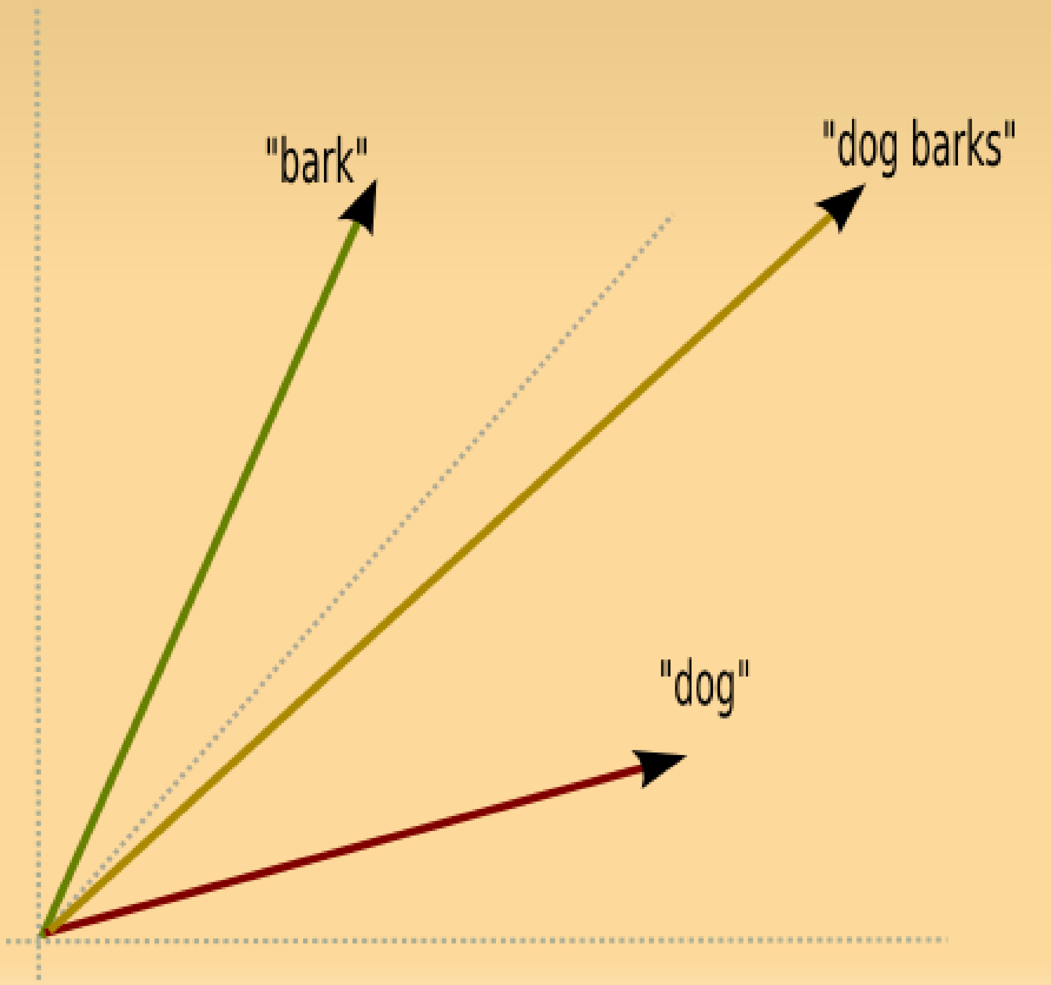
Latent Semantic Analysis

- Matrice éparse énorme: terms x documents
- Normalisation: log(cell)/entropie



Latent Semantic Analysis

- Tout mot/concept est un vecteur
- Tout prédicat/phrase est un vecteur
- Tout document est un vecteur



LSA

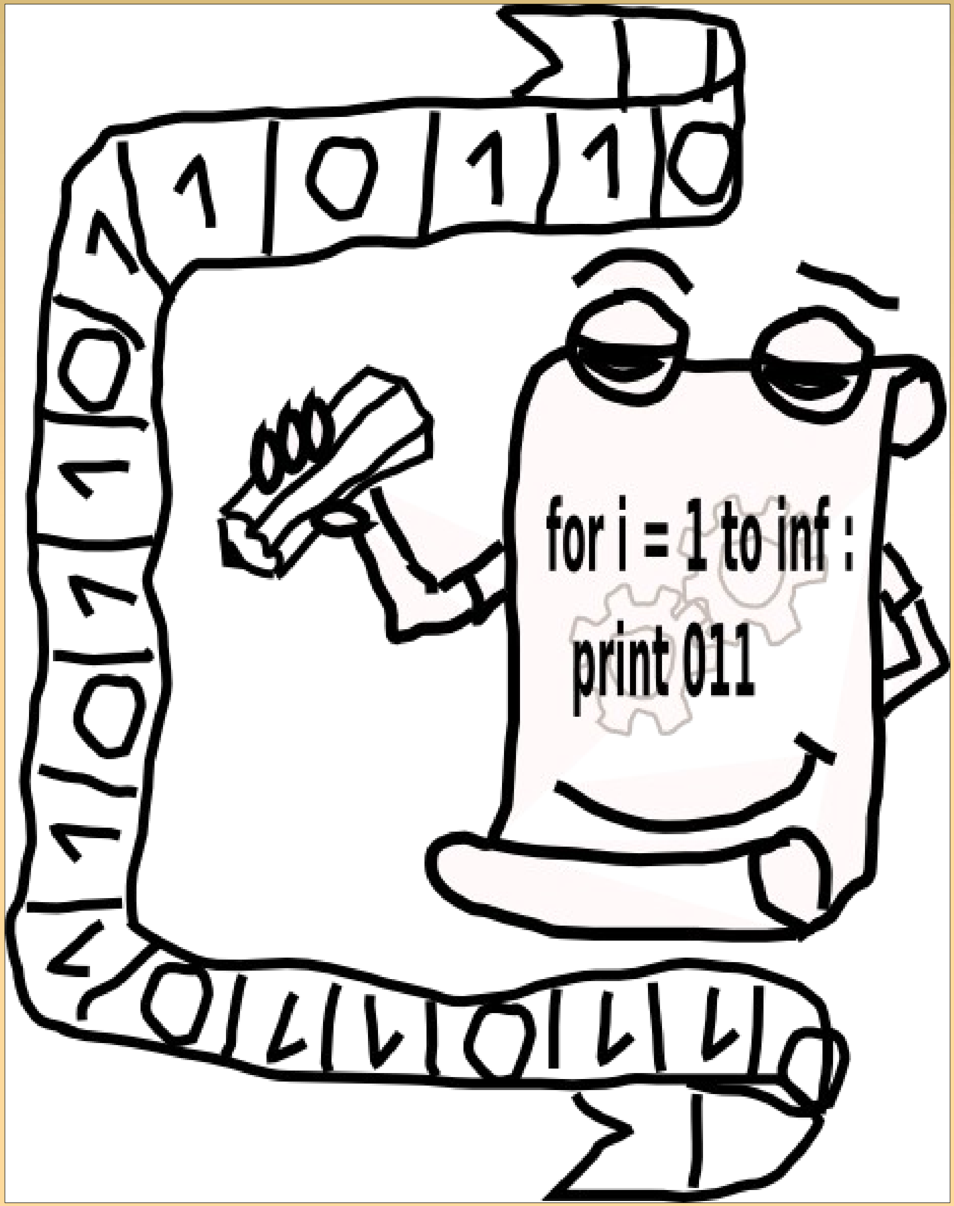
- SVD : réduction de dimensionalité
  - “compression” : approx 300 colonnes
- Mesure de similarité: cosinus des vecteurs
- Inconvenients
  - Non-incrémental
  - Passe mal à l’échelle
- Random indexing
  - Évite la grande matrice éparse du début

Kolmogorov complexity

011011011011011011011011011011011011011
011011011011011011011011011011011011011
011011011011011011011011011011011011011
011011011011011011011011011011011011011
011011011011011011011011011011011011011
011011011011011011011011011011011011011
011011011011011011011011011011011011011
011011011011011011011011011011011011011
011011011011011011011011011011011011011

Kolmogorov complexity

01101101101101101101
01101101101101101101
01101101101101101101
01101101101101101101
01101101101101101101
01101101101101101101
01101101101101101101
01101101101101101101
01101101101101101101



Kolmogorov complexity

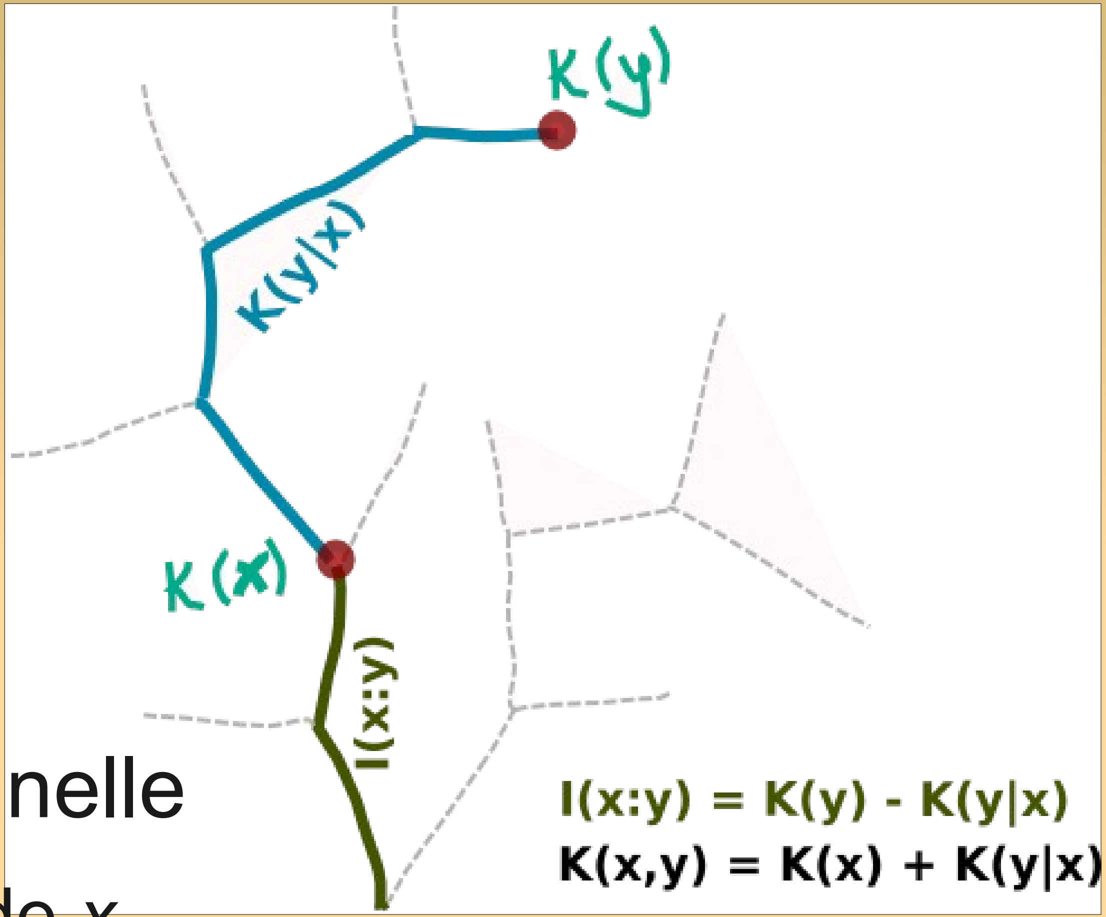
0010010000111111011010101000100010000010
1101000110000100011010011000100110000110
0110001010001011100000001101110000001110
0110100010010100100000001001001110000010
0010001010011001111100110001110100000000
0100000010111011111010100110001110110010
0111001101100100010010100010100101000000
1000011110011000111000110100000000100110
1110111101111110010101000110011011001111
0011010011101001000011000110110011000000
0101011000010100110110111110010010111111
0001010000110111010011111111000010011010
1011011010110110101010100011100001001001

Kolmogorov complexity

0010010000111111011010101000100010000 10
1101000110000100011
0110001010001011100
0110100010010100100
0010000101001100111 1
0100000010111011111 0
0111001101100100010
1000011110011000111
1110111110111110010 1
0011010011101001000
0101011000010100110
0001010000110111010
1011011010110110101010001110000010010001

Kolmogorov complexity

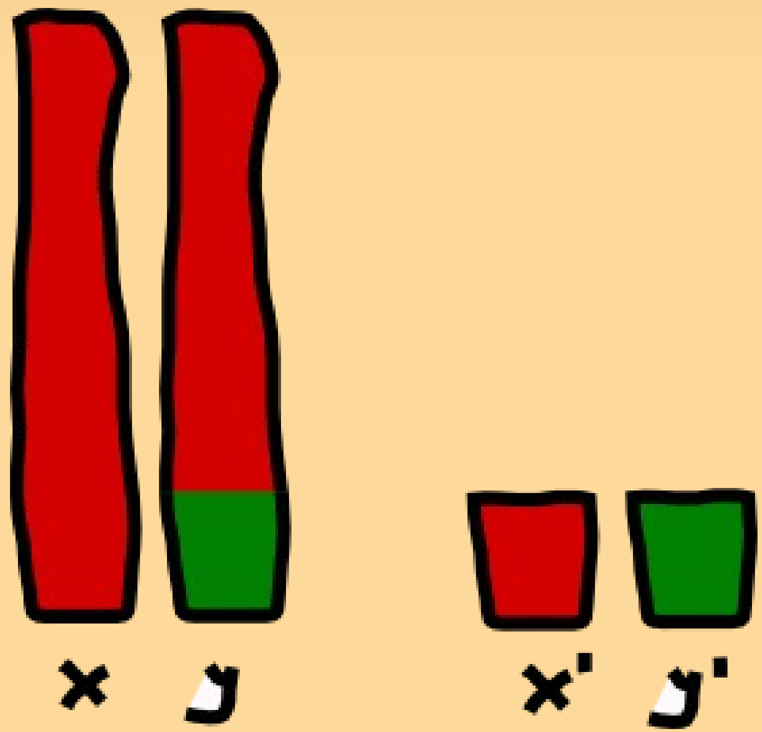
- Absolue
- Non-calculable
  - Mais approximable: Gzip, bzip2 etc.



- Complexité conditionnelle
  - *K(x|y)* la complexité de *x* si l'on a déjà *y* en entrée (≤ *K(x)*)
  - I(x:y) ≈ I(y:x)

Information distance

- E(x,y)= max {K(x|y), K(y|x)}
  - Le program qui transforme *x* en *y* et l'inverse
  - The largest transformation program "includes" the smallest (almost)
  - Absolue, minorise tout
- Normalization
  - Quelle paire est plus similaire en couleur ?
  - d(x,y) = K(y|x) / K(y)



Normalized compression distance

- Etant donne le compresseur *C* (ex. Bzip2)
  - Comme C(y|x) = C(x,y) – C(x), si C(y) > C(x)
  - $$NCD(x, y) = \frac{C(xy) - C(x)}{C(y)}$$
  -
  - On peut donc calculer des distances entre fichiers, génomes, romans, pièces musicales, en comprimant les objets séparément et ensemble
  - Algorithmes de clustérisation sans paramètres

Le compresseur doit être *normal*

- Un compresseur *C* est *normal* si :
  - Idempotence: C(xx) = C(x)
  - Monotonicité: C(xy) >= C(x)
  - Symmetrie: C(xy) = C(yx)
  - Distributivité: C(xy)+C(z) <= C(xz)+C(yz)
- Alors seulement la distance NCD est une metrique de similarité:
  - D(x,y) = 0 iff x = y; D(x,y) = D(y,x), et
  - D(x,y) <= D(x, z)+D(z,y) (triangle ineq)
  - Et contrainte de densité: $\sum_{y} 2^{-d(x,y)K(x)} \leqslant 1$

Pourquoi ces formules?

- De toute façon quand on veut approximer *K* on est en état de pèche
  - On ne sait pas si on approxime bien ou pas
- La compression peut donner des résultats intéressants
  - mais uniquement si le compresseur est (quasi)normal
- Chercher donc des compresseurs normaux

Normalized Google Distance

- Google comme compresseur
  - Pourquoi Google comprime : le code dérivé des probabilités G(x) = -log *p(x)* est minimal. Si hits(*y*)< hits(*x*):

$$NGD(x,y)=\frac{G(xy)-G(x)}{G(y)}$$

$$NGD(x,y)=\frac{\log hits(y)-\log hits(xy)}{\log N-\log hits(x)}$$

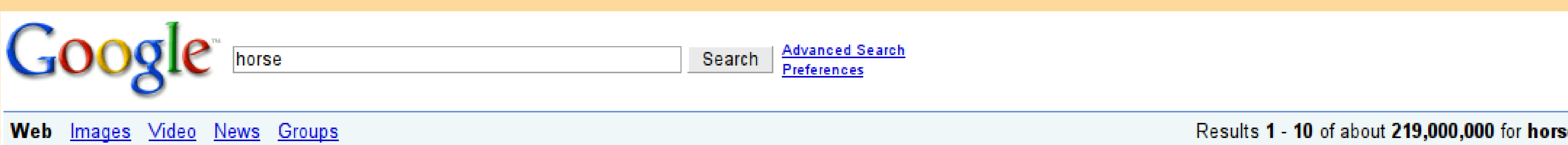
Google JSON API

http://ajax.googleapis.com/ajax/services/search/web?v=1.0&q=horse&rsz=small

{"responseData": {"results":[{"GsearchResultClass":"GwebSearch","unescapedUrl":"
http://en.wikipedia.org/wiki/Horse",
"url":"http://en.wikipedia.org/wiki/Horse",
"visibleUrl":"en.wikipedia.org",e: Your Guide to Equine Health
Care","titleNoFormatting":"The Horse: Your Guide to Equine Health
Care","content":"\u003cb\u003eHorse\u003c/b\u003e health news and veterinarian-
approved equine health care information from TheHorse.com. Learn more about basic
care, injuries, diseases, lameness, \u003cb\u003e...\u003c/b\u003e"}],

....

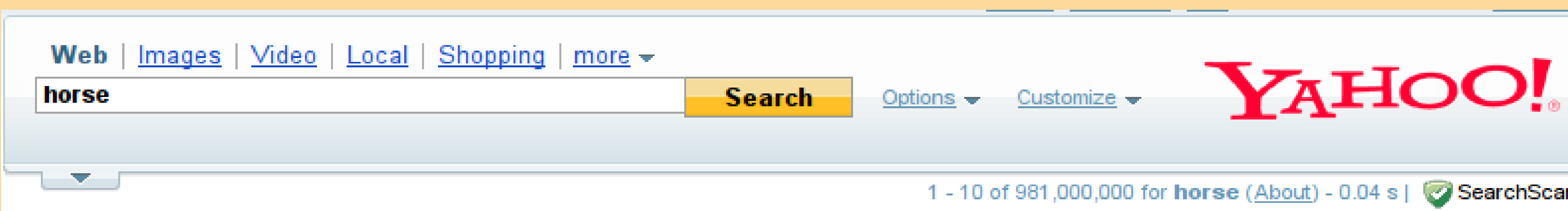
"cursor":{"pages":[{"start":"0","label":1},{"start":"4","label":2},
{"start":"8","label":3},{"start":"12","label":4},{"start":"16","label":5},
{"start":"20","label":6},{"start":"24","label":7},
{"start":"28","label":8}],**"estimatedResultCount":"27700000"**,"curren
tPageIndex":0,"moreResultsUrl":"http://www.google.com/search?
oe\u003dutf8\u0026ie\u003dutf8\u0026source\u003duds\u0026start\u003d0\u0026hl\u003d
fr\u0026q\u003dhorse"}}, "responseDetails": null, "responseStatus": 200}



Yahoo BOSS API

http://boss.yahooapis.com/ysearch/web/v1/horse?appid=...&format=xml (or json)

```
<ysearchresponse responsecode="200">
<nextpage>
/ysearch/web/v1/horse?count=10&appid=...&format=xml&start=10
</nextpage>
<resultset_web count="10" start="0"
totalhits="32651389" deephits="981000000">
....
```



Exemple : street - building

log(f('street')) = log2(134,000,000) = 26.9976577597819; g = f/N = 0.0067; G = log(1/g) = 7.22162318909168

log(f('building')) = log2(74,400,000) = 26.1487992855448; g = f/N = 0.00372; G = log(1/g) = 8.07048166332878

log(f('"street" "building"')) = log2(76,700,000) = 26.1927232419397; g = f/N = 0.003835; G = log(1/
= 8.02655770693388

(26.9976577597819 - 26.1927232419397) / (34.2192809488736 – 26.1487992855448)

NGD('street', 'building') = 0.0997381013204846

Exemple : street - slap

log(f('street')) = log2(134,000,000) = 26.9976577597819; g = f/N = 0.0067; G = log(1/g) = 7.22162318909168

log(f('slap')) = log2(3,700,000) = 21.8190938400658; g = f/N = 0.000185; G = log(1/g) = 12.4001871088079

log(f('"street" "slap"')) = log2(258,000) = 17.9770115400853; g = f/N = 1.29e-05; G = log(1/g) = 16.2422694087883

(26.9976577597819 - 17.9770115400853) / (34.2192809488736 – 21.8190938400658)

NGD('street', 'slap') = 0.727460492373477

Indexation de Wikipedia

- Lucene : outil Java d'indexation
  - Snowball English stemmer; wikipedia tokenizer
- Indexation
  - par paragraph (petite fenêtre de coccurence)
  - NGD à l'origine se fait sur Google par page
  - en.wikipedia : 18GB text
  - 8M pages
  - 40M paragraphs

Indexation de Wikipedia

- 7 millions terms
  - 7k avec freq > 10.000 (ex: kindergarten, basket)
  - 31k avec freq > 1000 (ex: radiocarbon, kryptonit)
  - 92k freq > 200 (ex: hölder, ipsilater, jolanda)
- Counting hits a posteriori is slow for lots of documents
- Solution: matrice de coocurrence pendant le parsing
  - Matrice de coocurrence doit rentrer en RAM
  - Optimisation : symmetrique, zero diag

Carte sémantique

- Calculer *toutes* les distances entre *tous* les mots (10.000)
- Pour chaque mot récupérer les voisins les plus proches
  - Mémoire sémantique: on peut retrouver les “voisins” d'un concept
- Difficile sur un vrai moteur de recherche public
  - Mais faisable sur son petit home-grown search engine

Exemples

- *HUNT*: prey, witch, deer, fish, hound, wild, predat, treasur, herd, trap, wildlif, dog, bounti, hike, lodg,
- *LEADERSHIP*:rite, resign, organiz, faction, skill, caucus, leader, membership, apostol,cabinet, coalit
- *FRANCE*:french, département, commune, pari, région, belgium, itali, comté, germani, inse, arrondiss
- *CHILDREN*:marri, husband, marriag, coupl, older, alon, spous, togeth, someon, household, individu
- *BABY*:pregnant, cry, gonna,infant, wanna, child, ain't, mama, sweet, tonight, doll, daddy, goodby, mother

Exemples (2)

- *ROAD*: junction, highway, terminus, intersect, rout, traffic, lane, motorway, bypass, parkway,
- *STREET*:avenu, downtown, boulevard, wall, corner, intersect, manhattan, neighborhood, coron, shop
- *MONSTER*: creatur, dungeon, beast, alien, horror, unleash, evil, summon, dragon, demon, loch.
- *SEVENTH* :eighth, sixth, ninth, fifth, tenth, eleventh, twelfth.
- *TERRY* :ron, bobbi, jimmi, ted, larri, jeff, ken
- *VICEROY*:granada, marquess, napl, peru, spaniard, río

Mais... (words without meaning)

- *WHICH (freq: 2M)*: there, been, had, into, also, alon, some, most, them, but, onli, howev
- *BAD (200k)*: faith, isn't, realli, doesn't, sure, can't, agre, someon, thing, obvious,
- *GOOD (500k)*: think, don't, i'm, thing, realli, look, faith, seem, sure, veri, i'v, isn't, know, say, get
  - *Mais la fréquence n'est pas tout: PERFORM (lui aussi 500k)*:concert, theatr, stage, audienc, orchestra, solo, ensembl, soloist, sing, tour, repertoir, festiv, theater, singer, danc, grammi, musician

LSA vs. NGD: Cosinus vs. K(x|y)

- Ex (session Octave):

```
$ x = 50 * rand(1,300) - 25
$ y = 1 + x
$ x * y'  / (norm(x) * norm(y))
0.99753
$ y = x .^ 2
$ x * y'  / (norm(x) * norm(y))
0.097678
```

- (une transformation mathématiquement simple peut mener à des différences structurales importantes)

Bibliographie

- Bennett et al, *The information distance,*
- Li et al, *The similarity metric*
- Cilibrasi, Vitanyi, *Clustering by compression*
- Cilibrasi, Vitanyi, *The Google similarity distance*
- Landauer, *A solution to Plato's problem: the LSA...*, 1997
- Kintsch. *Comprehension. A paradigm for cognition*, 2000