

Recherche d'information textuelle

P. Gallinari

LIP6

Université Paris 6

Patrick.Gallinari@lip6.fr

www-connex.lip6.fr/~gallinar/

Master Informatique M2 : [Apprentissage pour la recherche d'information textuelle et multimédia](#)

Plan

- Introduction
- Recherche d'information textuelle
 - Notions de base, modèles de recherche
- Recherche Web
- Modèles latents
 - Représentation et analyse des corpus

☐ Intervenants

- Patrick Gallinari (RI texte)
- Sylvain Lamprier (RI texte)
- Sabrina Tollari (RI image –video)
- Nicolas Thome (RI image –video)

☐ Ressources

■ Livres

- ☐ [Christopher D. Manning](#), [Prabhakar Raghavan](#) and [Hinrich Schütze](#), *Introduction to Information Retrieval*, Cambridge University Press. 2008.
- ☐ [Massih-Reza Amini](#), Eric Gaussier, *Recherche d'information, Applications, modèles et algorithmes*, Eyrolles 2013
- ☐ W. Bruce Croft, Donald Metzler, Trevor Strohman, *Search Engines Information Retrieval in Practice*, Addison Wesley, 2009

☐ Evaluation

- 50% examen, 50% TP

Introduction

Problèmes de base pour construire un système d'accès à l'information

- Acquisition
 - e.g. crawling et prétraitement (diversité des types de documents)
- Représentation - indexation, cf W3C, MPEG7
 - non structuré (texte, image), semi structuré e.g. video
- Modèle de recherche
 - présenter des informations pertinentes à l'utilisateur, e.g. liste ordonnée selon un critère
- Interaction utilisateur
 - feedback, recherche interactive, la RI est un processus centré utilisateur
- Evaluation
 - Protocole d'évaluation, e.g. Cranfield, mesures e.g. rappel-précision
- Et puis pour les données du web
 - Dynamicité
 - Performance
 - Passage à l'échelle
 - quantité de données (tera), stockage distribué
 - Adaptation du système
 - e.g. amélioration de composants

Diversité des sources d 'information

- Texte
 - Articles, livres (pdf, ps, ebook, html, xml,...)
- Images, Video, Son, Musique
- Web (pages, sites)
 - Dynamicité
- Sites sociaux : blogs, Twitter, etc
 - Dynamicité, structure relationnelle
- Messageries - fils de discussion, etc
- Information majoritairement peu structurée, mais structures exploitables (HTML, XML), relations (web réseaux sociaux), hiérarchies, ...

Diversité des demandes d'accès à l'information

- ☐ Consultation (browsing)
- ☐ Requêtes booléennes, mots clés
- ☐ Recherche automatique (e.g. robots)
- ☐ Suivi d'évènement, analyse de flux
- ☐ Extraction d'information plein texte, web caché, etc
- ☐ ...

Exemples de tâches en RI classique

- Trouver parmi un ensemble d'articles ceux qui concernent un sujet spécifique : pertinence d'un document ?
- Faire un résumé du contenu d'un document ou d'un ensemble de documents (éventuellement sur un sujet)
- Structuration (classification) automatique d'un ensemble de documents (groupes)
- Trouver dans un document les passages pertinents, les informations pertinentes concernant un sujet (mots - phrases)
- Suivre dans une collection d'articles l'évolution d'un sujet, Changements de sujets
- Veille scientifique - technique, Surveiller la concurrence
- Guetter l'arrivée d'informations (appels d'offre, CFP, nouveaux produits, ...)
- Dialoguer avec les clients (e.g. Hot Line, réclamations, ...)

Text Retrieval Conferences-TREC 2012 -

<http://trec.nist.gov/tracks.html>

- **Contextual Suggestion Track**
The Contextual Suggestion track investigates search techniques for complex information needs that are highly dependent on context and user interests.
- **Crowdsourcing Track**
The Crowdsourcing track will investigate emerging crowd-based methods for search evaluation and/or developing hybrid automation+crowd search systems.
- **Knowledge Base Acceleration Track**
This track looks to develop techniques to dramatically improve the efficiency of (human) knowledge base curators by having the system suggest modifications/extensions to the KB based on its monitoring of the data streams.
- **Legal Track**
The goal of the legal track is to develop search technology that meets the needs of lawyers to engage in effective discovery in digital document collections.
- **Medical Records Track**
The goal of the Medical Records track is to foster research on providing content-based access to the free-text fields of electronic medical records.
- **Microblog Track**
The Microblog track examines search tasks and evaluation methodologies for information seeking behaviors in microblogging environments.
- **Session Track**
The Session track aims to provide the necessary resources in the form of test collections to simulate user interaction and help evaluate the utility of an IR system over a sequence of queries and user interactions, rather than for a single "one-shot" query.
- **Web Track**
The Web track explores Web-specific retrieval tasks, including diversity and efficiency tasks, over collections of up to one billion Web pages.

Past tracks

- **Cross-Language Track** investigates the ability of retrieval systems to find documents that pertain to a topic regardless of the language in which the document is written.
- **Filtering Track** the user's information need is stable (and some relevant documents are known) but there is a stream of new documents. For each document, the system must make a binary decision as to whether the document should be retrieved
- **Interactive Track** studying user interaction with text retrieval systems. studies with real users using a common collection and set of user queries.
- **Novelty Track** investigate systems' abilities to locate new (i.e., non-redundant) information.
- **Robust Retrieval Track** includes a traditional ad hoc retrieval task task, but with the focus on individual topic effectiveness rather than average effectiveness.
- **Video Track** research in automatic segmentation, indexing, and content-based retrieval of digital video. The track became an independent evaluation (TRECVID).
- **Web Track** search tasks on a document set that is a snapshot of the World Wide Web. Last ran in TREC 2004.

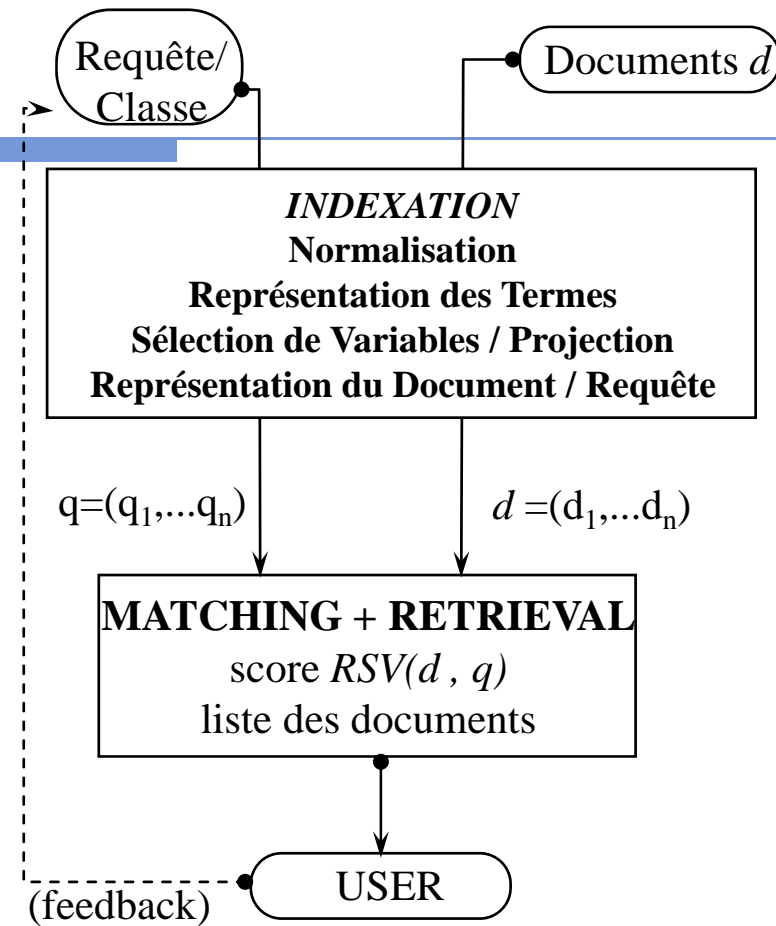
Recherche d'information textuelle

Problèmes de base abordés

- ☐ Représentation des documents, dictionnaires, index
 - Prétraitements
 - Indexation
 - Compression
- ☐ Modèles de recherche d'information
- ☐ Evaluation

Schéma général d'un système de RI classique

- Processus
 - 3 étapes principales
- Modèles
 - hypothèses : Sac de mots, Indépendance des termes
 - Logique
 - Vectoriel
 - Probabiliste
 - Langage
 - Réseaux bayesiens
 - etc



RD : notions de base

- Requête : expression en texte "libre" formulée par l'utilisateur
 - e.g. "text mining", "je voudrais trouver des documents qui parlent de ...", paragraphes entiers, .
- Document : texte, abstract, passage de texte, texte + structure (e.g. balises HTML : titres, paragraphes, ...)...
- Corpus : ensemble de documents textuels (statique ou dynamique), éventuellement liens entre documents. Taille : 10^6 , 10^9 , ...
- Catégorie : liste de mots clé



Deux lois de base caractérisant corpus de textes et documents

Lois de puissance

□ Loi de Zipf

- Caractérise la fréquence d'occurrence en fonction du rang
- Empiriquement : $\text{fréquence} \cdot \text{rang} = \text{cte}$
- Le 1^{er} mot est environ 2 fois plus fréquent que le 2nd qui est 2 fois plus fréquent que le 3^e etc
- Brown Corpus (> 1 M mots)

Mot	Rang	Fréquence	%
the	1	69971	7%
of	2	36411	3.5 %
and	3	28852	2.6%

- Implications
 - Quelques mots communs représentent la plus grande partie des textes (stopwords)

Expression formelle :

$$f(r, s, N) = \frac{1/r^s}{\sum_{n=1}^N 1/n^s}$$

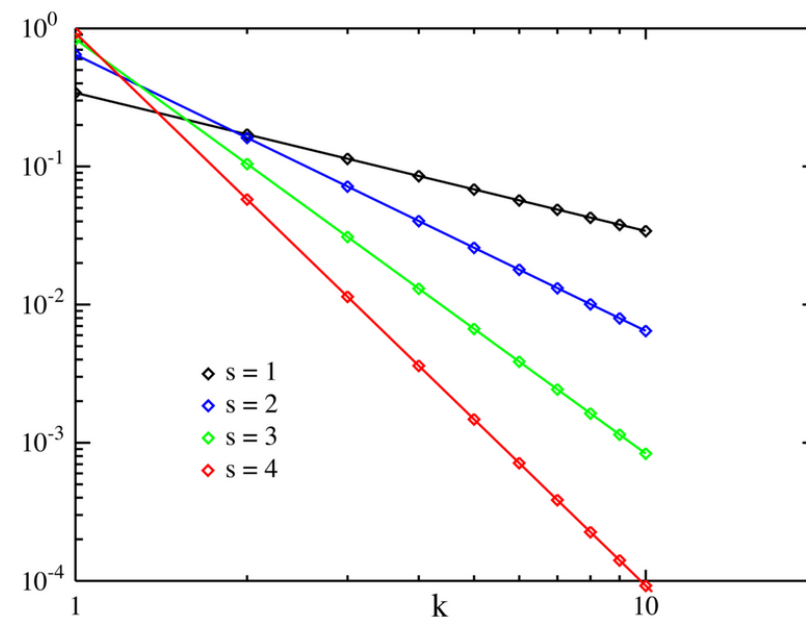
$$\log(f) = -\log r^s - \log \sum_{n=1}^N 1/n^s$$

r : rang

N : taille du corpus

s : paramètre qui dépend du corpus

En anglais $s \approx 1$, i.e. $f.r \approx 1$



$N = 10$, log fréquence vs log rang
(Wikipedia)

-
- Autres phénomènes suivant une loi de puissance à la Zipf (Fréquence vs rang)
 - Fréquence d'accès des pages web
 - Population des villes
 - Trafic internet par site
 - Noms dans une population
 - etc

Loi de Heaps

- Caractérise le nombre de mots distincts dans un document

$$V = Kn^\beta$$

V : taille du vocabulaire

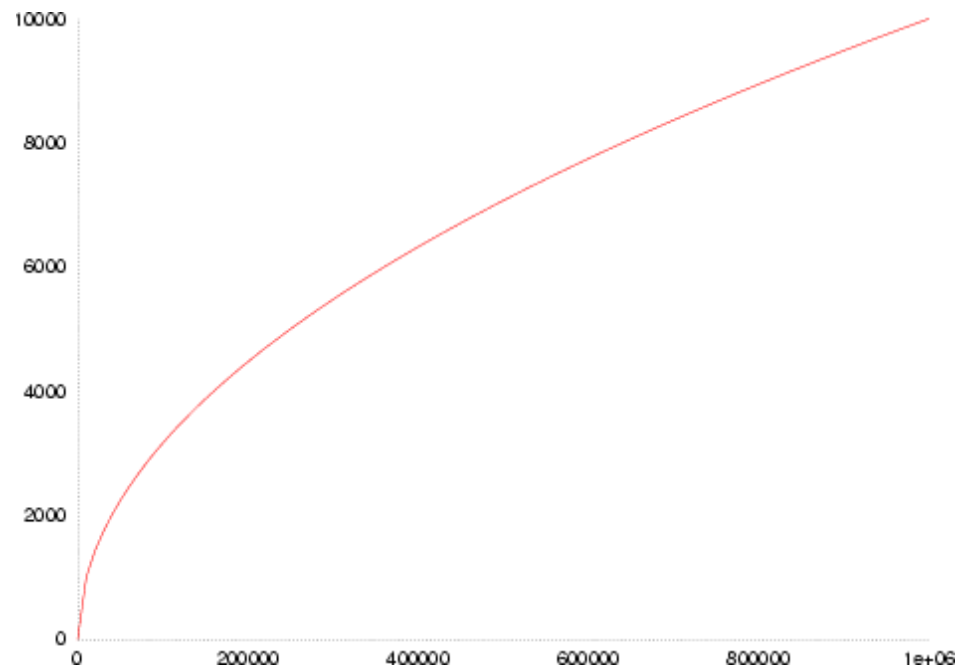
n : taille du texte

K, β paramètres dépendant du texte


Anglais

K entre 10 et 100 et β entre 0.4 et 0.6

Croissance sous linéaire du vocabulaire en fonction de la taille du texte ou du corpus

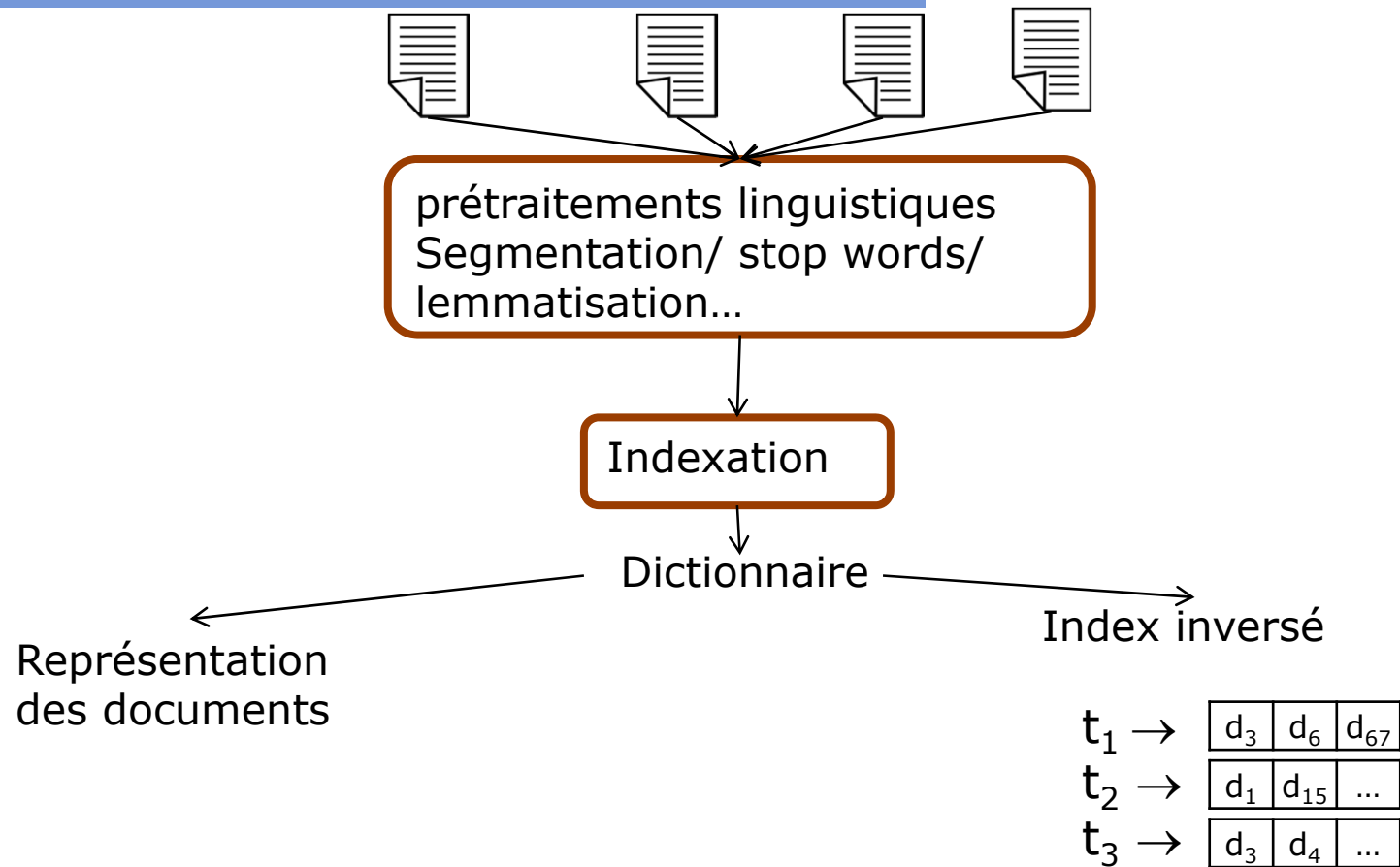


V en fonction de n
(Wikipedia)



Prétraitement et représentation des textes

Chaine d'indexation



Prétraitement et représentation des textes - segmentation

- Analyse lexicale (segmentation – tokenisation)
- Conversion du texte en un ensemble de termes
 - Unité lexicale ou radical
 - Espaces, chiffres, ponctuations, etc
 - Dépend de la spécificité des langues traitées
 - e.g. langues asiatiques (pas de signe de séparation) vs indo-européennes
 - Même pour les langues d'une même famille, nombreuses spécificités
 - e.g. aujourd'hui constitue un seul mot
 - Arbeiterunfallversicherungsgesetz (33 lettres)
Loi sur l'assurance des accidents du travail
 - Logiciel « TreeTagger » pour les langues indo-européennes

Prétraitement et représentation des textes

- Stopwords

- Quelles unités conserver pour l'indexation ?
 - Stop words - anti-dictionnaire
 - Les mots les plus fréquents de la langue "stop words" n'apportent pas d'information utile e.g. prépositions, pronoms, mots « athématiques »,.. (peut représenter jusqu'à 30 ou 50% d'un texte)
 - Ces "stop words" peuvent être dépendants d'un domaine ou pas L'ensemble des mots éliminés est conservé dans un anti-dictionnaire (e.g. 500 mots).
 - Les mots les plus fréquents ou les plus rares dans un corpus (frequency cut-off)
 - Les connaissances sémantiques permettent également d'éliminer des mots
 - Techniques de sélection de caractéristiques

Stoplist - exemple

□ a
about
above
accordingly
after
again
against
ah
all
also
although
always
am
an
and
and/or
any
anymore
anyone
are
as
at
away

□ b
be
been
begin
beginning
beginnings
begins
begone
begun
being
below
between
but
by

□ was
we
were
what
whatever
when
where
which
while
who
whom
whomeve
r
whose
why
with
within
without
would
yes
your
yours
yourself
yourself
s

Prétraitement et représentation des textes

Normalisation

- Normalisation
 - Objectif obtenir une forme canonique pour les différents mots d'une même famille
- Normalisation de la forme textuelle
 - Accents, casse, ponctuations, symboles spéciaux, dates
- Stemming
 - Utilisation d'une forme canonique pour représenter les variantes morphologiques d'un mot
 - e.g. dynamic, dynamics, dynamically, ...seront représentés par un même terme dynamic, naviguer, naviguant, navireidem
 - Augmente le rappel, peut diminuer la précision
 - Techniques (exemples) :
 - systèmes itératifs à base de règles simples (e.g. pour l'anglais Porter stemming -largement employé) : on établit une liste de suffixes et de préfixes qui sont éliminés itérativement.
 - méthodes à base de dictionnaires mot - forme canonique. Intérêt : langue présentant une forte diversité lexicale (e.g. français)
- Lemmatisation
 - Analyse linguistique permettant de retrouver la forme présente dans les dictionnaires
 - E.g. infinitifs pour les verbes, le singulier pour les noms etc
- Regroupement
 - de mots similaires au sens d'un critère numérique

Porter Stemming

- Largement utilisé en anglais
 - 5 phases de réduction des mots appliquées séquentiellement
 - Règles de re-écriture avec priorité d'application
 - Exemple (from Manning et al. 2008)
 - On utilise d'abord la règle qui s'applique sur le plus long suffixe
- | | |
|---------------------------|---------------------------------|
| ■ <i>sses</i> → <i>ss</i> | <i>Caresses</i> → <i>caress</i> |
| ■ <i>ies</i> → <i>i</i> | <i>ponies</i> → <i>poni</i> |
| ■ <i>ss</i> → <i>ss</i> | <i>caress</i> → <i>caress</i> |
| ■ <i>s</i> → | <i>cats</i> → <i>cat</i> |

Prétraitement et représentation des textes

- La pondération des termes
 - Mesure l'importance d'un terme dans un document
 - Comment représenter au mieux le contenu d'un document ?
 - Considérations statistiques, parfois linguistiques
 - Loi de Zipf : élimination des termes trop fréquents ou trop rares
 - Facteurs de pondération
 - E.g. tf (pondération locale), idf (pondération globale)
 - Normalisation : prise en compte de la longueur des documents, etc

Prétraitement et représentation des textes

□ Représentations :

- booléenne : existence des termes (fréquent en catégorisation)
- réelle : fréquence des termes, locale (pr à un texte), globale (pr à un ens de textes), relative à la longueur du texte.
- Sélection de caractéristiques
- Projections : réduction supplémentaire (SVD, ACP, NMF, Word2vec...)

Modèles d'indexation

Indexation

- Technique la plus fréquente : **index inversé**
 - chaque terme de l'index est décrit par le numéro de référence de tous les documents qui contiennent ce terme et la position dans ce document du terme.
 - Permet une accélération considérable de la recherche pour une requête. Cet index peut être ordonné en fonction décroissante de la fréquence des termes.
 - Implémentation : différentes structures de données
 - tries (stockage des chaînes de caractère dans des arbres) – retrouve une chaîne de caractère en temps proportionnel à sa longueur
 - Table de hashage, etc

Construction des index inversés

- ❑ L'index est construit séquentiellement en lisant les documents
- ❑ L'index est trié par termes et documents
- ❑ Différentes informations peuvent être associées aux mots du dictionnaire
 - Ici doc frequency
- ❑ Les documents sont arrangés en une liste triée suivant doc_id

Construction des index inversés



Doc 1

I did enact Julius Caesar: I was killed
i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus
hath told you Caesar was ambitious:

Figure from Manning et al. 2008

term	docID	term	docID	term	doc. freq.	→	postings lists
I	1	ambitious	2	ambitious	1	→	2
did	1	be	2	be	1	→	2
enact	1	brutus	1	brutus	2	→	1 → 2
julius	1	brutus	2	capitol	1	→	1
caesar	1	capitol	1	caesar	1	→	1
I	1	caesar	1	caesar	2	→	1 → 2
was	1	caesar	2	did	1	→	1
killed	1	caesar	2	enact	1	→	1
i'	1	did	1	hath	1	→	2
the	1	enact	1	I	1	→	1
capitol	1	hath	1	i'	1	→	1
brutus	1	I	1	it	1	→	2
killed	1	I	1	julius	1	→	1
me	1	i'	1	killed	1	→	1
so	2	it	2	let	1	→	2
let	2	julius	1	me	1	→	1
it	2	killed	1	noble	1	→	2
be	2	killed	1	so	1	→	2
with	2	let	2	the	2	→	1 → 2
caesar	2	me	1	told	1	→	2
the	2	noble	2	you	1	→	2
noble	2	so	2	was	2	→	1 → 2
brutus	2	the	1	with	1	→	2
hath	2	the	2				
told	2	told	2				
you	2	you	2				
caesar	2	was	1				
was	2	was	2				
ambitious	2	with	2				

Construction des index inversés

□ Remarques

- Les index sont optimisés pour permettre différentes opérations
 - e.g. requêtes booléennes
 - $Q = \text{Paris} \ \& \ \text{Hilton}$ nécessite un merge des deux listes associées à chacun des termes
- Le stockage efficace des index et des listes de documents est fondamental
 - Pour les petits corpus tout peut être en mémoire ou index en mémoire et documents sur disques
 - Pour les grands corpus il faut des solutions distribuées
- Pour de nombreuses applications (web), il est important d'avoir des index dynamiques

Construction des index inversés

- Lors de la construction de l'index, les documents sont lus séquentiellement
- La phase de tri (par terme et doc_id) cf schéma ne peut pas être réalisée en mémoire pour les gros corpus.
 - On est obligé de passer par du tri sur disque (tri externe)
- Par la suite on présente le principe de deux algorithmes de tri
 - - tri externe (BSBI)
 - - tri distribué (Map Reduce)

Blocked sort-based Indexing

- Algorithme
 - Objectif : trier les paires (term_id, doc_id) suivant les clés term_id puis doc_id pour produire un index inversé
 - Le corpus est partitionné
 - Chaque élément de la partition est analysé doc par doc pour produire un index inversé avec cet ensemble de termes x documents
 - cf les 2 étapes sur le schéma : liste (term_id, doc_id) par doc puis tri de cette liste suivant les 2 clés
 - l'index partiel est écrit sur disque
 - Quand tout le corpus a été analysé on fusionne les index partiels
- Complexité
 - $O(T \log T)$ avec T nombre de paires (term_id, doc_id)
- Exemple (chiffres)

□ Étape merge

- On ouvre tous les blocs simultanément
- On utilise un buffer read de petite taille par bloc et un buffer write pour le résultat final
- On fait le merge en écrivant le plus petit term_id pas encore écrit

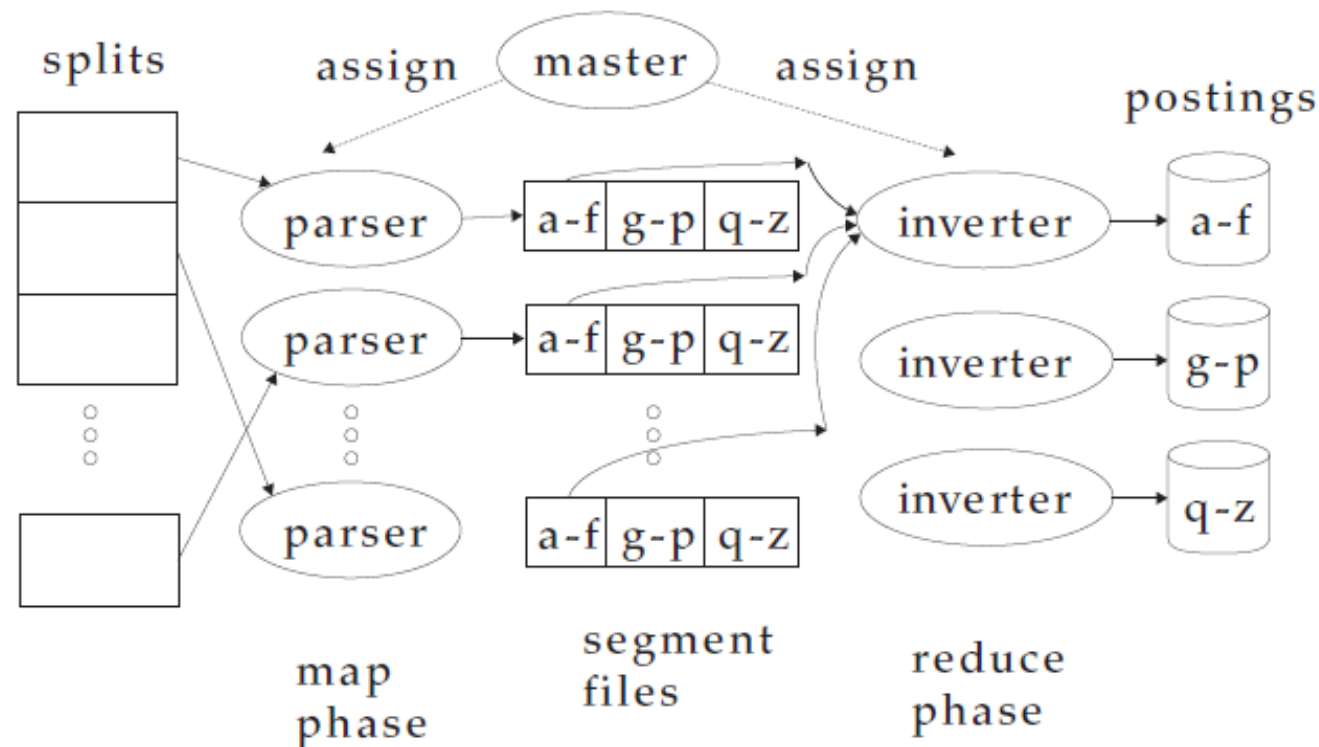
Indexation distribuée (principe)

- Dans le cas de grands corpus, l'indexation est distribuée sur différentes machines
 - e.g indexation du web
 - La distribution peut se faire par termes ou documents
 - Dans l'exemple on considère un partitionnement par terme
 - L'algorithme décrit est une instanciation du schéma MapReduce
 - Un nœud maître distribue les tâches à un ensemble de nœuds

Indexation distribuée (principe)

- On considère pour simplifier que les termes et les documents sont fournis sous la forme `term_id`, `doc_id`
 - Le passage `term` -> `term_id` est fait lors de l'étape de « parsing » des documents
- Les documents sont repartis en « paquets », chaque paquet sera assigné à une machine
 - Typiquement un paquet aura une taille de 16 ou 64 MB (pas trop gros, pas trop petit)
- Phase Map
 - Lecture par chaque machine des documents d'un paquet et création des listes locales à chaque paquet (noté « parser » sur la fig.) le parse d'un paquet est fait par une machine).
 - Celles-ci sont écrites sur des segments de fichiers ([a-f][g-p][q-z]) sur la figure
- Phase Reduce
 - Création des listes globales de documents pour chaque terme, (noté « inverser » sur la figure). La liste inversée pour un paquet est faite par une machine.

Indexation distribuée (principe) – Fig. from Manning et al. 2008



► **Figure 4.5** An example of distributed indexing with MapReduce. Adapted from Dean and Ghemawat (2004).

Modèles de recherche

Modèle

vectoriel

probabiliste

de langue

Modèles de recherche

☐ *hypothèse de base*

- Plus la requête et le document ont de mots en commun, plus grande sera la pertinence du document
- Plus la requête et le document ont une distribution de termes similaire, plus grande sera la pertinence du document

Les classiques

- *modèle booléen*
 - Modèle pionnier
 - recherche de documents s'appariant de façon exacte avec la requête . Requête = expression logique ET..OU..NON.
 - Transparent pour l'utilisateur, rapide (web)
 - Rigide, non robuste, pas de pondération de l'importance des termes,..
- *modèle vectoriel*
 - recherche de documents possédant un fort degré de similarité avec la requête
 - Permet d'ordonner les documents
 - Expression du besoin : requête en langage naturel
 - Rq : sur le web, la requête moyenne est de 2,5 mots clé !
- *modèle probabiliste*
 - probabilité qu'un document soit pertinent pour la requête
 - Qualités : idem modèle vectoriel



Modèle vectoriel

Modèle vectoriel

- Espace de caractéristiques t_i , $i = 1 \dots n$ i.e. termes sélectionnés pré-traités
- Représentation des documents - requêtes : vecteur de poids dans l'espace des caractéristiques
document: $d = (x_0, \dots, x_{n-1})$
requête: $q = (y_0, \dots, y_{n-1})$
- x_k poids de la caractéristique k dans le document d , e.g.
 - ☐ présence-absence,
 - ☐ fréquence du terme dans le document, dans la collection (cf. idf)
 - ☐ importance du terme pour la recherche
 - ☐ facteurs de normalisation (longueur du document)
- Les mots sont supposés indépendants

Modèle vectoriel (2)

- Avantages par rapport au modèle booléen
 - les documents sont évalués sur une échelle continue
 - l'importance des termes est pondérée
 - permet de traiter du texte libre

- Inconvénients
 - hypothèse d'indépendance des termes
 - initialement conçu pour des documents courts, pour des documents longs, facteurs de normalisation, approches hiérarchiques par paragraphes (sélection de paragraphes pertinents + combinaison des scores des paragraphes)

Une méthode de référence tf-idf

☐ Term frequency

- $tf(t_i, d)$: # occurrences de t_i dans le document d

☐ Inverse document frequency (idf)

- $df(t_i)$: # documents contenant t_i
- $idf(t_i)$: fréquence inverse
- idf décroît vers 0 si t_i apparaît dans tous les documents
- N nbre docs dans le corpus

$$idf(\varphi_i) = \log \left(\frac{1 + N}{1 + df(t_i)} \right)$$

$$x_i = tf(t_i, d)idf(t_i)$$

☐ Codage tf-idf

☐ Remarques

- Il existe de nombreuses variantes de ces poids
- tf varie avec la taille des documents
 - ☐ e.g si on double la taille des documents, tf double, le document sera considéré plus pertinent

- Variantes du tf

- Elles visent à réduire la variabilité du tf
- Pondération fréquentielle normalisée

- $\lambda + (1 - \lambda) \frac{tf(t,d)}{tf_{max}(d)}$

- Avec λ terme de lissage et $tf_{max}(d)$ fréquence maximale dans d

- Pondération logarithmique

- $$\begin{cases} 1 + \log(tf(t,d)) & \text{si } tf(t,d) > 0 \\ 0 & \text{sinon} \end{cases}$$

- Limite la croissance du tf et la variabilité des valeurs prises

Modèle vectoriel (Salton 1968)

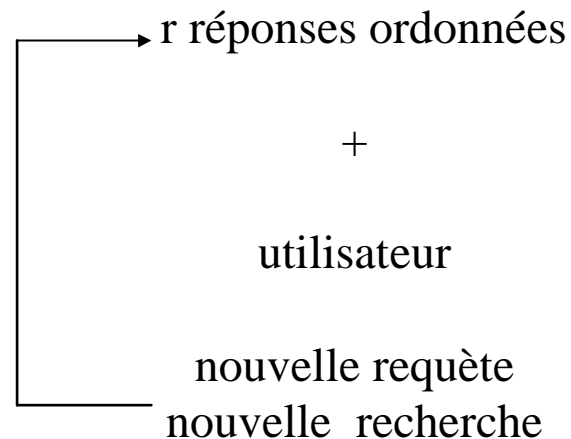
- Différentes fonctions de score peuvent être employées avec un codage fréquentiel des documents
- La plus répandue
 - Codage tf-idf (présent dans l'index inversé)
 - Score : cosinus entre les vecteurs du document et de la requête

$$\square \quad s(d, q) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}}$$

- Avec une similarité produit scalaire
 - $x_i = \begin{cases} 1 + \log(tf(t_i, d)) & \text{si } tf(t, d) > 0 \\ 0 & \text{sinon} \end{cases}$ et $y_i = \begin{cases} idf(t_i) & \text{si } t \in q \\ 0 & \text{sinon} \end{cases}$

Recherche interactive

□ Méthode classique : relevance feedback



- relevance : v. a. dans $\{0, 1\}$
- idée : utilisateur examine une partie des meilleurs documents et les étiquette 1/0
- la requête est reformulée (enrichissement)

Recherche interactive

- Liste ordonnée des r meilleurs documents

$$D_r(q) = \{d_1, d_2, \dots, d_r\}$$

- Partition de ces r documents (ou d'une partie) par l'utilisateur

$$D_r(q) = D_r^{rel}(q) \cup D_r^{nonrel}(q)$$

- Principe du relevance feedback

$$q' = f(q, D_r^{rel}, D_r^{nonrel})$$

Recherche interactive-Exemple

- Query expansion : reestimation des poids de la requête - Rocchio 1971 (heuristique)
 - réestimation de la requête :
 - $$y' = \alpha \frac{y}{\|y\|} + \frac{\beta}{|D_r^{rel}|} \sum_{d_j \in D_r^{rel}} \frac{x^j}{\|x^j\|} - \frac{\gamma}{|D_r^{nonrel}|} \sum_{d_j \in D_r^{nonrel}} \frac{x^j}{\|x^j\|}$$
 - Avec y^j codage tf-idf du document d^j
 - Variante : $D_r^{nonrel} = \emptyset$
 - améliorations allant de 20% à 80 % par rapport à sans RF.
 - Différentes variantes :
 - considérer
 - optimiser α et β
 - optimiser le nombre de documents du feedback ...

□ Automatic query expansion

- pas de feedback utilisateur, les k premiers documents sont considérés comme pertinents
- Marche mieux quand la distribution de D_r^{rel} est unimodale, cas multimodal risque de disparition des modes non principaux
- Le système va fournir des documents similaires à ceux déjà trouvés ...



Modèle probabiliste

Modèle probabiliste

□ Hypothèses

- Une paire (d,q) est la réalisation d'un tirage aléatoire dans un espace document \times requête
- À chaque paire on associe une variable aléatoire binaire R qui vaut 1 si d est pertinent et 0 sinon

□ Probability Ranking Principle (Robertson 77)

- Présenter les documents à l'utilisateur selon l'ordre décroissant de leur probabilité de pertinence $P(R=1|d,q)$
- Propriété
 - Ce principe est optimal dans le sens où il optimise le risque de Bayes pour la règle de décision suivante :
 - D est pertinent ssi $P(R = 1|d, q) > P(R = 0|d, q)$

Binary independence model

- C'est le modèle de base classique associé à ce principe
 - (1) d et q sont représentés comme des vecteurs binaires présence/ absence de termes
 - $d = (x_1, \dots, x_n), q = (y_1, \dots, y_n)$ avec $x_i, y_i \in \{0,1\}$
 - (2) Les termes dans les documents et les requêtes sont indépendants (sac de mots)
 - (3) Les termes non présents dans la requête sont uniformément répartis dans l'ensemble des documents pertinents et non pertinents

-
- On va utiliser comme score le rapport des probabilités a posteriori

- $$o(d, q) = \frac{P(R=1|d, q)}{P(R=0|d, q)} = \frac{P(d|R=1, q)}{P(d|R=0, q)} \cdot \frac{P(R=1|q)}{P(R=0|q)}$$

- Obtenu avec la règle de Bayes $P(R|d, q) = \frac{P(d|R, q)P(R|q)}{p(d|q)}$

- En utilisant l'hypothèse d'indépendance des termes

- $$o(d, q) = \prod_{i: x_i=1} \frac{P(x_i=1|R=1, q)}{P(x_i=1|R=0, q)} \prod_{i: x_i=0} \frac{P(x_i=0|R=1, q)}{P(x_i=0|R=0, q)} \cdot \frac{P(R=1|q)}{P(R=0|q)}$$

- Notons

- $p_i = P(x_i = 1|R = 1, q)$ la probabilité que le terme x_i apparaisse dans un doc. pertinent pour q
 - $u_i = P(x_i = 1|R = 0, q)$ la probabilité que le terme x_i apparaisse dans un doc. non pertinent pour q

- $o(d, q) = \prod_{i:x_i=1} \frac{p_i}{u_i} \prod_{i:x_i=0} \frac{1-p_i}{1-u_i} \cdot \frac{P(R=1|q)}{P(R=0|q)}$

- Sous l'hypothèse (3)

- $o(d, q) = \prod_{i:x_i=y_i=1} \frac{p_i}{u_i} \prod_{i:x_i=0,y_i=1} \frac{1-p_i}{1-u_i} \cdot \frac{P(R=1|q)}{P(R=0|q)}$

- $o(d, q) = \prod_{i:x_i=y_i=1} \frac{p_i}{u_i} \cdot \frac{1-u_i}{1-p_i} \prod_{i:y_i=1} \frac{1-p_i}{1-u_i} \cdot \frac{P(R=1|q)}{P(R=0|q)}$

- Pour une requête donnée

- $\frac{P(R=1|q)}{P(R=0|q)}$ et $\prod_{i:y_i=1} \frac{1-p_i}{1-u_i}$ sont des constantes

- (ne dépendent que de la requête et pas du document)

- On va utiliser comme score de pertinence

- $s(d, q) = \sum_{i:x_i=y_i=1} \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$

- Estimation des probabilités a posteriori p et u
 - Maximum de vraisemblance sur une base d'apprentissage (i.e. on calcule les fréquences relatives des différents événements)
 - Tableau des fréquences de documents

document	pertinent	Non pertinent	total
$x_i = 1$	a	$df_i - a$	df_i
$x_i = 0$	$A - a$	$(N - df_i) - (A - a)$	$(N - df_i)$
total	A	$N - A$	N

- Avec ces fréquences : $p_i = \frac{a}{A}$, $u_i = \frac{df_i - a}{N - A}$
- En pratique, on utilise un smoothing pour éviter les 0
 - e.g. on ajoute un facteur α à chaque terme de fréquence $(a + \alpha)$, etc

Modèle probabiliste

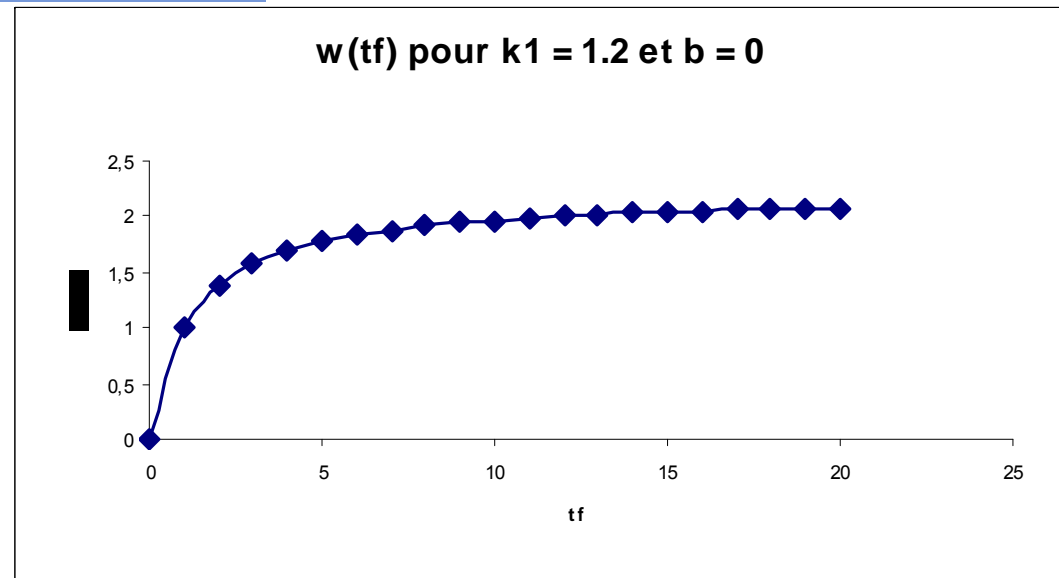
- Nombreuses variantes / extensions“ Problèmes
 - longueur des documents (hypothèse implicite d'égale longueur)
 - expansion des requêtes
 - # doc pertinents considérés (e.g. cas recherche on line <> off line)
 - cooccurrence de termes, prise en compte de « phrases » ...

Okapi - Un système « probabiliste » (Robertson et al.)

Term Frequency

$$w(tf_t) = \frac{tf_t(k_1 + 1)}{K + tf_t}$$

$$K = k_1 * ((1 - b) + b(DL / AVDL))$$



Prise en compte de la longueur des documents

DL : longueur du document

AVDL : longueur moyenne des docs.

k1 et b constantes e.g. k1 = 1.2, b = 0.75

Okapi (2)

□ Inverse Document Frequency

- Pas d'information de pertinence sur les documents

$$idf = \log \frac{N}{n_t}$$

- Information de pertinence sur les documents

$$idf = \log \frac{(r_t + 0.5)(N - n_t - R + r_t + 0.5)}{(R - r_t + 0.5)(n_t - r_t + 0.5)}$$

	Relevant	Non Relevant	total
Contient le terme t	r	n - r	n
Ne contient pas le terme t	R - r	N - n - R + r	N - n
total	R	N - R	N

Okapi (3)

- Score du document d pour la requête q

$$Score\ Okapi = \sum_{t \in q} \frac{tf_{t,d}(k_1 + 1)}{K + tf_{t,d}} * "idf(t)"$$

- Automatic RF

- Sélectionner comme pertinents les B premier documents renvoyés, tous les autres sont non pertinents
- Calculer des poids pour les termes de ces documents
- Ajouter les poids à la requête pour les x (e.g x = 20) meilleur termes

$$"idf" = \log \frac{(r_t + 0.5)(N - n_t - B + r_t + 0.5)}{(B - r_t + 0.5)(n_t - r_t + 0.5)}$$



Modèles de langue

Modèles de langue (Ponte, Croft, Hiemstra, .. 98-99)

- Variables
 - d : document que l'utilisateur a en tête
 - t_i : i^{eme} terme de la requête
- Considérons une requête q de n termes t_1, t_2, \dots, t_n
 - Les documents seront ordonnés selon la pertinence du document pour la requête : $P(d|q) = \frac{P(q|d)P(d)}{P(q)}$
 - Pour q donnée, l'ordre relatif du score de deux documents ne dépend pas de $P(q)$
 - $P(d)$ est choisie uniforme
 - Score d'un document : $P(t_1, \dots, t_n|d)$
 - Probabilité que la requête ait été formulée en ayant en tête le document d
 - On a alors un modèle statistique par document

Modèles de langue (Ponte, Croft, Hiemstra, .. 98-99)

- Hypothèse d'indépendance des termes de la requête conditionnellement à d
 - $P(q|d) = P(t_1, \dots, t_n|d) = \prod_{i=1}^n P(t_i|d)$
- Remarque
 - Si un terme de la requête est absent du document, le score de d pour q devient 0
 - En pratique on utilise un lissage de cette probabilité
 - Le plus courant est un modèle de mélange multinomial entre la distribution des termes sur le document et la distribution des termes dans la collection :
 - $P(t_i|d) = (1 - \lambda_i)P_{MV}(t_i|d) + \lambda_i P_{MV}(t_i)$
 - Avec $P_{MV}(t_i|d)$ estimateur du MV de la probabilité du terme t_i dans le document et $P_{MV}(t_i)$ estimateur du MV de la probabilité d'apparition du terme t_i dans la collection et λ_i facteur de pondération

Modèles de langue (Ponte, Croft, Hiemstra, .. 98-99)

- Estimateurs de maximum de vraisemblance

- $P_{MV}(t_i|d) = \frac{tf(t_i,d)}{\sum_t tf(t,d)}$ et $P_{MV}(t_i) = \frac{\sum_{d'} tf(t_i,d')}{\sum_{d',t} tf(t,d')}$

- Si $\lambda_i = \lambda$ constante, on obtient le lissage dit de Jelinek-Mercer

- Alternativement, les λ_i peuvent être estimés par algorithme EM en maximisant la vraisemblance des observations (document pertinent, requête associée)

□ Lissage de Dirichlet

- Une alternative fréquente au Lissage Jelinek-Mercer consiste à utiliser l'interpolation :

- $P(t_i|d) = (1 - \lambda_d)P_{MV}(t_i|d) + \lambda_d P_{MV}(t_i)$

avec $\lambda_d = \frac{\mu}{|d| + \mu}$, $|d|$ la taille du document et μ un paramètre dont la valeur est choisie empiriquement

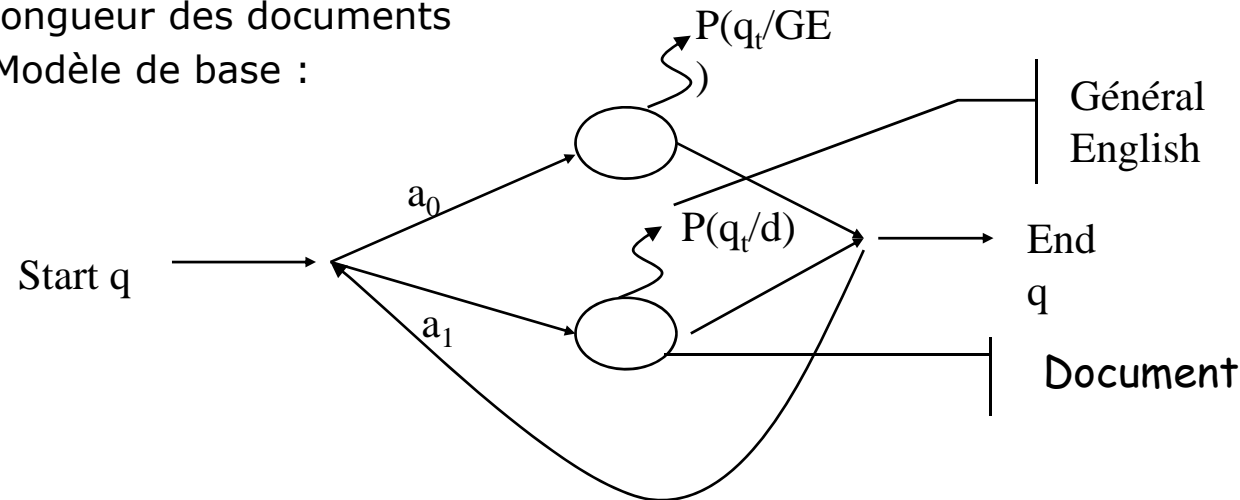
- En substituant, on obtient

- $$P(t_i|d) = \frac{tf(t_i, d) + \mu \frac{\sum_{d'} tf(t_i, d')}{\sum_{d', t} tf(t, d')}}{|d| + \mu}$$

Le modèle de langue comme un HMM

(BBN – Miller et al. 99)

- score : $p(q / R, d)$
 - q et d sont des variables aléatoires
 - q est l'observation, 1 modèle HMM par document
 - TREC 6, 7, ~500 k docs, 50 requêtes
 - Le modèle incorpore naturellement les statistiques sur les termes, la longueur des documents
 - Modèle de base :



$$p(q / d \text{ relevant}) = \prod_t (a_0 p(q_t / GE) + a_1 p(q_t / d))$$

Evaluation en recherche d'information

Evaluation en RI

- Problème difficile, pas de mesure absolue
- Critères de qualité d'un système de RD
 - efficacité de la recherche
 - possibilités de formuler des requêtes riches
 - outils de navigation dans la collection
 - mise à jour et richesse du corpus
- Nombreuses mesures qui donnent des renseignements partiels sur le comportement du système
- Efficacité de la recherche :
 - hyp : on possède un corpus, un ens. De requêtes, des jugements sur les doc. R et $\neg R$ pour une requête.

Evaluation en RI

- L'évaluation quantitative en RI ad-hoc se base sur le paradigme de Cranfield. On dispose d'un
 - Corpus de documents
 - Ensemble de requêtes
 - Souvent les requêtes (mots clés) sont associés à une description plus complète (phrases) du besoin d'information
 - Besoin d'information : « Trouver tous les documents parlant de l'utilisation des réseaux de neurones dans le domaine du traitement du langage naturel »
 - Requête : réseaux + neurones + langage + naturel
 - Ensemble de jugements de pertinence pour chaque paire (document, requête)
 - Ces jugements peuvent être binaires ou être donnés sous forme d'un score, typiquement {0, 1, 2, 3, 4, 5}
 - Ils sont formulés en fonction du besoin d'information

-
- De nombreuses collections de documents ont été développés par la communauté scientifique
 - Cranfield fin des années 50
 - TREC (NIST)
 - CLEF
 - NTCIR (Japon et autres langues asiatiques, cross language evaluation)

Evaluation en IR : mesures de rappel - précision

- Corpus de documents non ordonnés
- Les deux mesures les plus courantes sont la précision et le rappel
 - $precision = \frac{\#documents\ pertinents\ découverts}{\#documents\ découverts}$
 - $= P(pertinent|découvert)$
 - $rappel = \frac{\#documents\ pertinents\ découverts}{\#documents\ pertinents\ dans\ la\ collection}$
 - $= P(découvert|pertinent)$
- Ces deux mesures peuvent être calculées pour chaque requête et moyennées
 - Les deux quantités sont en général antagonistes
 - Suivant les utilisations, on peut vouloir favoriser précision (e.g. web) ou rappel

Evaluation en IR : mesures agrégées

- On utilise souvent également une mesure unique qui est la moyenne harmonique – dite F measure

- $F_1 = \frac{2PR}{P+R}$

- Ou sous une forme plus générale

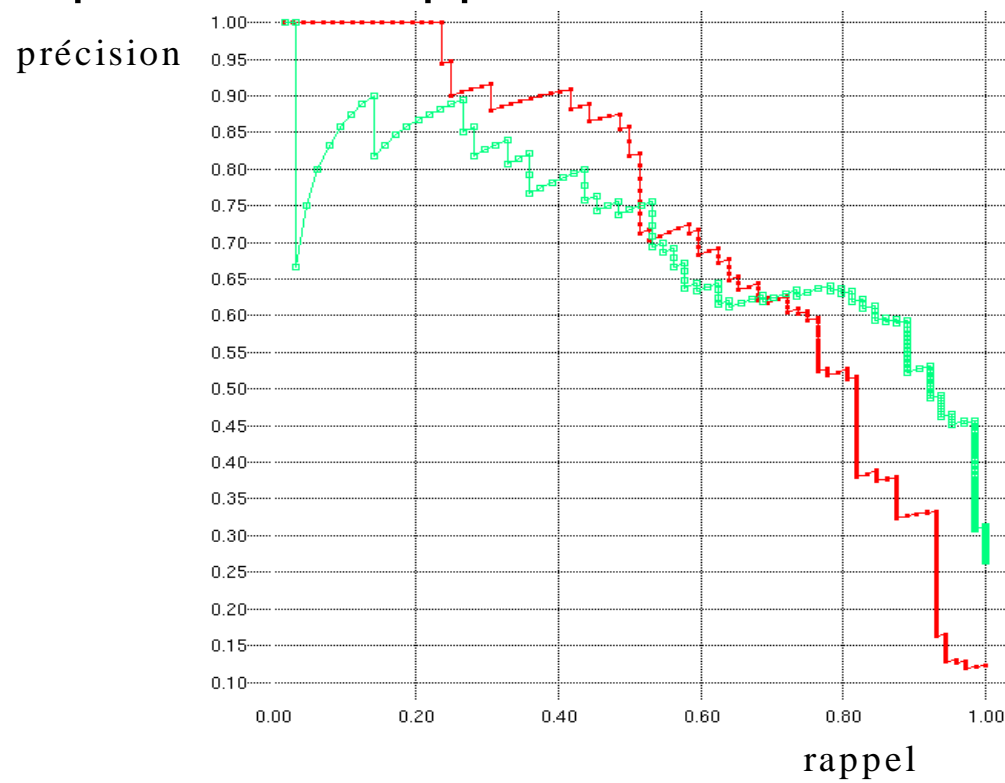
- $F = \frac{(\beta^2+1)PR}{\beta^2P+R}$

- $\beta > 1$ le rappel est favorisé, $\beta < 1$ la précision est favorisée

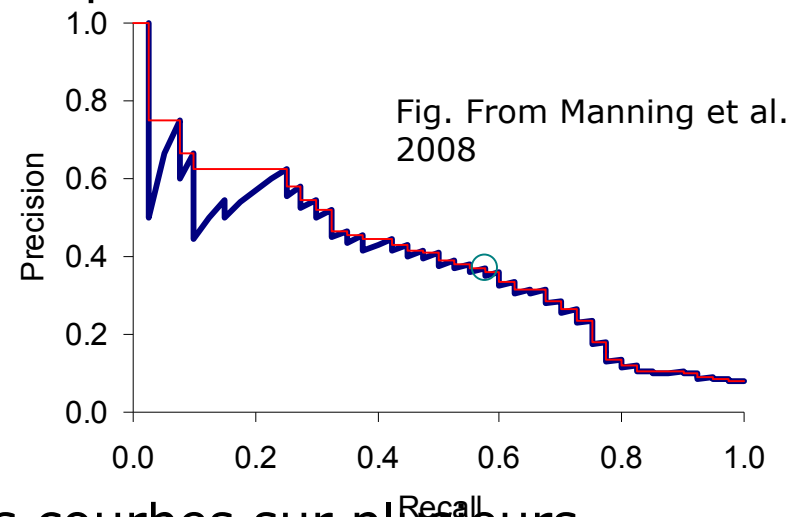
Evaluation en IR : mesures de rappel – précision – résultats ordonnés

- Les moteurs de recherche renvoient en général des listes ordonnées
 - On va adapter les mesures précédentes à ces listes
 - Pour une requête:
 - Precision à k :
 - $P@k = \frac{\text{\#documents pertinents découverts}}{k}$
 - Rappel à k :
 - $R@k = \frac{\text{\#documents pertinents découverts parmi les } k \text{ premiers}}{\text{\#documents pertinents dans la collection}}$
 - On trace ensuite $P(R)$, la courbe précision - rappel
-
- r : nombre de documents inspectés par l'utilisateur parmi les doc. fournis par le système, i.e. les r premiers de la liste
 - Valeurs typiques, 5, 10, 20, 25, 100, 1000

□ Courbe précision-rappel



- Si on veut moyenner sur plusieurs requêtes, on emploie la précision interpolée
- $P_{interp}(R) = \max_{r' \geq R} P(R')$
- On obtient alors une courbe interpolée



- On peut ensuite moyenner ces courbes sur plusieurs requêtes

-
- Moyenne sur plusieurs requêtes (11 points interpolation)

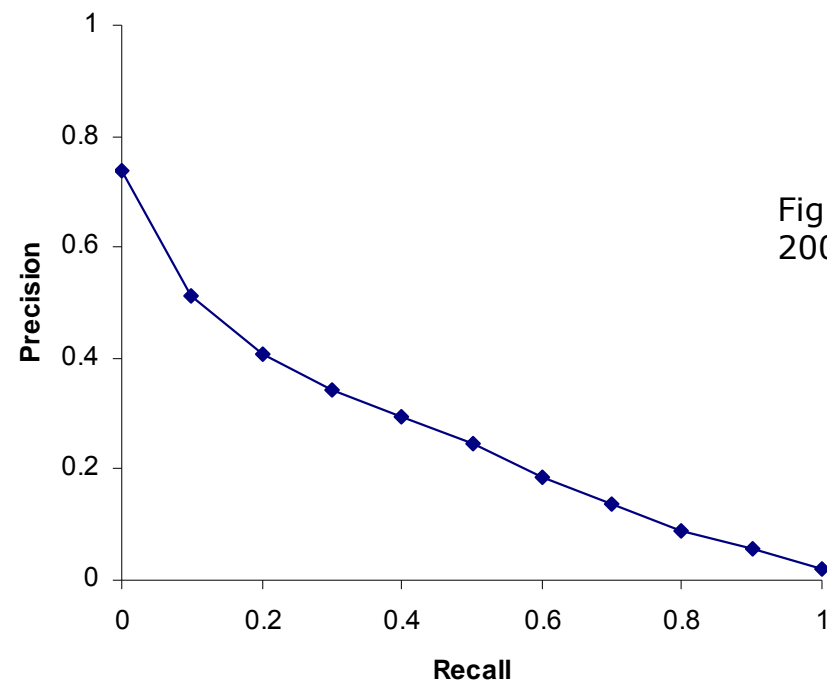


Fig. From Manning et al.
2008

-
- Mesures agrégées
 - Précision moyenne (AveP)
 - On calcule la moyenne arithmétique de la précision prise sur un certain nombre de points de rappels
 - (typiquement 11 points (0, 0.1, ..., 0.9, 1))
 - C'est une approximation de l'aire sous la courbe précision-rappel
 - Mean Average Precision (MAP)
 - Si on a plusieurs requêtes, on moyenne la précision moyenne sur l'ensemble des requêtes

Précision - exemple

+ : pertinent	Liste 1	Liste 2	Liste 3
- Non pertinent	d1 (+)	d4 (-)	d4 (-)
	d2 (+)	d5 (-)	d1 (+)
	d3 (+)	d6 (-)	d2 (+)
	d4 (-)	d1 (+)	d5 (-)
	d5 (-)	d2 (+)	d6 (-)
	d6 (-)	d3 (+)	d3 (+)
p_3	1	0	2/3
p_6	0.5	0.5	0.5
Precision moyenne non interpolée	1	0.38	0.55
Precision moyenne interpolée 11 points	1	0.5	

- Précision moyenne non interpolée
 - Moyenne de la precision pour l'ensemble des docs pertinents de la liste
- Précision moyenne interpolée
 - La précision est calculée à différents niveaux de rappel (0%; 10%, 20%, ...100%)
 - Si la précision remonte après le point de rappel i , on prend la valeur de précision la plus forte rencontrée après le point i (interpolation)

Gain cumulé normalisé

- **Discounted cumulative gain (DCG)**
 - Utilisé dans le cadre de la recherche Web
 - Utilise une information de pertinence graduée (5 niveaux)
 - Mesure le gain d'information apporté par un document en fonction de sa position dans la liste des résultats
 - Pour la RI Web seules les premières informations présentées sont importantes
- Hypothèses
 - Les documents pertinents sont plus utiles quand ils apparaissent à un rang élevé.
 - Les documents très pertinents sont plus utiles que les peu pertinents qui sont plus utiles que les non pertinents.

- Cumulative Gain (CG) (Ancêtre de DCG)

- CG au rang p

$$CG_p = \sum_{i=1}^p rel_i$$

- Où rel_i est la pertinence graduée du doc i
 - Ne tient pas compte de l'ordre des documents

- Discounted Cumulative Gain (DCG)

- Prise en compte de l'ordre des documents par une fonction décroissante du rang

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i)}$$

- Autres formulations possibles

Normalized DCG

- Pour moyenner DCG sur un ensemble de requête, on calcule une version normalisée NDCG
 - On suppose que l'on dispose d'une liste idéale de résultats dont le DCG_p vaut $IDCG_p$

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

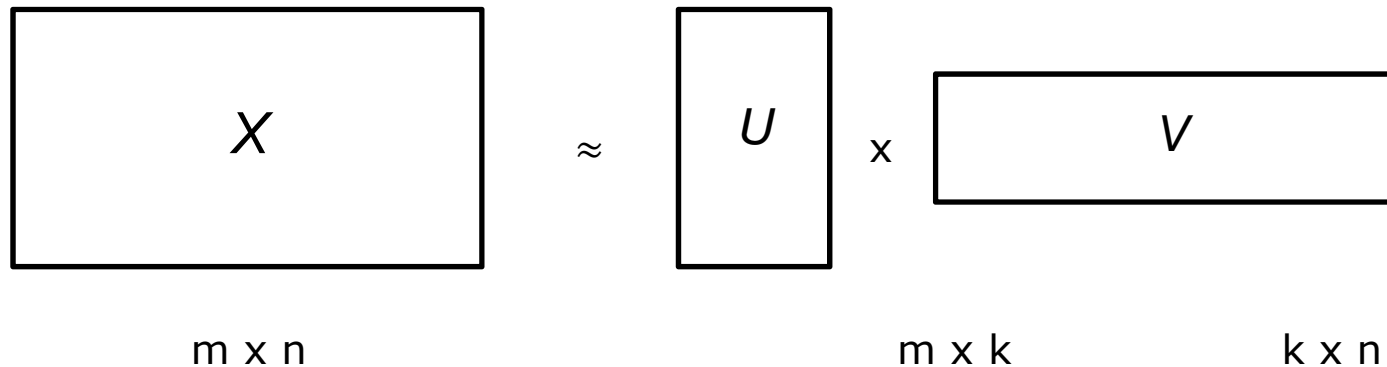
- On moyenne ensuite sur l'ensemble des requêtes
- Il faut bien sûr disposer d'une liste idéale
 - Petites mains du web...

Latent variable models

- Matrix Factorization
- LSI
- Probabilistic Latent Indexing

Matrix factorization

- $X = \{x^1, \dots, x^n\}, x^i \in R^m$
- X is a $m \times n$ matrix with columns the x^i s
- Low rank approximation of X
 - Find factors $U, V, / X \approx UV$
 - With U an $m \times k$ matrix, V a $k \times n$ matrix, $k < m, n$



- Many different decompositions
 - e.g. Singular Value Decomposition, Non Negative Matrix Factorization, Tri factorization, etc

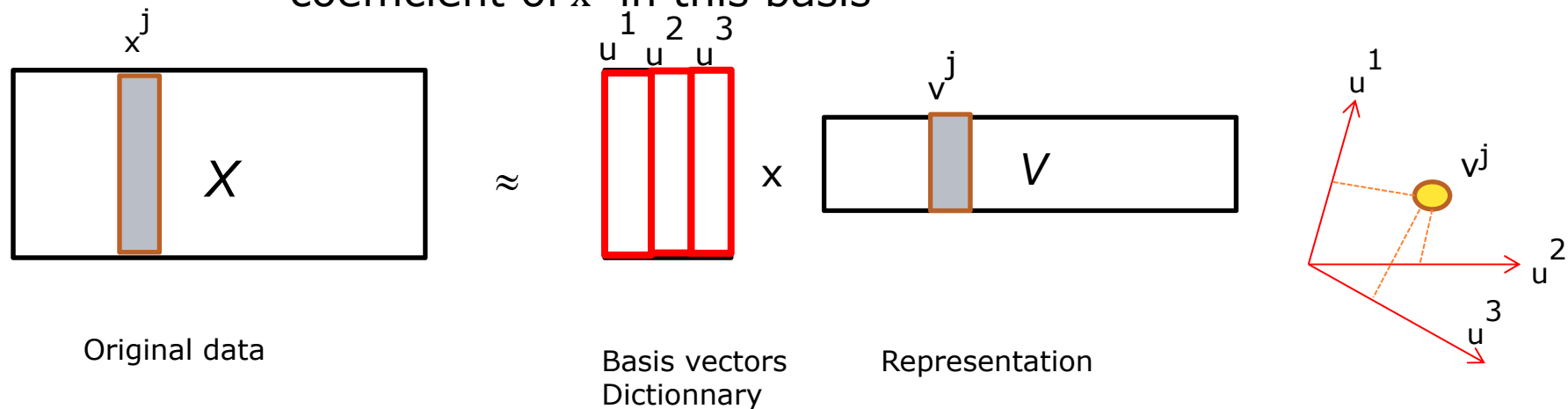
2 views of matrix factorization

- Decomposition in a vector basis

- $X \approx UV$

- $x^j = \sum_{i=1}^k u^i v_i^j$

- Columns of U , u^i are basis vectors, the v_i^j are the coefficient of x^j in this basis



2 views of matrix factorization

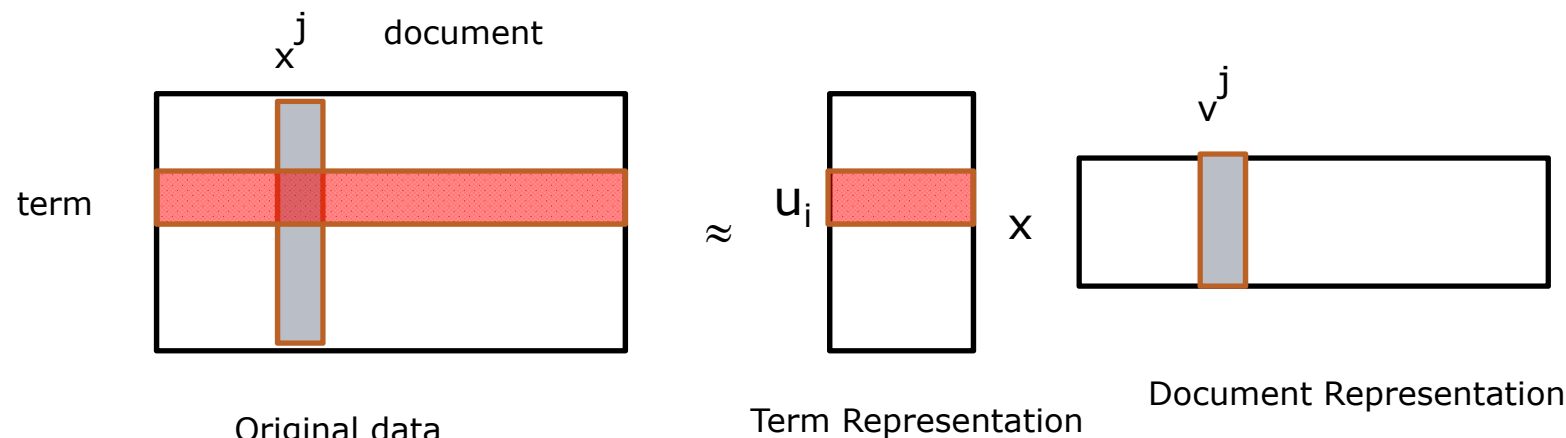
- Sum of rank 1 matrices

- $X = \sum_{i=1}^k u^i v_i$

- Where u^i is the i th column of U and v_i is the i th row of V

□ Interpretation

- If X is a term x document matrix



- Terms and documents are represented in a common representation space of size k
- Their similarity is measured by a dot product in this space

linear algebra review

- X a $m \times n$ matrix
 - The rank of X is the number of linearly independent rows or columns
 - $\text{rank}(X) \leq \min(m, n)$
- X a square $m \times m$ matrix
 - Eigenvector, eigenvalue of X
 - $(x, \lambda) / Xx = \lambda x$
 - The number of non zero eigen values of X is at most $\text{rank}(X)$

-
- Diagonalisation of a real square matrix
 - X a real valued $m \times m$ matrix of rank m
 - $X = U\Lambda U^{-1}$
 - the columns of U are the eigenvectors of X and Λ is a diagonal matrix whose entries are the eigenvalues of X in decreasing order
 - Diagonalisation of a symmetric real valued matrix
 - X a real valued $m \times m$ symmetric matrix of rank m
 - $X = U\Lambda U^T$
 - The columns of U are orthogonal and unit length normalized
 - $U^T = U^{-1}$

Singular value decomposition of a rectangular matrix

- X a $m \times n$ matrix of rank r
- SVD of X
 - $X = U\Sigma V^T$
 - Σ diagonal matrix of singular values of XX^T
 - Singular values are square roots of eigenvalues
 - U matrix of eigenvectors of XX^T
 - V matrix of eigenvectors of $X^T X$
 - If $\text{rank}(X) = r$ then
 - only r eigenvalues are non 0
 - $\text{Image}(X) = \text{span}(u^1, \dots, u^r)$
 - U, V are orthogonal

□ Best rank k approximation

□ Let $k < r$

■ $\min_{B: \text{rank}(B)=k} \|X - B\|^2 = \|X - X_k\|^2$

■ $X_k = \sum_{i=1}^k u^i \sigma_i (v^T)^i = U_k \Sigma_k V_k^T$



Latent Semantic Analysis

LSI

- Motivation

- Tirer parti des cooccurrences des termes dans les documents pour obtenir des représentations de taille plus petite et apporter des solutions aux problèmes de synonymie et polysémie rencontrés dans les modèles vectoriels
- Exemple
 - $q = (\dots, \text{car}, \dots)$, $d = (\dots, \text{automobile}, \dots)$

- Principe

- Projeter requêtes et documents dans un espace de dimension réduite où les termes qui cooccurrent sont « proches »

LSI interprétation

- X is a term x document matrix
 - m terms in rows, n documents in columns
 - x_{ij} could be 0/1 or tf-idf for example
 - U_k rows encode the term projection on the latent factors
 - U is the matrix of eigenvectors of the term cooccurrence matrix
 - V_k^T columns encode the term projection on the latent factors
 - V^T is the matrix of eigenvectors of the document cooccurrence matrix
 - U and V are orthonormal

-
- Représentation d'une requête ou d'un document dans l'espace des termes :
 - $q_k = \Sigma_k^{-1} U_k^T q$
 - Les termes qui co-occurrent fréquemment sont projetés au même « endroit »
 - idem pour la projection des termes dans l'espace des documents avec V
 - Calcul de la similarité : e.g. $RSV_{\cos}(q', d')$
 - Par rapport au modèle vector space
 - Les documents sont représentés dans un espace dense de taille réduite
 - Résout quelques pb de synonymie
 - On perd les facilités d'indexe inversé
 - Coût du calcul de la SVD

Probabilistic Latent Semantic Analysis

Preliminaries : unigram model

- Generative model of a document

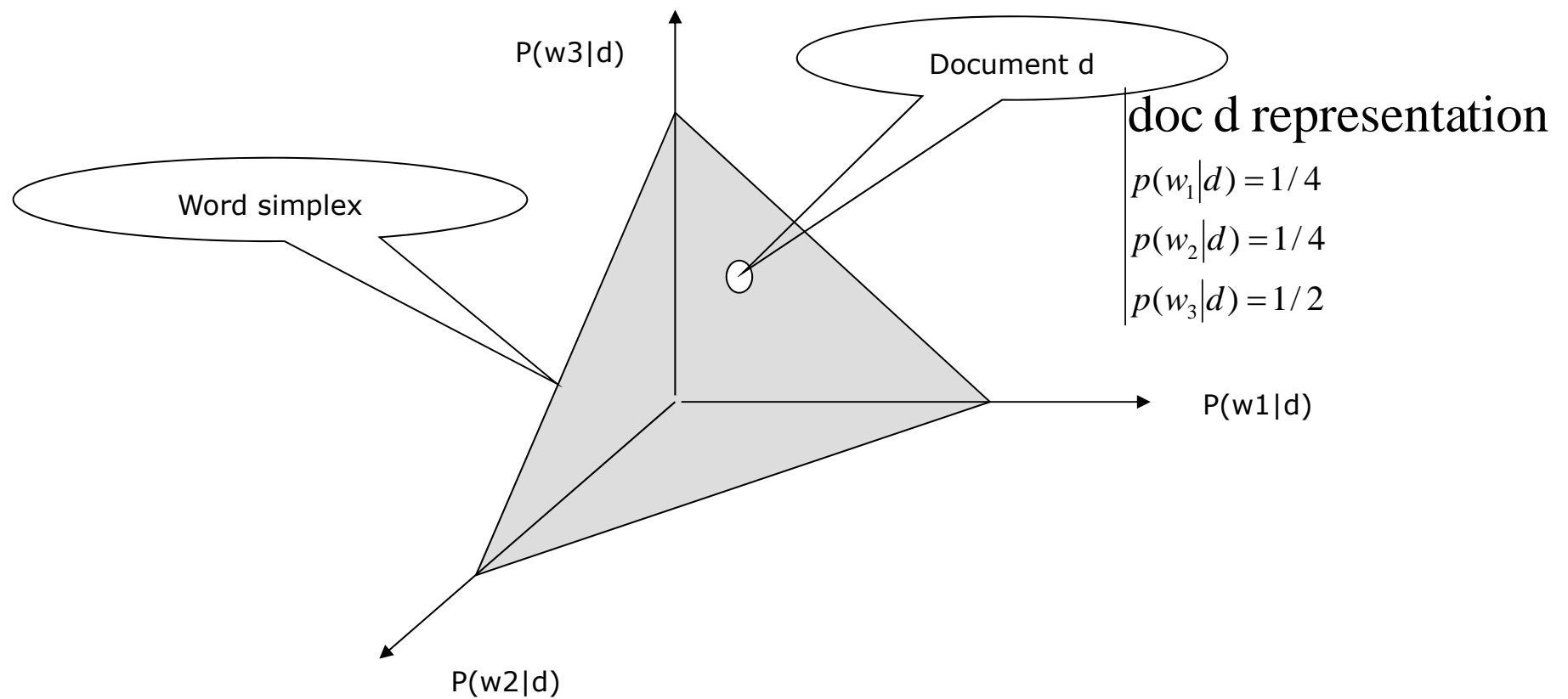
$$p(d) = \prod_i p(w_i|d)$$

- Select document length
- Pick a word w with probability $p(w)$
- Continue until the end of the document

- Applications

- Classification
- Clustering
- Ad-hoc retrieval (language models)

Preliminaries - Unigram model – geometric interpretation



Latent models for document generation

- Several factors influence the creation of a document (authors, topics, mood, etc).
 - They are usually unknown

- Generative statistical models
 - Associate the factors with latent variables
 - Identifying (**learning**) the latent variables allows us to uncover (**inference**) complex latent structures

Probabilistic Latent Semantic Analysis - PLSA (Hofmann 99)

☐ Motivations

- Several topics may be present in a document or in a document collection
- Learn the topics from a training collection
- Applications
 - ☐ Identify the semantic content of documents, documents relationships, trends, ...
 - ☐ Segment documents, ad-hoc IR, ...

PLSA

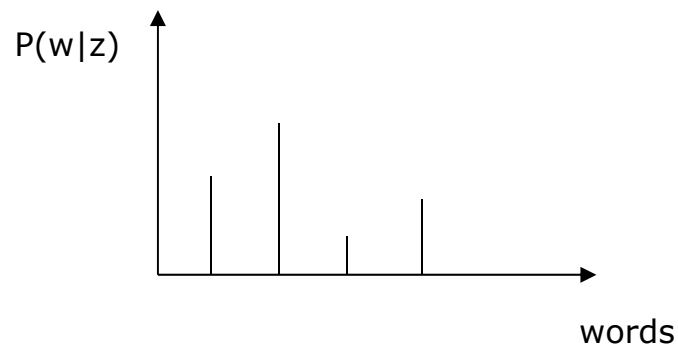
- The latent structure is a set of topics
 - Each document is generated as a set of words chosen from selected topics
 - A latent variable z (topic) is associated to each word occurrence in the document

- Generative Process
 - Select a document d , $P(d)$
 - Iterate
 - Choose a latent class z , $P(z/d)$
 - Generate a word w according to $P(w|z)$

 - Note : $P(w|z)$ and $P(z/d)$ are multinomial distributions over the V words and the T topics

PLSA - Topic

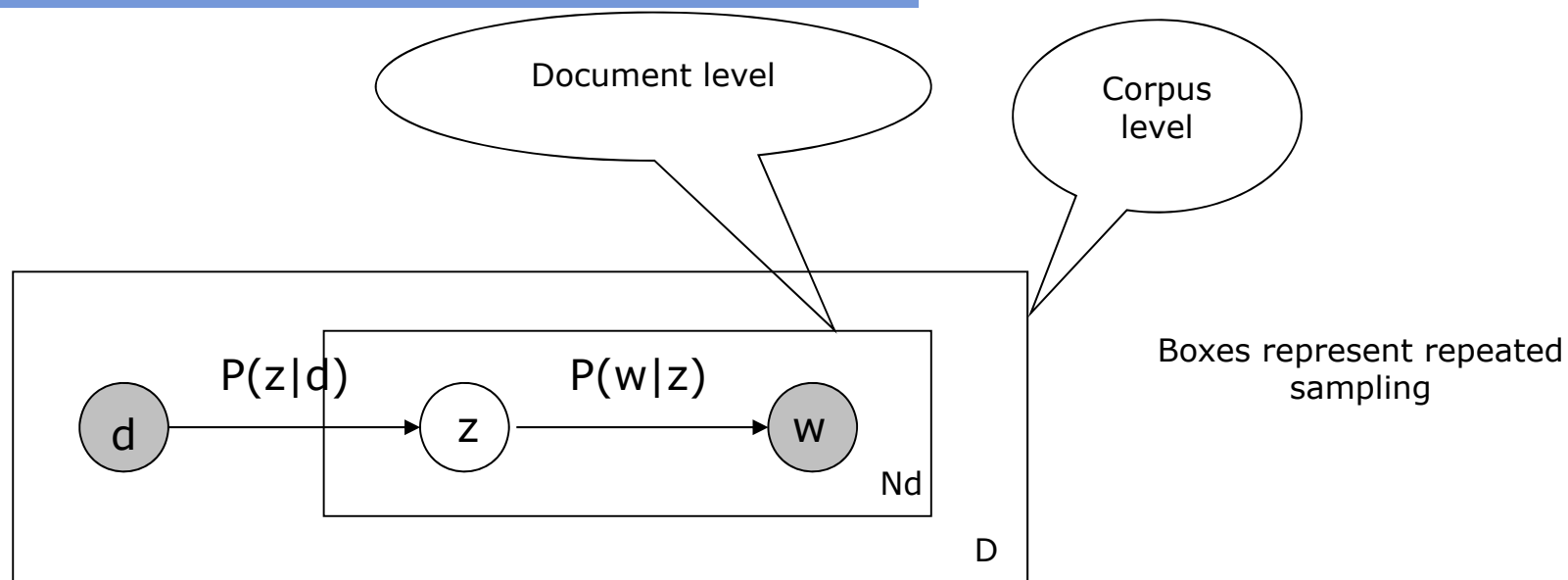
- A topic is a distribution over words



<i>word</i>	$P(w z)$
machine	0.04
learning	0.01
information	0.09
retrieval	0.02
.....

- Remark
 - A topic is shared by several words
 - A word is associated to several topics

PLSA as a graphical model



$$\begin{cases} P(d, w) = P(d) * P(w|d) \\ P(w|d) = \sum_z P(w|z)P(z|d) \end{cases}$$

PLSA model

☐ Hypothesis

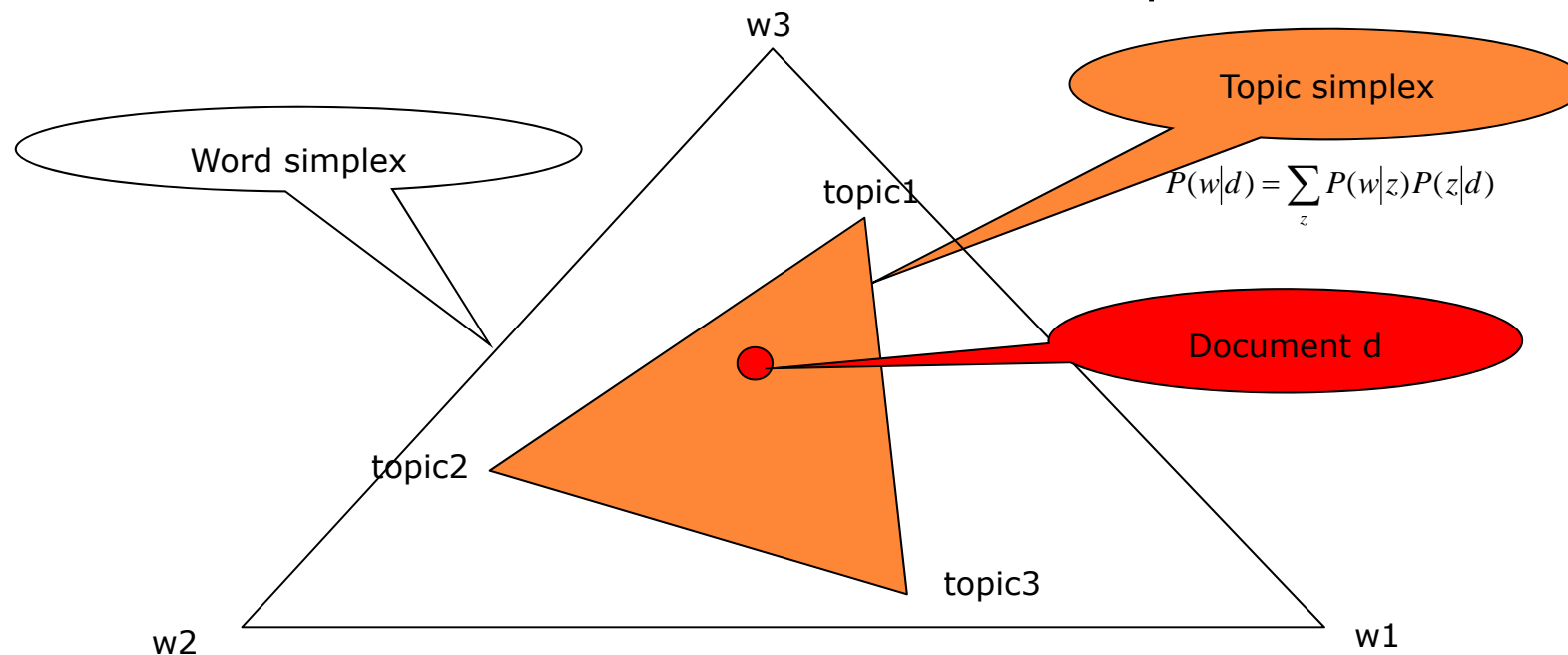
- # values of z is fixed a priori
- Bag of words
- Documents are independent
 - ☐ No specific distribution on the documents
- Conditional independence
 - ☐ z being known, w and d are independent

☐ Learning

- ☐ Maximum Likelihood : $p(\text{Doc-collection})$
- ☐ EM algorithm and variants

PLSA - geometric interpretation

- Topic_i is a point on the word simplex
- Documents are constrained to lie on the topic simplex
- Creates a bottleneck in document representation



Applications

- ☐ Thematic segmentation
- ☐ Creating documents hierarchies
- ☐ IR : PLSA model
- ☐ Clustering and classification
- ☐ Image annotation
 - Learn and infer $P(w/image)$
- ☐ Collaborative filtering

- ☐ Note : #variants and extensions
 - E.g. Hierarchical PLSA (see Gaussier et al.)

Latent Dirichlet Allocation - LDA (Blei et al. 2003)

- LDA is also a topic model
 - Extends PLSA
- Motivations
 - Generalization over unseen documents
 - Define a probabilistic model over documents
 - Not present in PLSA
 - Allows to generate (model) unseen documents
 - Overtraining
 - In PLSA, the number of parameters grows with the corpus size
 - LDA constrains the distribution of topics for each document and words for each topic

LDA - model

- Similar to PLSA with the addition of a prior distribution on the topic distribution
- Generative process
 - For a document
 - Topic distribution
 - Choose $\theta \sim \text{Dirichlet}(\alpha)$ a distribution over topics
 - Words
 - For each document word w
 - Choose a topic $z \sim \text{multinomial}(\theta)$
 - Choose a word w from $p(w \mid \theta, \Phi)$ multinomial probability conditioned on topic z

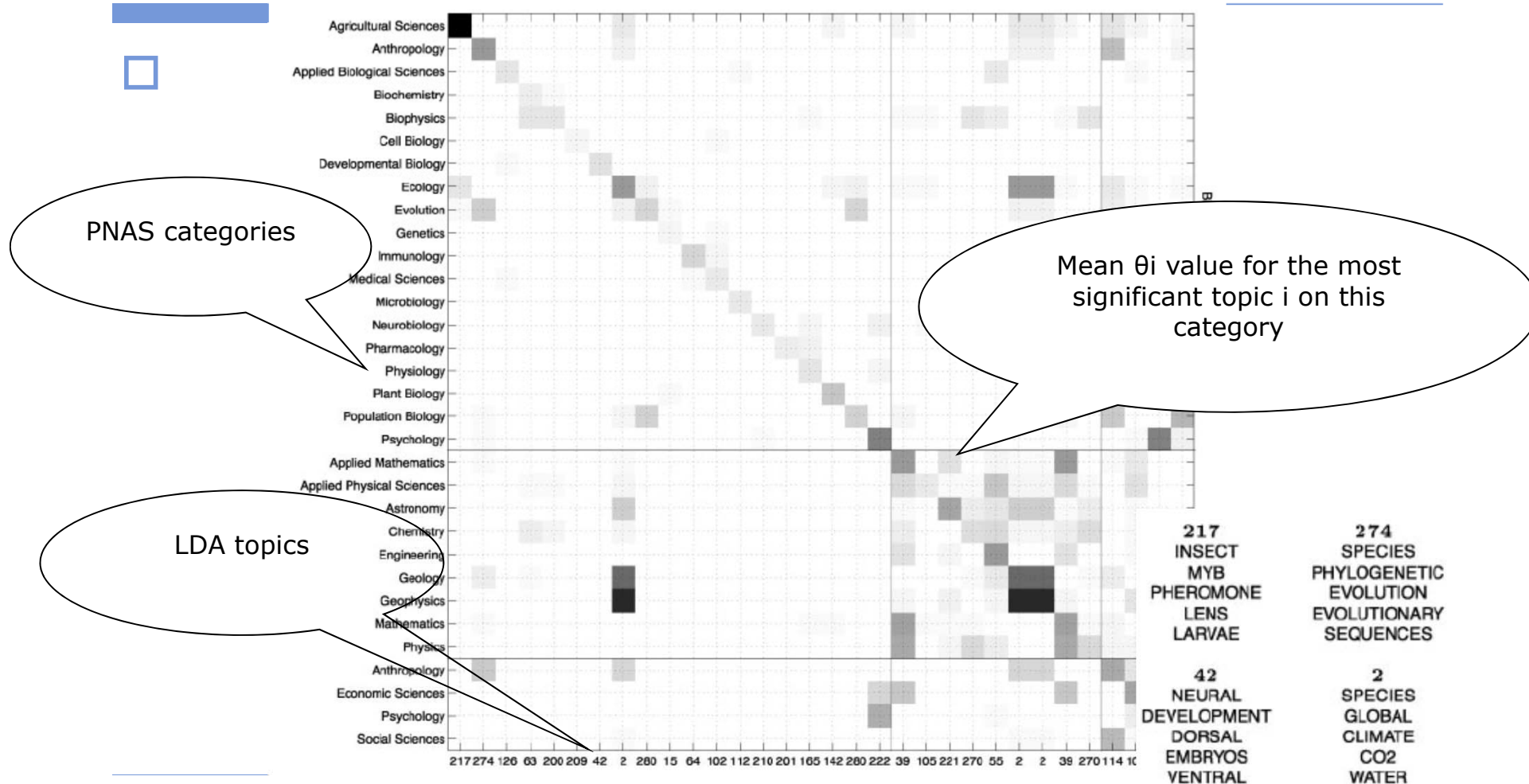
LDA tagging (Blei et al 2003)

“Arts”	“Budgets”	“Children”	“Education”
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

Figure 8: An example article from the AP corpus. Each color codes a different factor from which the word is putatively generated.

Finding topics in PNAS (Griffith et al. 2004)



Author-recipient topic model (McCallum et al. 2004)

Learning from Enron data

Identify

- Topic
- Author-recipient

Topic 5 “Legal Contracts”		Topic 17 “Document Review”		Topic 27 “Time Scheduling”		Topic 45 “Sports Pool”	
section	0.0299	attached	0.0742	day	0.0419	game	0.0170
party	0.0265	agreement	0.0493	friday	0.0418	draft	0.0156
language	0.0226	review	0.0340	morning	0.0369	week	0.0135
contract	0.0203	questions	0.0257	monday	0.0282	team	0.0135
date	0.0155	draft	0.0245	office	0.0282	eric	0.0130
enron	0.0151	letter	0.0239	wednesday	0.0267	make	0.0125
parties	0.0149	comments	0.0207	tuesday	0.0261	free	0.0107
notice	0.0126	copy	0.0165	time	0.0218	year	0.0106
days	0.0112	revised	0.0161	good	0.0214	pick	0.0097
include	0.0111	document	0.0156	thursday	0.0191	phillip	0.0095
M.Hain	0.0549	G.Nemec	0.0737	J.Dasovich	0.0340	E.Bass	0.3050
J.Steffes		B.Tycholiz		R.Shapiro		M.Lenhart	
J.Dasovich	0.0377	G.Nemec	0.0551	J.Dasovich	0.0289	E.Bass	0.0780
R.Shapiro		M.Whitt		J.Steffes		P.Love	
D.Hyvl	0.0362	B.Tycholiz	0.0325	C.Clair	0.0175	M.Motley	0.0522
K.Ward		G.Nemec		M.Taylor		M.Grigsby	
Topic 34 “Operations”		Topic 37 “Power Market”		Topic 41 “Government Relations”		Topic 42 “Wireless”	
operations	0.0321	market	0.0567	state	0.0404	blackberry	0.0726
team	0.0234	power	0.0563	california	0.0367	net	0.0557
office	0.0173	price	0.0280	power	0.0337	www	0.0409
list	0.0144	system	0.0206	energy	0.0239	website	0.0375
bob	0.0129	prices	0.0182	electricity	0.0203	report	0.0373
open	0.0126	high	0.0124	davis	0.0183	wireless	0.0364
meeting	0.0107	based	0.0120	utilities	0.0158	handheld	0.0362
gas	0.0107	buy	0.0117	commission	0.0136	stan	0.0282
business	0.0106	customers	0.0110	governor	0.0132	fyi	0.0271
houston	0.0099	costs	0.0106	prices	0.0089	named	0.0260
S.Beck	0.2158	J.Dasovich	0.1231	J.Dasovich	0.3338	R.Haylett	0.1432
L.Kitchen		J.Steffes		R.Shapiro		T.Geaccone	
S.Beck	0.0826	J.Dasovich	0.1133	J.Dasovich	0.2440	T.Geaccone	0.0737
J.Lavorato		R.Shapiro		J.Steffes		R.Haylett	
S.Beck	0.0530	M.Taylor	0.0218	J.Dasovich	0.1394	R.Haylett	0.0420
S.White		E.Sager		R.Sanders		D.Fossum	

Table 2: An illustration of several topics from a 50-topic run for the Enron email data set.

Each topic is shown with the top 10 words and their corresponding conditional probabilities. The quoted titles are our own summary for the topics. Below are

Recherche Web

Recherche Web

Introduction

RI Web vs RI classique

- Corpus
 - Taille, Nature, Dynamicité
- Contexte
 - Réseau, localisation, historique
- Individus
 - Grande variabilité
 - Prise en compte progressive des profils pour la recherche web

Individus

☐ Besoin

■ Transactionnel

- ☐ Achats en ligne

- ☐ Accéder à une ressource

 - Musique, livre, réservation avions – hotels,...

 - Météo, Google-Maps, downloads, ...

■ Informationnel

- ☐ Consultation

- ☐ Se renseigner sur un sujet

■ Navigation

- ☐ Joindre une page donnée

☐ Interaction

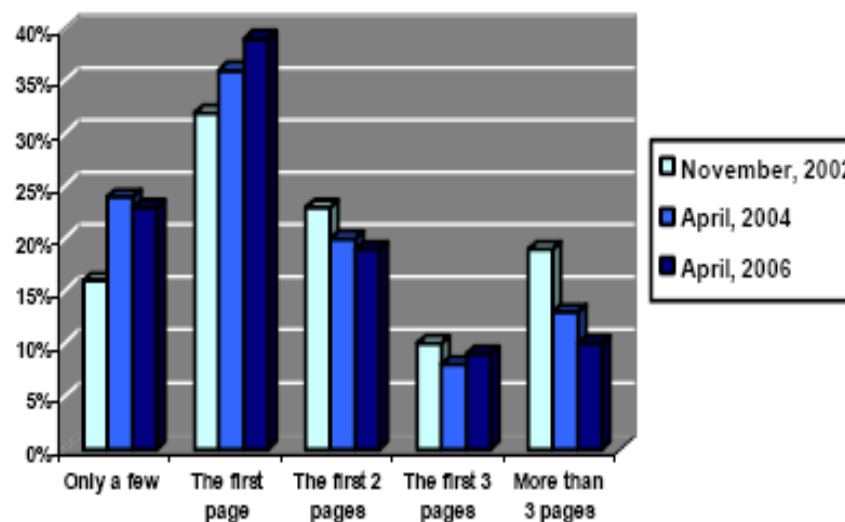
- Recall souvent peu important, precision mise en avant

Go-globe.com – juin 2011



Individus - exemple

"When you perform a search on a search engine and are looking over the results, approximately how many entries do you typically review before clicking one? (Select One)"



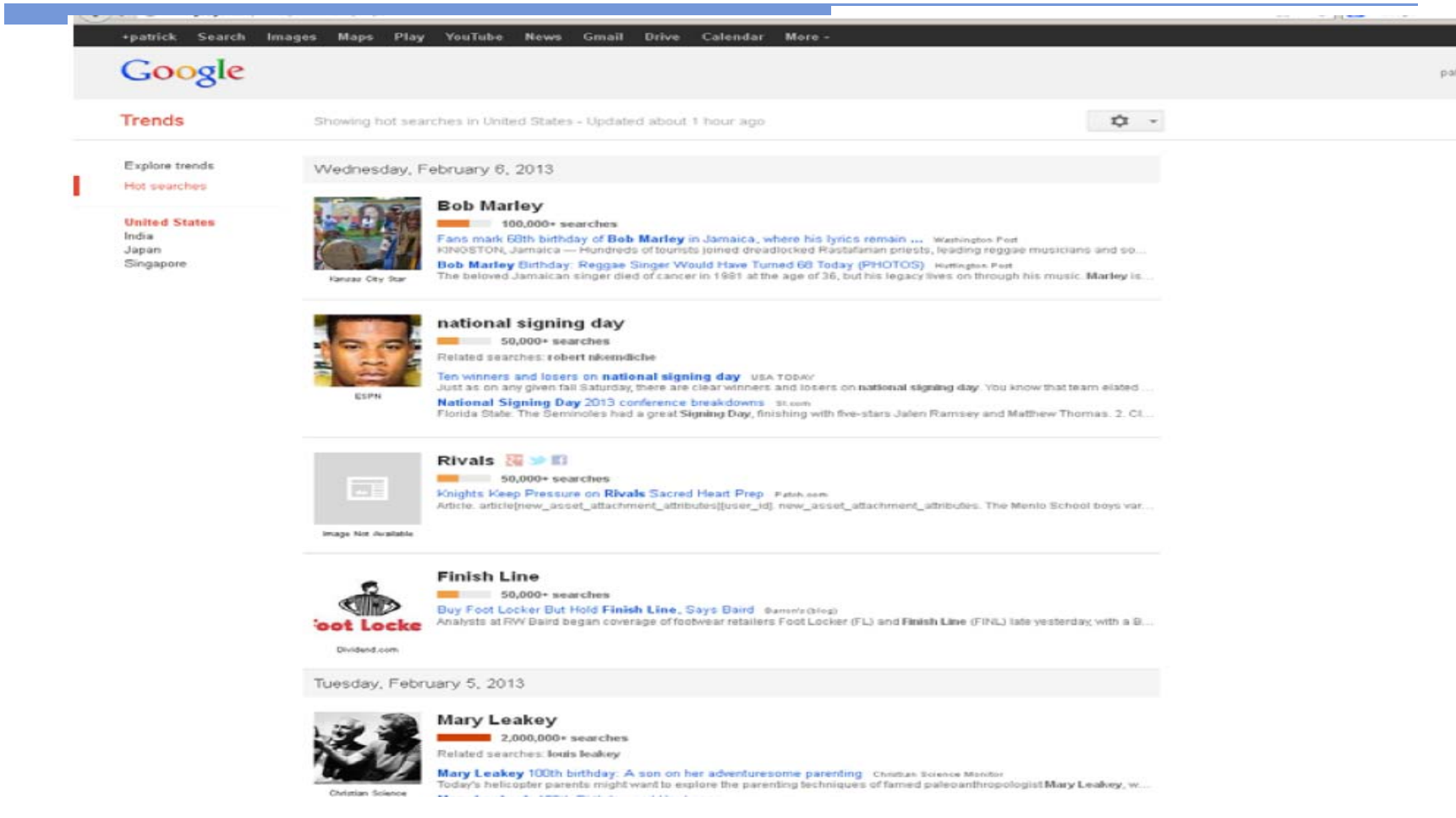
- http://www.iprospect.com/premiumPDFs/WhitePaper_2006_SearchEngineUserBehavior.pdf

Individus

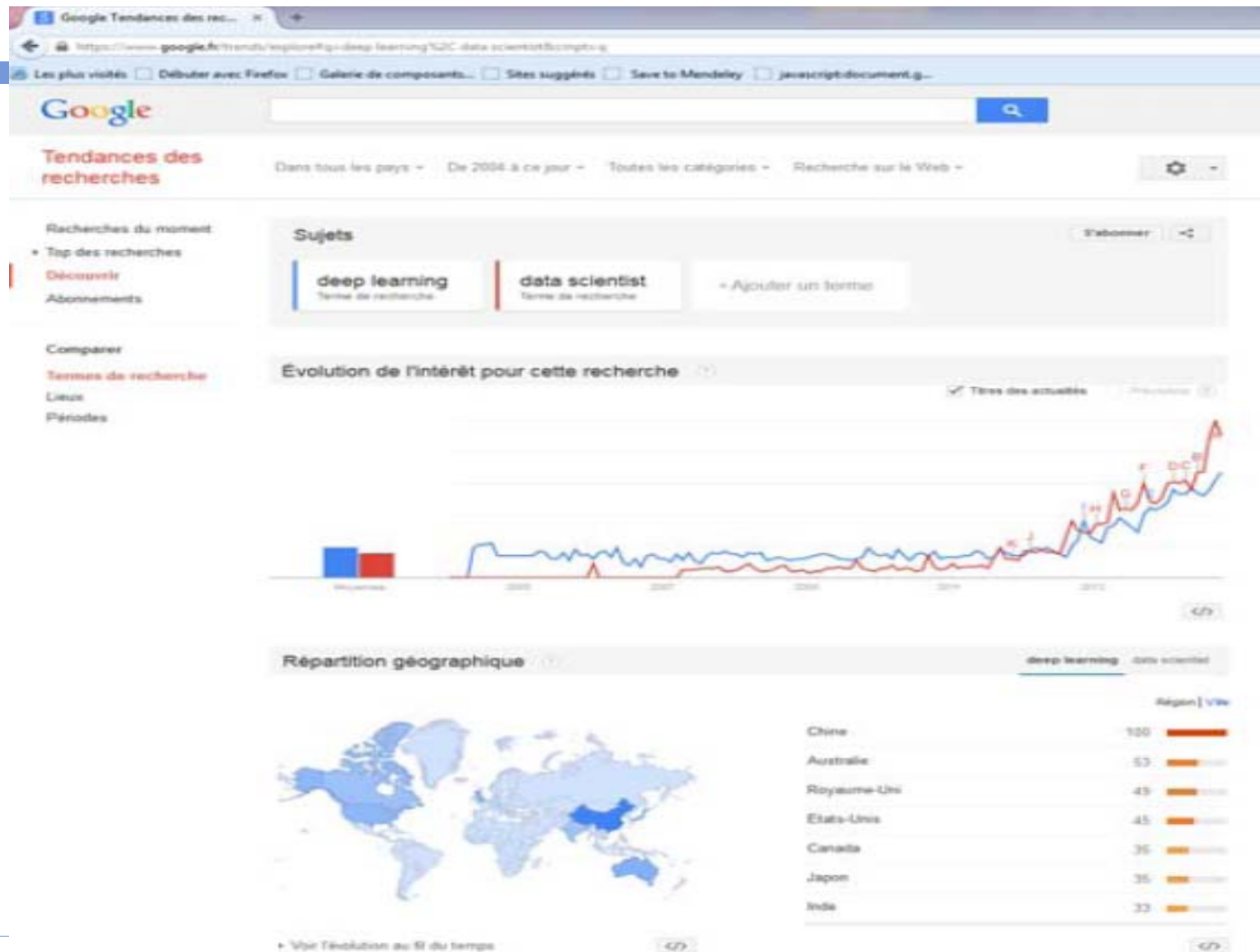
- Requêtes
 - Loi de puissance
 - peu de requêtes populaires
 - Beaucoup de requêtes rares
 - Taille moyenne requêtes < 3 mots
 - 1998, moyenne 2.35
 - 2001 moyenne 2.54
 - Besoins d'information dynamiques
 - Utilisation pauvre du langage de requête

Google Trends

Le buzz du moment : expressions à la plus forte progression

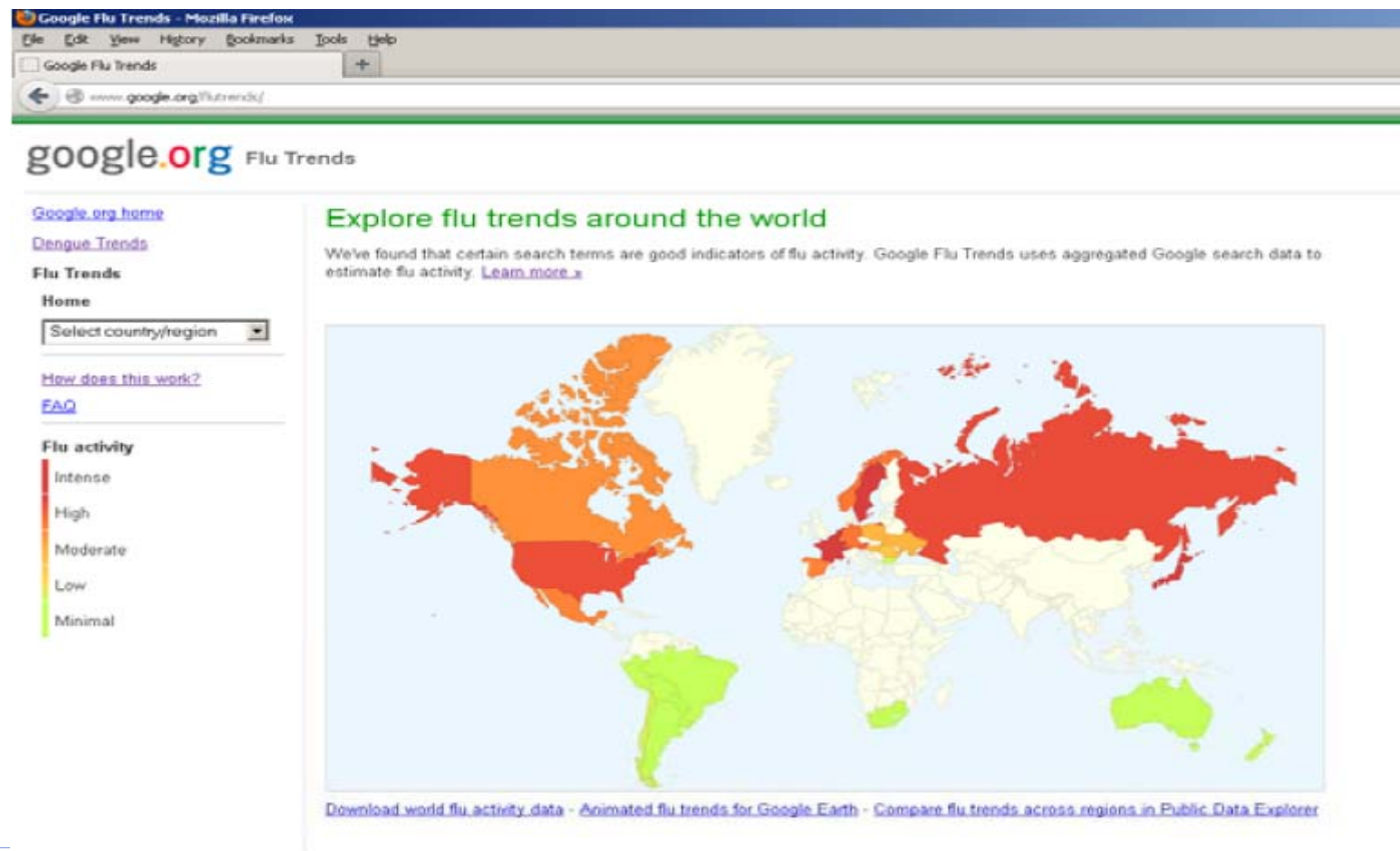


Les statistiques de Google Trends



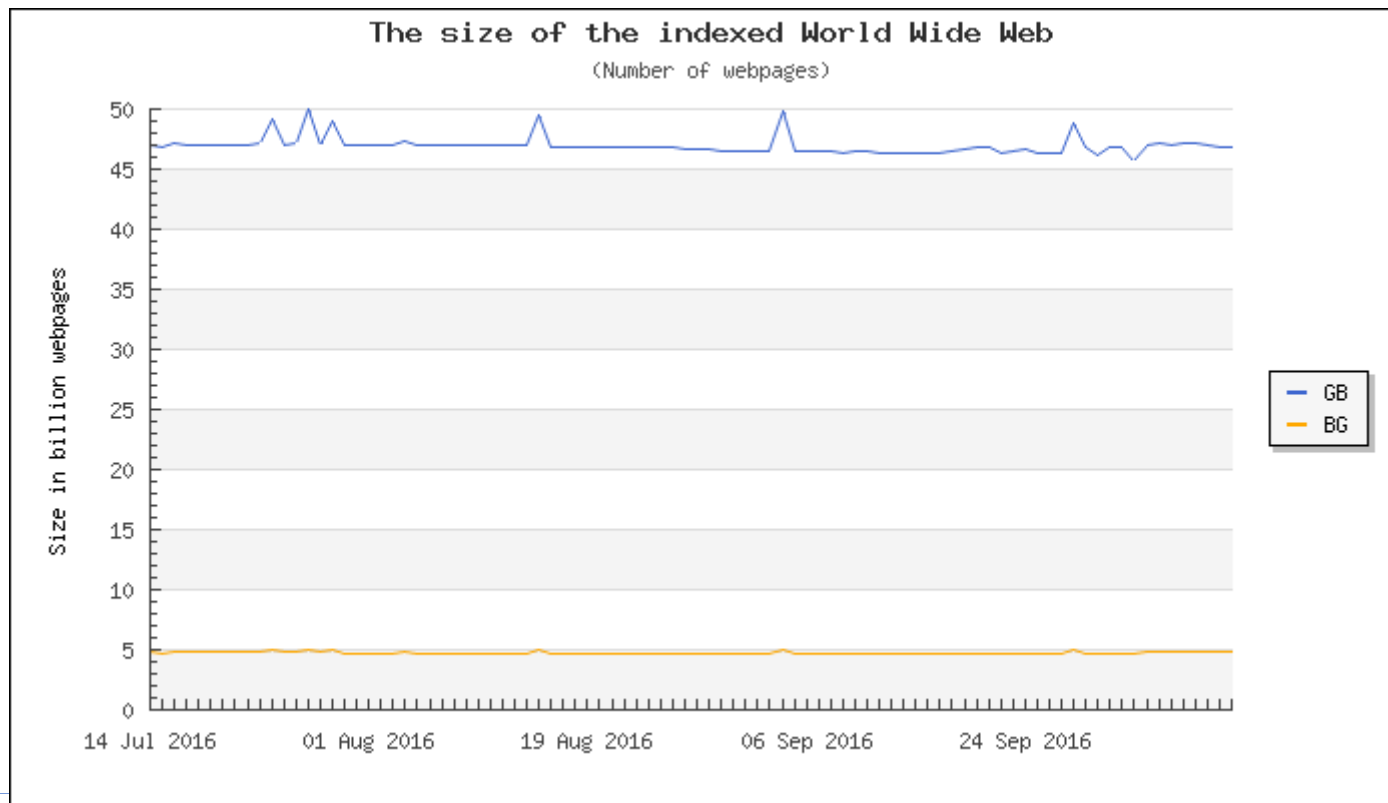
Recherche d'information textuelle

Flu Trends



Taille du Web indexé

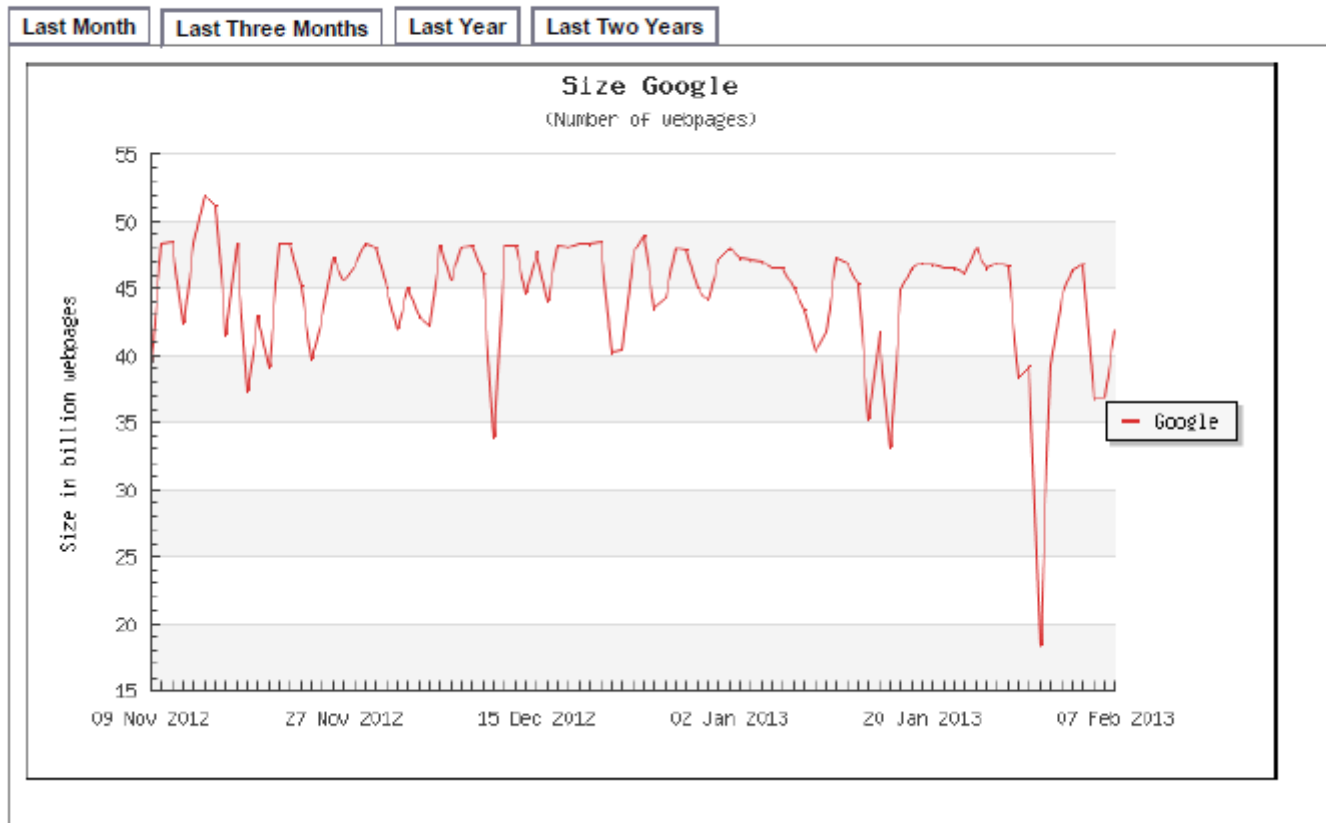
according to WorldWideWebSize.com



Taille du Web indexé par Google



The size of the World Wide Web:
Estimated size of Google's index



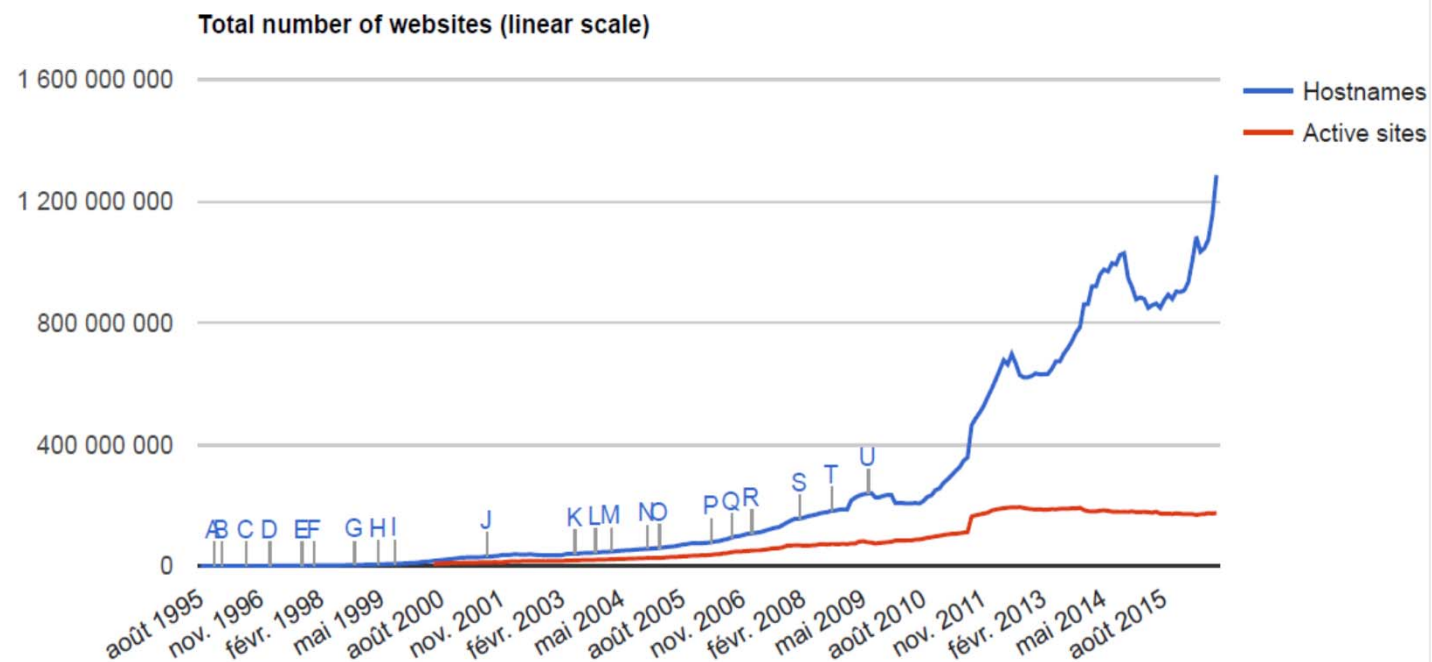
Corpus

- ☐ Croissance désordonnée
 - Pas de coordination
- ☐ Nature des informations
 - Contient des informations obsolètes, mensongères, etc
 - Texte, html, images, structuré (XML), BD,...
 - Statique vs dynamique
 - ☐ Le web dynamique n'est pas indexé
 - Quelques travaux
 - ☐ Web caché
 - 1 ou 2 facteurs d'échelles plus gros que le web visible ?
 - Multilingue
 - ☐ Difficulté des analyses lexicales

Corpus

- Forte croissance
 - La taille du web réel n'est pas connue
 - Qu'est ce qui est mesuré
 - Nombre d'hotes
 - Nombres de pages statiques
 - Etudes sur l'estimation du nombre de pages
 - Plusieurs méthodes : marches aléatoires, etc
 - Nombre de sites (cf Netcraft)

Croissance du web



- http://news.netcraft.com/archives/web_server_survey.html
- Total Sites Across All Domains August 1995 – August 2016

Structure globale du Web

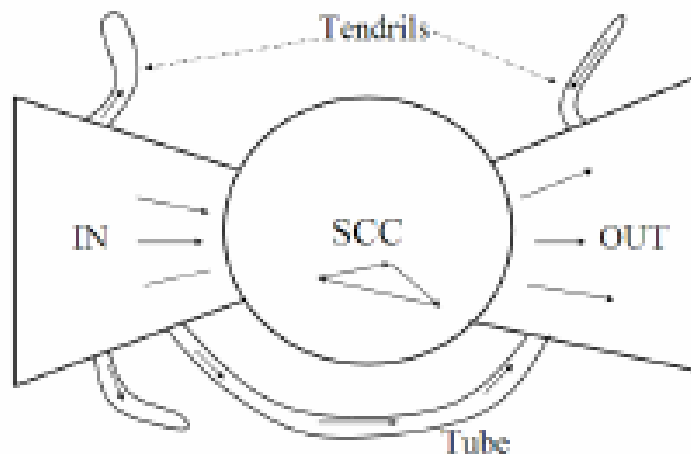
- Connexions

- Loi de puissance

- Le nombre de pages web de in-degeree i est proportionnel à $1/i^k$ avec $k = 2.1$

Bow-Tie shape of the web

- Trois grandes catégories de pages web
 - In, Out, SCC qui se distinguent par les possibilités de navigation



► Figure 19.4 The bowtie structure of the Web. Here we show one tube and three tendrils.

Navigation par hyperliens

In → SCC

SCC → Out

SCC → SCC

- From Manning et al. 2007

Spam et pub

Spam sur le Web

- Motivation : Search Engine Marketing
 - Référencement - Search Engine Optimization
 - Mettre en avant ses pages / son site dans les résultats des moteurs de recherche
 - Motivations
 - Diverses : commerciales, politiques, etc
 - Devenu une industrie
 - Les moteurs essaient de faire respecter des règles aux SEO
 - Paid placement, contextual advertising
 - Modèle de publicité à la Google
 - Google AdWords, Yahoo! Search Marketing, Microsoft adCenter
 - Le SEM est une des motivations majeures pour le SPAM
 - Guerre entre les spammers et les moteurs de recherche
 - Adversarial information retrieval

Bestiaire du Spam

☐ Modification du contenu

■ Keyword stuffing

- ☐ Répétition de termes pour augmenter le tf-idf
- ☐ Variantes : meta-tags, texte caché (couleur du fond ..), adresses url fréquemment demandées, etc
- ☐ générateurs de texte : pipotrons, patchworks, générateurs markoviens
- ☐ Visait les 1ers moteurs de recherche (tf-idf), facilement détecté actuellement
 - e.g. déréférencement de BMW par Google en 2006

■ Cloaking

- ☐ Délivrer des informations différentes suivant l'utilisateur (robot vs personne)
 - Permet d'indexer des pages avec des mots (robot) différents du contenu vu par l'utilisateur humain
 - Si la requête http provient d'un crawler : servir un faux contenu (fausse indexation)
 - Si la requête http provient du browser d'un utilisateur servir du spam

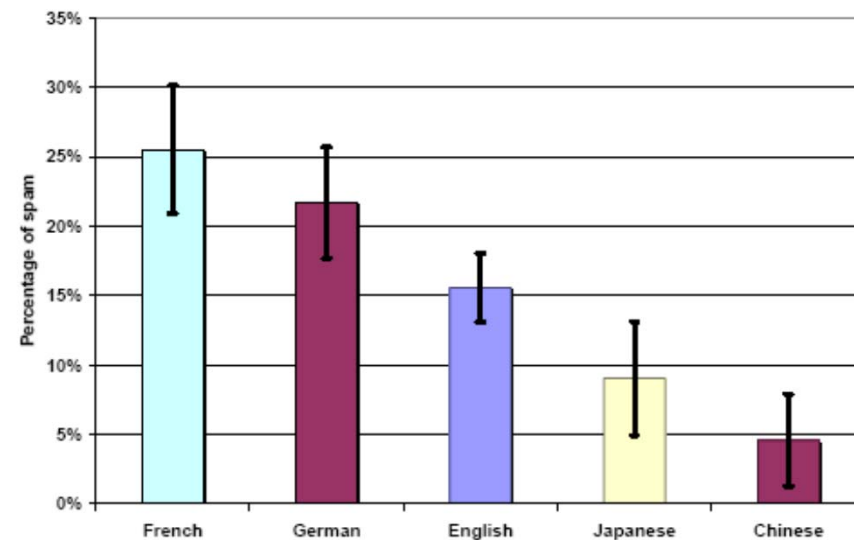
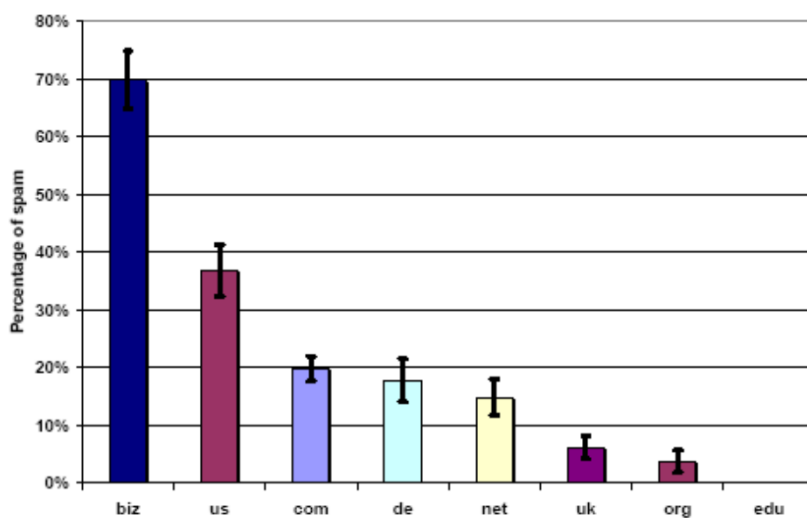
-
- Basés sur les liens: pompes à pagerank
 - Link farms
 - Référencement mutuel de sites
 - Développer un grand nombre de sites interconnectés qui pointent également sur des cibles dont on fait remonter le pagerank
 - Honey pot
 - Réplication de sites ou annuaires très référencés – le site sera ensuite référencé par d'autres utilisateurs et augmentera son rang
 - Blog ou wiki spam
 - Faire pointer sur son site à partir de sites où l'on peut écrire
 - Clic spam
 - Épuiser le crédit de concurrents en faisant cliquer que les liens sponsorisés (pay per clic model)
 - Camouflage
 - Doorway
 - Faire référencer une page avec un bon score (choix de mots clé, des liens etc)
 - L'utilisateur qui demande la page est renvoyé sur d'autres pages (commerciales etc)

- Parasitage

- ☐ recyclage de domaines expirés, cybersquatting
- ☐ pollution ou piratage de sites réputés fiables : **blogs**, forums, petites annonces...
- ☐ Botnets, clickbots (ClickBot.A)

- variantes

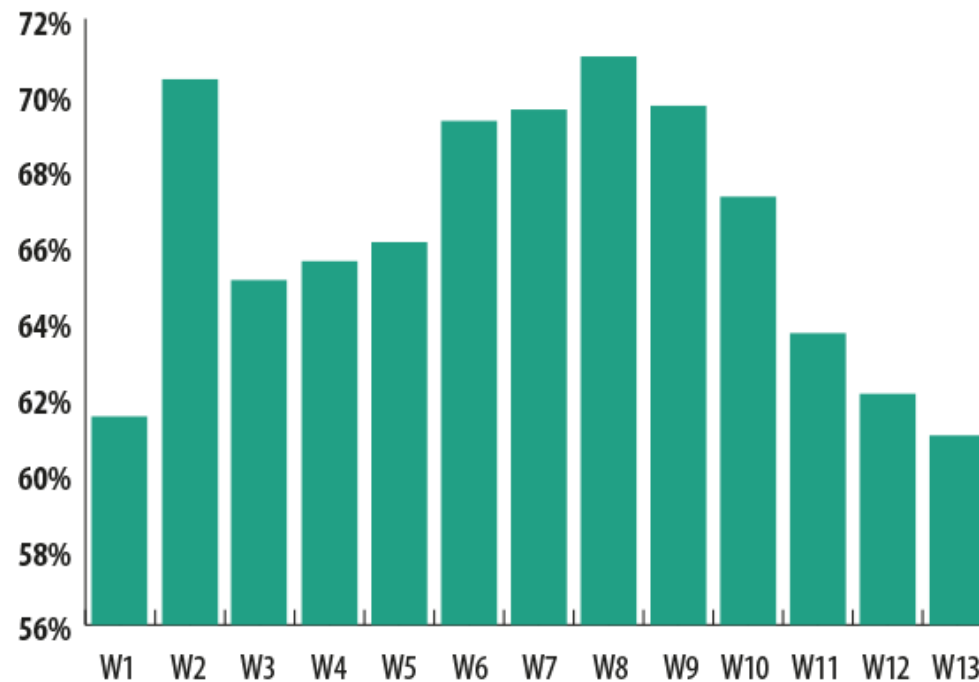
- ☐ Phishing
 - obtenir des renseignements personnels pour une usurpation d'identité. Faire croire à la victime qu'elle s'adresse à un tiers de confiance — banque, administration, etc. — afin de lui soutirer des renseignements personnels : mot de passe, numéro de carte de crédit, date de naissance, etc.
 - Cf mesure 2009 vérification transactions par les banques (sms de confirmation)
- ☐ Social spam : générateurs d'amis



- [Ntoulas et al. 2006], la figure 2 représente le taux de Web spam dans les 8 domaines les plus populaires sur le Web, la figure 3 le taux de spam dans les 5 langues les plus populaires. Ces statistiques sont calculées sur 100 millions de pages, globalement représentatives du Web.

Email spam 2014 1^{er} trimestre

<https://usa.kaspersky.com> (%trafic email qui est du spam)



Spamdexing : ferme à liens (Projet Madspam, T. Urvoy, Orange Labs)

The screenshot shows the Orange Labs website interface. At the top, there's a navigation bar with links like 'web', 'images', 'shopping', 'dans le site', 'annuaire', and 'plus ...'. Below this is a search bar containing 'cuisine nouvelle' and a 'rechercher' button. To the right of the search bar are links for 'espace client', 'assistance', 'offres et boutiques', and 'repères mobile'.

The search results section shows '11 646 413 réponses en 0,17 s'. Under 'Liens commerciaux', there are links for 'Cuisine' (www.hygena.fr/cuisine) and 'Cuisines Mondial Kit' (www.cuisines-mondialkit.fr). Under 'Résultats Web', there are several search results. Two of these results are highlighted with red boxes and arrows pointing to each other, indicating a spamdexing technique. The first highlighted result is a link to 'www.cuisine-nouvelle.com/' with a title that repeats the phrase '- recette champignons recette champignons recette champignons -' and a description that includes 'cuisine REÇOIS LES MEILLEURS LOGICIELS PAR E-MAIL ... humour mon annuaire blagues en flash recettes de cuisine tuning ...'. The second highlighted result is a link to 'www.cuisine-nouvelle.com/' with a title that repeats the same phrase and a description that includes 'Recette de cuisine, nouvelle, essais, petites histoires à faire...'. A red arrow points from the first highlighted result to the second, and a red question mark is placed next to the second result.

Orange Labs - R&D - méthodes automatiques pour la détection du spamdexing - mars 2009

Spamdexing : ferme à liens

elle amour
en flash

horoscope
recettes de cuisine

photos humour
tunning


mon a
sonneries

Ces services, gratuits pour l'instant, sont proposés en partenariat avec des **dessins illusion optique** sociétés, telles , ou . Le principe est simple, il suffit de s'abonner via **dessins illusion optique** Internet ou d'envoyer un SMS au numéro donné et l'on reçoit en retour un MMS **dessins illusion optique** avec des photos de l'appartement de ses rêves, un autre avec le temps qu'il fait **dessins illusion optique** sur ou encore un autre avec les images de la soirée précédente, passée dans un **dessins illusion optique** endroit branché... , de son côté, attend de mieux cerner les souhaits de sa **dessins illusion optique** clientèle pour lancer d'éventuels services. Quant à , ses porte-parole se sont **dessins illusion optique** refusés à tout commentaire avant la mise en service officielle des MMS.

<http://www.blagues-histoires-droles.com>
<http://www.zoneillusions.org>
<http://www.carte-a-rire.com>
<http://www.illusion-optique.com>
<http://www.videos-comiques.com>
<http://www.videos-rigolo.com>
<http://www.blaguissimo.com>
<http://www.flash-rires.com>

Recherche d'information textuelle

Spamdexing : ferme à liens

 "avec des photos de l'appartement de ses rêves" [Recherche avancée](#)
[Préférences](#)

Rechercher dans : ☒ Web ☐ Pages francophones ☐ Pages : France

Web Résultats **1 - 10** sur un total d'environ **5 990** pour "**avec des photos de l'appartement de ses rêves**"

[photos tuning sur www.auto customise.com](#)
... tuning Internet ou d'envoyer un SMS au numéro donné et l'on reçoit en retour un MMS photos tuning **avec des photos de l'appartement de ses rêves**, ...
[photos-tuning.auto-customise.com/](#) - 39k - [En cache](#) - [Pages similaires](#)

[blague telefonique blague telefonique blague telefonique](#)
... et l'on reçoit en retour un MMS blague telefonique **avec des photos de l'appartement de ses rêves**, un autre avec le temps qu'il fait blague telefonique ...
[blague-telefonique.flash-rires.com/](#) - 14k - [En cache](#) - [Pages similaires](#)

[forum edonkey sur www.mon edonkey.com](#)
... forum edonkey Internet ou d'envoyer un SMS au numéro donné et l'on reçoit en retour un MMS forum edonkey **avec des photos de l'appartement de ses rêves**, ...
[forum-edonkey.mon-edonkey.com/](#) - 16k - [En cache](#) - [Pages similaires](#)

[recette buffet recette buffet recette buffet](#)
... buffet Internet ou d'envoyer un SMS au numéro donné et l'on reçoit en retour un MMS recette buffet **avec des photos de l'appartement de ses rêves**, ...
[www.recettes-exotiques.com/](#) - 15k - [En cache](#) - [Pages similaires](#)

[sites de blague sites de blague sites de blague](#)
... de blague Internet ou d'envoyer un SMS au numéro donné et l'on reçoit en retour un MMS sites de blague **avec des photos de l'appartement de ses rêves**, ...
[sites-de-blague.rire-et-photos.com/](#) - 13k - [En cache](#) - [Pages similaires](#)

Liens commerciaux

[Photos Appartement](#)
Cherchez Photos Appartement
Photos Appartement sur Ask!
[www.ask.com](#)

Plus de 5000 pages ventilées sur des centaines de sites

Recherche d'information textuelle

Spam blogs

Grands sourires

Blog à la ciboulette

Chez Kek (le blog en cours)

Chez kek le blog d'avant

Le journal de Max

les amis de cali et kek

Pour l'esprit

arrêt sur pillages

colères essentielles du
superflu

Le blog de Pek (news
graphisme et design)

Commentaires

On n'est peut être comme les derniers dinosaures...

Ecrit par : [Lew](#) | jeudi, 06 janvier 2005

Moi je pense qu'on mourra tous par la faute d'humain mais ce sera les plus andouilles qui y arriveront et ceux qui se battent pour notre bien perdront leur bataille.

Le pire dans tout cela et s'il y a autant de "con" (désolé pour le terme tu pourras l'effacer si tu le souhaite mais je le trouve adapté) c'est à faute de tout les OUBLIS.

On oublis vite le drame pour en faire à notre, à quoi sert notre histoire????

Ecrit par : [Ludoffy](#) | jeudi, 06 janvier 2005

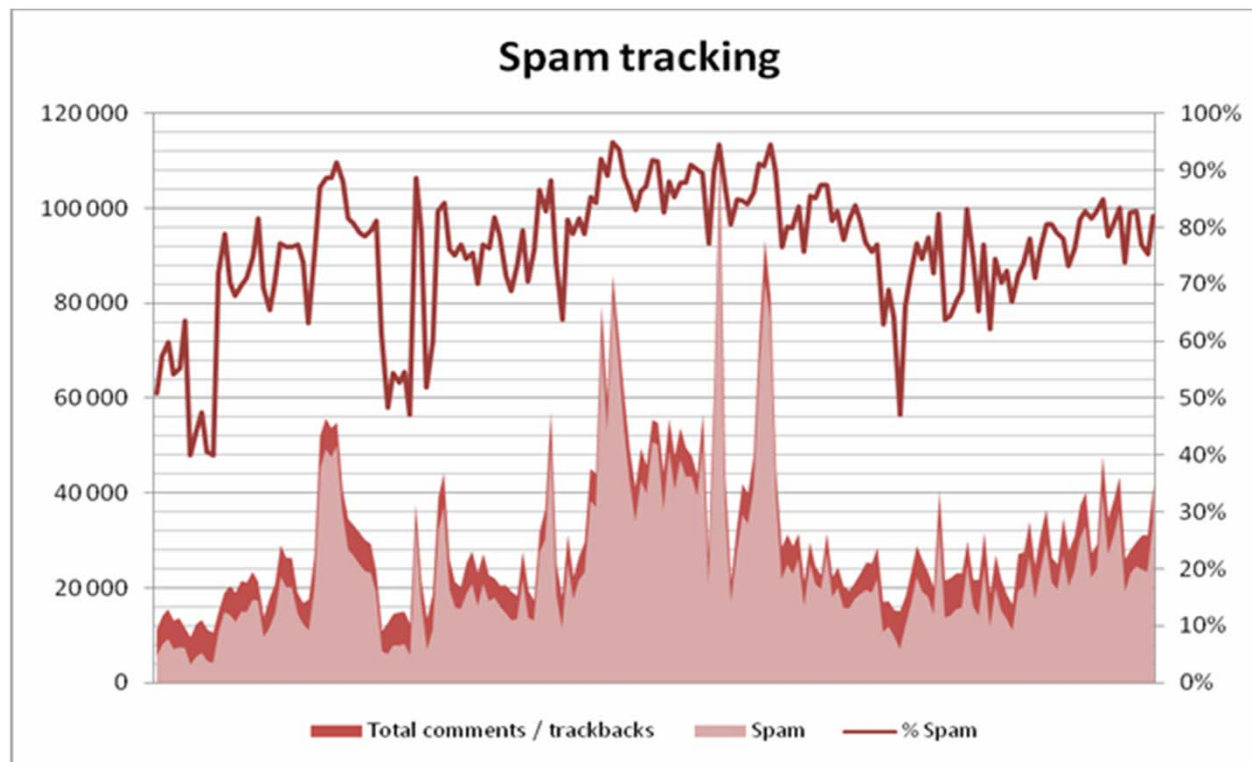
Ludoffy : te voilà bien pessimiste.. Les blogs et le web sont là pour nous aider à éviter l'oubli.

Ecrit par : [KroniK](#) | jeudi, 06 janvier 2005

You are invited to check out some information in the field of texas hold em texas hold em
<http://www.jmhic.com/texas-hold-em.html> ... Thanks!!!

Spam Blogs (Société BlogSpirit)

□ pourcentage de Spam - 2007



La lutte contre le Spam

- Editorial
 - Blacklists, dénonciation (Google), ...
 - <http://www.google.com/contact/spamreport.html>
- Usage
 - Préférer les pages très utilisées, bien référencées
- Analyse de liens
 - Guilt by association
 - Algos robustes de référencement
- Machine learning
 - Cf **Adversial retrieval** initiative : Airweb
 - <http://airweb.cse.lehigh.edu/>

Publicité sur le Web

- Différents modèles
 - Cost per Mil (CPM)
 - Paye au nombre de fois où la bannière est affichée
 - Cost per Clic (CPC)
 - Paye au nombre de fois où la bannière est cliquée
 - Clic spam
 - Epuiser le crédit d'un concurrent en cliquant sur ses bannières

Les clics sur les adds

Etude 2008 – sur 68 million de domaines

<http://www.attributor.com/blog/get-your-fair-share-of-the-ad-network-pie/>

Ad Server Market Share

Ad Server	Monthly Unique Users	Market Share	Unique Domains	Market Share
Google	1,107,489,739	35.30%	91,462	77.28%
DoubleClick	1,079,203,140	34.39%	6,748	5.70%
Yahoo	362,201,931	11.54%	5,147	4.35%
MSN	309,290,121	9.86%	8,099	6.84%
AOL	156,109,326	4.98%	1,976	1.67%
Adbrite	73,446,676	2.34%	3,575	3.02%

Ad Server Market Share Grouped by Site Traffic

Ad Server	< 100k UU's	100k – 1MM UU's	> 1MM UU's
Adbrite	4.07%	4.90%	0.48%
AOL	1.95%	6.55%	5.74%
DoubleClick	9.13%	29.92%	48.04%
Google	71.38%	41.56%	15.85%
MSN	6.57%	6.30%	12.85%
Yahoo	4.66%	7.33%	16.49%

PageRank et Hits

Analyse de liens

Analyse de lien

- Popularisée par Google avec PageRank
- Actuellement une composante parmi beaucoup d'autres des moteurs de recherche
 - De l'ordre de 400 caractéristiques prises en compte
- Cours : 2 algorithmes historiques
 - PageRank (Brin & Page 1998)
 - HITS (Kleinberg 1998)
 - Très nombreuses variantes
 - E.g. trustrank

Les liens

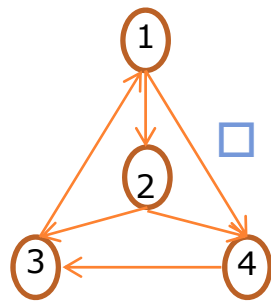
- Le web est vu comme un graphe orienté
- Les liens sont porteurs d'information
 - Un lien entre pages indique une relation de pertinence
 - Un lien est un indicateur de qualité
 - Le texte d'un lien résume la page cible
 - L'indexation d'une page doit prendre en compte les liens vers cette page (contexte)

PageRank - details

- General
 - Popularized by google
 - Assign an authority score for each web page
 - Using only the structure of the web graph (query independent)
 - Now one of the many components used for computing page scores in Google S.E.
- Intuition
 - Assign higher scores to pages with many in-links from authoritative pages with few out-links
- Model
 - Random surfer model
 - Stationary distribution of a Markov Chain
 - Principal eigenvector of a linear system

Notations

- $G = (V, E)$ graph
- A *adjacency* matrix
 - Binary matrix
 - $a_{ij} = 1$ if there is a link between i and j
 - $a_{ij} = 0$ otherwise
- P *transition* matrix
 - $P = \left(p_{ij} = \frac{a_{ij}}{d_i} \right), i, j = 1..n$
 - d_i degree of node v_i ($d_i = \sum_j a_{ij}$)
 - p_{ij} is the probability to move from node i to node j in the graph
 - P is *row stochastic*
 - $\sum_j p_{ij} = 1$



$$\square \quad A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

\square Basic PageRank

- Initialize the PageRank score vector to a stochastic vector e.g. $p(0) = \frac{1}{4} \mathbf{1}$

- Update the PRank vector until convergence

$$\square \quad p(k+1) = P^T p(k)$$

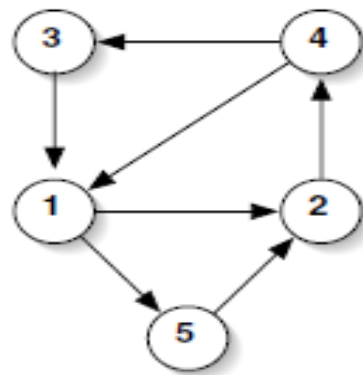
$$\square \quad p(0) = \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}, \quad p(1) = \begin{pmatrix} 2/8 \\ 1/8 \\ 3/8 \\ 2/8 \end{pmatrix}, \quad p(2) = \begin{pmatrix} 6/16 \\ 2/16 \\ 5/16 \\ 3/16 \end{pmatrix}, \quad \dots$$

- \square Conditions for convergence, unicity of solution ?

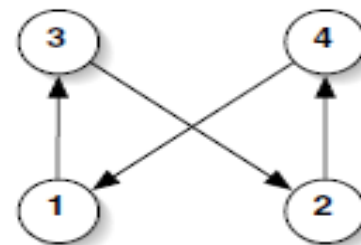
Non Negative Matrices

- A square matrix $A_{n \times n}$ is *non negative* if $a_{ij} \geq 0$
 - Notation $A \geq 0$
 - Example: graph incidence matrix
- $A_{n \times n}$ is *positive* if $a_{ij} > 0$
 - Notation $A > 0$
- $A_{n \times n}$ is *irreducible* if
 - $\forall i, j, \exists t \in \mathbb{N} / (A^t)_{ij} > 0$
 - If A is a graph incidence matrix, this means that G is strongly connected
 - There is a path between any pair of vertices
- $A_{n \times n}$ is *primitive* if $\exists t \in \mathbb{N} / A^t > 0$
 - A primitive matrix is irreducible
 - Converse is false

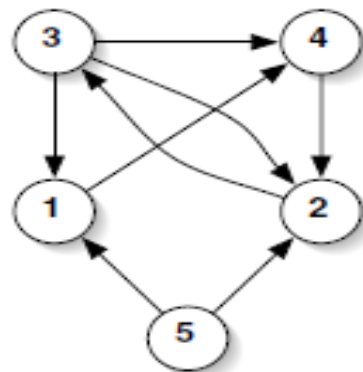
Examples (Baldi et al. 2003)



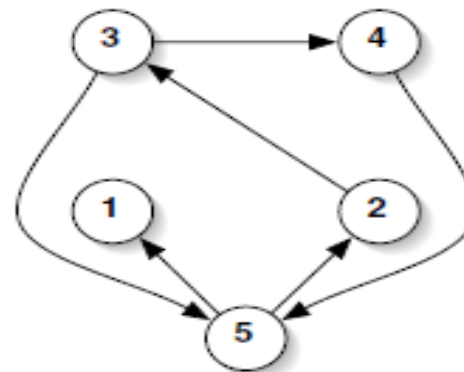
(a)



(b)



(c)



(d)

Figure 5.1 Graphs with different types of Markov chains. (a) is primitive, (b) is irreducible (with period 4) but not primitive, (c) and (d) are reducible.

Perron-Frobenius theorem

- $A_{n \times n}$ a non negative irreducible matrix
 - A has a real and positive eigenvalue λ / $\lambda \geq |\lambda'|$ for any other eigenvalue λ'
 - λ corresponds to a strictly positive eigenvector
 - No other eigenvector is positive
 - λ is a simple root of the characteristic equation $(A - \alpha I_n) = 0$

- Remarks
 - λ is called the *dominant eigenvalue* of A and the corresponding eigenvector the *dominant eigenvector*
 - The dominant eigenvalue is denoted λ_1 in the following
 - There might be other eigenvalues λ_j / $|\lambda_j| = |\lambda_1|$
 - e.g. $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is non negative and irreducible, with two eigenvalues 1, -1 on the unit circle

-
- Perron-Frobenius theorem for a primitive matrix
 - In property 1, the inequality is strict
 - i.e. A has a real and positive eigenvalue $\lambda_1 / \lambda_1 > |\lambda'|$ for any other eigenvalue λ'
 - For a primitive stochastic matrix
 - $\lambda_1 = 1$ since $A\mathbf{1} = \mathbf{1}$
 - Why is it interesting ?
 - Simple procedure for computing the eigenvalues of a matrix using the powers of a matrix

Intuition on the power method

□ Let

- $x \in R^n$
- (u_1, \dots, u_n) the eigenvectors of A ,
- (c_1, \dots, c_n) the coordinates of x in the eigenvector basis

□ Then

- $Ax = \sum_i c_i \lambda_i u_i$
- $A^t x = \sum_i c_i \lambda_i^t u_i$
- λ_1 dominates, then $A^t x \rightarrow c_1 \lambda_1^t u_1$ for t large
 - True if x non orthogonal to u_1
 - Since u_1 positive, any positive vector will do
 - e.g. $x = \mathbf{1}_n$

Power method

- Let A be a primitive matrix
- Start with an arbitrary vector x_0
 - $y_t = Ax_t$
 - $x_{t+1} = y_t / \|y_t\|$
- Convergence
 - Converges towards u_1 the eigenvector associated to λ_1 , the largest eigenvalue of A .
 - Whatever the initial vector x_0
- Rate of convergence
 - Geometric with ratio $|\lambda_2|/|\lambda_1|$
 - $\lambda_1 > \lambda_2$ are the first two dominant eigenvalues of A

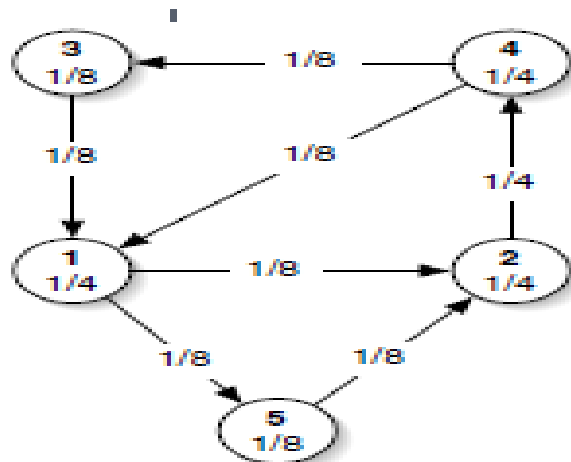
Pagerank

- Recall
 - G a directed graph (Web)
 - A its adjacency matrix
 - P the transition matrix

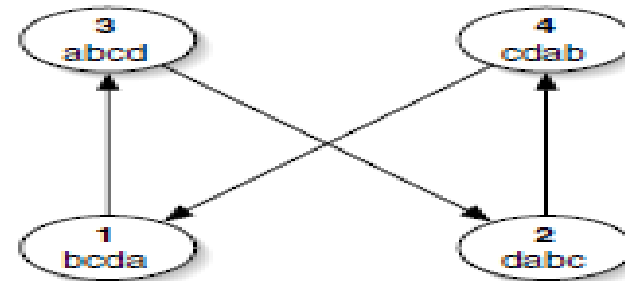
□ Intuition

- Rank of a document is high if the rank of its parents is high
- Embodied e.g. in
 - $r(v) \propto \sum_{w \in \text{parents}(v)} \frac{r(w)}{\text{outdegree}(w)}$
 - $r(v)$: rank value at v
- Each parent contributes
 - Proportionally to $r(w)$
 - Inversely to its out degree
- Amounts at solving
 - $r = Mr$ for a given matrix M
 - Eigenvector problem

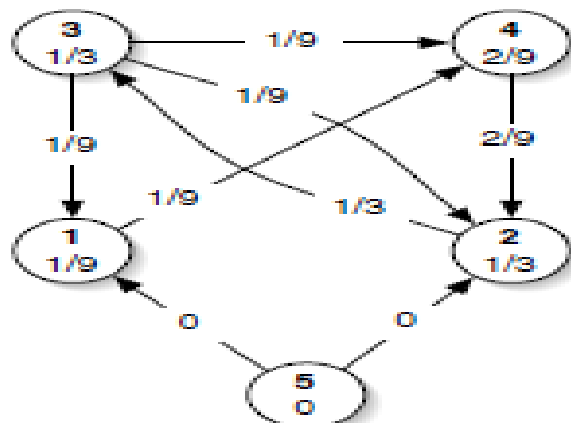
E



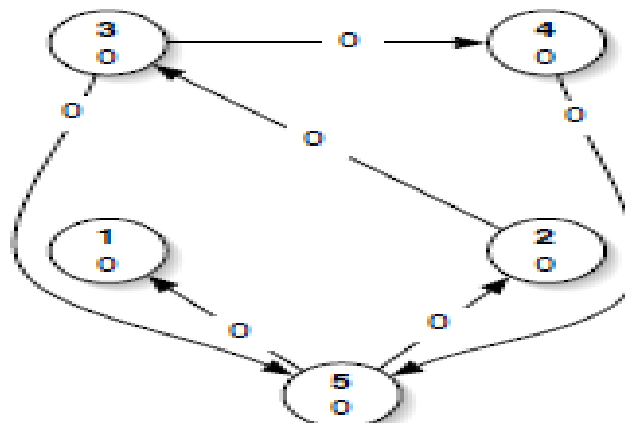
(a)



(b)



(c)



(d)

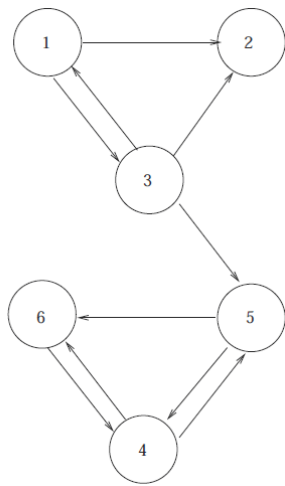
Figure 5.5 Rank propagation on graphs with different types of incidence matrices. Equation (5.14) converges to a nontrivial steady state in case (a) and (c), to a limit cycle in case (b), and to zero in case (d).

-
- In order to converge to a stationary solution
 - Remove sink nodes
 - Many such situations in the web
 - Images, files, etc
 - Make M primitive

Adjustments of the P matrix

- The transition matrix P most often lacks these properties
 - Stochasticity
 - Dangling nodes (nodes with no outlinks) make P non stochastic.
 - Rows corresponding to dangling nodes are replaced by a stochastic vector v
 - a common choice is $v = \frac{1}{n}\mathbf{1}$, with $\mathbf{1}$ the vector of 1s
 - The new transition matrix is
 - $P' = P + a.v^T$
 - $a_i = 1$ if i is a dangling node, 0 otherwise
 - P' is row stochastic

Example (Langville&Meyer 2006)



$$\mathbf{P} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$\bar{\mathbf{P}} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

□ Primitive

- The M matrix shall be primitive in order for the PageRank vector to exist
- One possible solution is
 - $P'' = \alpha P' + (1 - \alpha) \mathbf{1} \cdot v^T$ with $0 < \alpha \leq 1$ and v a stochastic vector
 - Different v correspond to different random walks
 - v uniform $\frac{1}{n} \mathbf{1}$: teleportation operator in the random walk model
 - v non uniform: personalization vector
 - P'' is a mixture of two stochastic matrices
 - It is stochastic
 - P'' is trivially primitive since every node is connected to itself
 - α controls the proportion of time P' and $\mathbf{1} \cdot v^T$ are used
 - α also controls convergence rate of the Random Walk
 - P'' is called the Google matrix

□ Example (Langville&Meyer 2006)

$$\bar{\bar{\mathbf{P}}} = \alpha \bar{\mathbf{P}} + (1 - \alpha) \mathbf{e} \mathbf{e}^T / n = \begin{pmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{pmatrix}$$

Two formulations of the PageRank problem

1. Eigenvector solution

-
- Solve $P^T y = y$
 - With y stochastic vector
 - PageRank original algorithm uses the **power method**
 - $y(k) = P^T y(k-1)$, with any starting vector $y^T(0)$
 - Rewrites as
 - $y(k) = \alpha P^T y(k-1) + (1-\alpha)v$
 - $y(k) = \alpha P^T y(k-1) + (\alpha a^T y(k-1) + (1-\alpha))v$
 - Note
 - Computations can be performed on the sparse matrix P instead of the dense matrix P^T

Diapositive 165

p1

pg; 30/01/2011

□ Check

- Irreducibility guarantees the convergence P'' being stochastic, its dominant eigenvalue $\lambda_1 = 1$
- P'' being primitive, the eigenvector associated to λ_1 (PageRank vector) is unique

□ Rate of convergence

- For the web graph, convergence is governed by α
- The rate of convergence is the rate at which $\alpha^t \rightarrow 0$
 - Initial paper by Brin & Page uses $\alpha = 0.85$ and 50 to 100 iterations

Two formulations of the PageRank problem

2. Linear system formulation

- Solve $(I - P^T)y = 0$ with $y^T \mathbf{1} = 1$
 - This can be rewritten as a function of P directly
- Solve $(I - \alpha P^T)y = v$ with $y^T \mathbf{1} = 1$, and v a stochastic vector
- $(I - \alpha P^T)$ has some interesting properties
 - It is non singular
 - Column sums are $1 - \alpha$ for non dangling nodes or 1 for dangling nodes
- This formulation opens the way to iterative methods for solving linear equations
 - e.g. Jacobi, Gauss-Seidel, successive over relaxation methods

Jacobi method for solving linear systems

- Let the linear system
 - $Ax = b$
- Decompose A into
 - $A = D + R$ with D the diagonal of A
 - R diagonal is 0
- $Ax = b$ writes $Dx = b - Rx$
- If D invertible, Jacobi method solves the linear equation by
 - Matrix form: $x(k+1) = D^{-1}(b - Rx(k))$
 - Element form: $x_i(k+1) = \frac{1}{a_{ii}} (b_i - \sum_{j \neq i} a_{ij}x_j(k))$
- Converges if A is strictly diagonally dominant
 - i.e. $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ (strict row diagonal dominance)
 - *Th Levy-Desplanques*
 - a square matrix with a diagonal strictly dominant is invertible

-
- PageRank with Jacobi
 - Algorithm
 - Start with an arbitrary vector $y(0)$
 - Iterate
 - $y(k + 1) = (I - \alpha P_{diag})^{-1}(v + \alpha P_{offdiag}y(k))$

☐ Personalization vector

- Any probability vector v with positive elements can be used
- $v = \frac{1}{n} \mathbf{1}$ uniform teleportation
- Can be used to
 - ☐ personalize the search
 - ☐ Control spamming (link farms)


Convergence rate of PageRank

□ Theorem (Bianchini et al. 2005)

- Let y^* the stationary vector of PageRank, $e_{rel}(t) = \frac{\|y^* - y(t)\|_1}{\|y^*\|_1}$ the 1 norm of the relative error in the computation of PageRank at time t , then
$$e_{rel}(t) \leq \alpha^t e_{rel}(0)$$
- If there is no dangling page, then there exists $v \geq 0$ and $v = P''v$, s.t. the equality holds

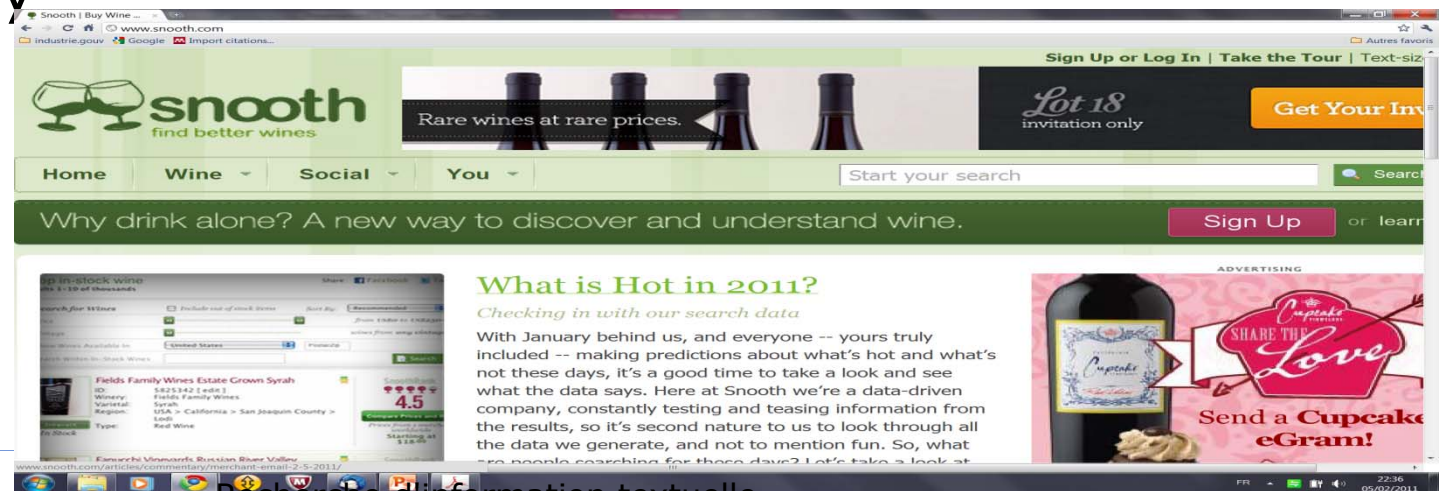
HITS (Kleinberg 98)

□ Hub



Website	Cellarer rank	Valuation	PageRank	Monthly traffic
Snooth	1	\$244 803	6	332962
Wine Spectator	2	\$88 238	7	87712
Wine Enthusiast	3	\$29 072	6	30938
Dr. Vino	4	\$28 690	6	30403
Cork'd	5	\$27 280	6	28429
Wine lovers page	6	\$25 812	6	26374
The Winedoctor	7	\$25 384	6	25775
eRobertParker	8	\$24 817	6	24981
Decanter	9	\$22 895	6	22291

□ Authority



Sign Up or Log In | Take the Tour | Text-size

Rare wines at rare prices.

Lot 18 invitation only

Get Your Inv

Home Wine Social You

Start your search

Why drink alone? A new way to discover and understand wine.

Sign Up or learn

What is Hot in 2011?

Checking in with our search data

With January behind us, and everyone -- yours truly included -- making predictions about what's hot and what's not these days, it's a good time to take a look and see what the data says. Here at Snooth we're a data-driven company, constantly testing and teasing information from the results, so it's second nature to us to look through all the data we generate, and not to mention fun. So, what are people searching for these days? Let's take a look at

Fields Family Wines Estate Grown Syrah

4.5

Send a Cupcake eGram!

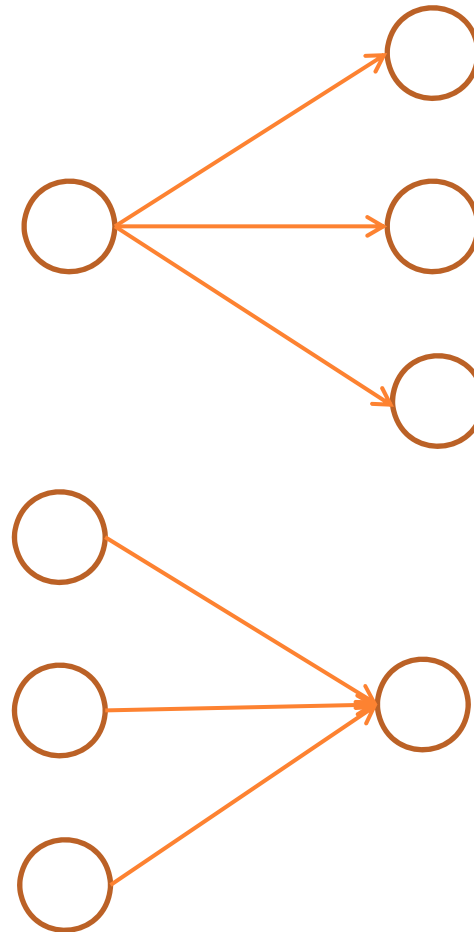
Recherche d'information textuelle

□ Hub

- Important reference pages
- Points to good authority pages
- **Hub score** of a page: sum of the authority scores of its children

□ Authority

- Important reference pages for a topic
- Pointed by good hub pages
- **Authority score** of a page: sum of the hub scores of its parents



HITS - Algorithm

- Input
 - Web subgraph relative to a query
 - The subgraph is composed of the retrieved documents + the linked (in and out) web documents
 - Only a part of the linked document is considered (e.g. 100)
- Output
 - Authority and hub scores, $h()$ and $a()$ for all pages in the graph
- Algorithm
 - Initialize
 - $a(v) = 1, h(v) = 1, \forall v$ (any positive vector value will do)
 - Repeat
 - $h_t(v) = \sum_{w \rightarrow v} a_{t-1}(w)$
 - $a_t(v) = \sum_{w \rightarrow v} h_{t-1}(w)$
 - Normalize h and a
 - $a_t = a_t / \|a_t\|$
 - $h_t = h_t / \|h_t\|$
 - Until convergence
 - Return the two lists

HITS algorithm (followed)

- For the subgraph, let
 - \mathbf{h} : vector of page hubs
 - \mathbf{a} : vector of page authorities
 - A the adjacency matrix
- In matricial form, the algorithm writes

$$\begin{cases} \mathbf{h}_t = A\mathbf{a}_{t-1} \\ \mathbf{a}_t = A^T\mathbf{h}_{t-1} \end{cases} + \text{normalization}$$

or

$$\begin{cases} \mathbf{h}_t = AA^T\mathbf{h}_{t-1} \\ \mathbf{a}_t = A^T A\mathbf{a}_{t-1} \end{cases} + \text{normalization}$$

□ Matrices

- $A^T A$ is called the authority matrix
 - Determines authority score
- $A A^T$ is called the hub matrix
 - Determines hub score
- Both are symmetric, positive semi-definite
 - The dominant eigenvalue λ_1 is unique

□ Algorithm

- The update algorithm is the power method for matrices $A A^T$ and $A^T A$
- It converges towards one of the dominant eigenvector associated to $A A^T$ and $A^T A$

□ Convergence

- Although λ_1 is unique, it may have multiple eigenvectors, so that the convergence will depend on the initial vectors $\mathbf{a}(0)$ and $\mathbf{h}(0)$.
- A trick similar to PageRank can be used to make the matrices primitive and converge to a unique eigenvector:
 - Replace AA^T with $\alpha AA^T + (1 - \alpha)\mathbf{1}.v^T$, $0 < \alpha \leq 1$ and v a stochastic vector
 - Same thing with $A^T A$

□ Example (Lanville – Mever 2006)

$$\mathbf{L} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 5 & 6 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 10 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}.$$

The respective authority and hub matrices are:

$$\mathbf{L}^T \mathbf{L} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 5 & 6 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 10 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad \text{and} \quad \mathbf{L} \mathbf{L}^T = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 5 & 6 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 10 \end{matrix} & \begin{pmatrix} 2 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{matrix}.$$

The normalized principal eigenvectors with the authority scores \mathbf{x} and hub scores \mathbf{y} are:

$$\mathbf{x}^T = (0 \quad 0 \quad .3660 \quad .1340 \quad .5 \quad 0) \quad \text{and} \\ \mathbf{y}^T = (.3660 \quad 0 \quad .2113 \quad 0 \quad .2113 \quad .2113).$$

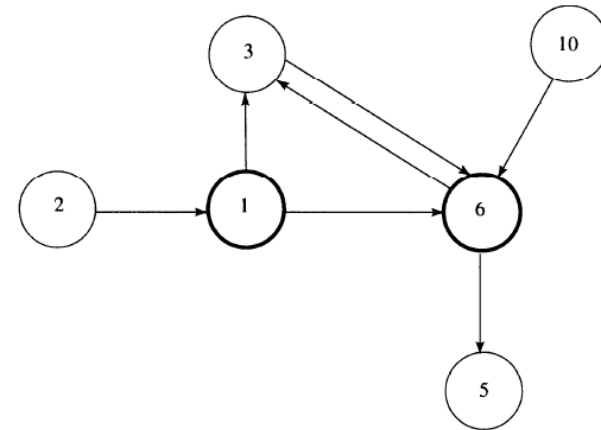


Figure 12.1 Neighborhood graph N for pages 1 and 6

□ Matrices

- A symmetric matrix B is *positive semi-definite* if
 - For all non zero vector x , $x^T B x \geq 0$
 - Or equivalently all eigenvalues are ≥ 0
- A matrix B is *positive definite* if
 - \geq is replaced by $>$



Ranking for information retrieval

Ranking tasks

- ☐ Many IR problems require combining a “large” number of features in order to rank items
- ☐ Metasearch
 - Combine the scores of several search engines
- ☐ Web search
 - Combine a large number of heterogeneous features
- ☐ Summarization
 - Find the most representative sentences in a document
- ☐ Collaborative filtering
 - Find the most relevant items for a user based on user similarities
- ☐

Machine learning for ranking items

- Machine learning offers a variety of generic methods for combining features “optimally” in order to rank items
- Classical setting
 - Learn a score function
 - Classification, Regression, Density estimation
 - Rank items according to their score
- This is only indirectly related to the ranking goal
 - Prediction errors are loosely correlated to ranking
 - e.g. classification/ regression errors are not important if ranking is correct
 - Learning samples are often biased towards negative examples
 - e.g. Ad-hoc IR
 - Solutions : e.g. sub sampling – do not exploit all available data
- Learning to rank aims at directly optimizing a ranking function
 - It allows combining heterogeneous features, scores, preference relations, ...

Learning to rank evolution

- Machine learning
 - Starts around 2000
 - Rankboost, Ranksvm etc (2002)
 - Today
 - (too) Many algorithms, objective functions, theory, bounds, etc
- IR
 - Early applications around 2006
 - Metasearch 2003
 - SIGIR 2007: 1 session, 1 Wshop 100 participants
 - SIGIR 2008: 2 sessions, 1 Wshop
 - Challenges: IR objective functions, scaling

Classical setting

- Learn a total order on a set of items X
 - This order will allow comparing all couples of items in X
 - Given this total order, any subset of X can be ordered

- For ad-hoc IR
 - X is a set of couples (document, query)
 - The total order is the natural order of document relevance given a query

How to learn?

- Simplest method
 - Learn to order pairs of examples
 - An error occurs when two elements are incorrectly ordered
 - Only the relative scores in a pair are important

- The training set consists in a set of ordered item pairs
 - Items for ad-hoc: $(d, q, y(d,q))$ where y is the relevance judgement
 - Not all pairs need be ordered: only a small subset of the examples is needed
 - This will provide a partial order on the elements of X

- A **ranking function** f will be learned on this partially ordered set
 - f scores each item : $f(d,q)$
 - f will be used on new items (e.g. $(d, q_{\text{new}}), (d_{\text{new}}, q_{\text{new}})$)
 - It will then allow us to extend the partial order to a total order on all the elements

Notations

- An item x in X (e.g. (d, q)) will be represented by a real vector

$$x = (x_1, x_2, \dots, x_n)$$

- Examples

- Metasearch

- For a given query q , the x_i s will be the scores of different search engines on a document d

- Ad hoc search

- For a given query q , the x_i s will be different scores (PageRank, Okapi, etc)

- The ranking function will be a linear combination of x features

$$F_w(x) = \sum_{i=1}^n w_i x_i$$

- $w = (w_1, \dots, w_n)$ are parameters to be learned
- In the following, one considers only bipartite ranking

Ranking loss

- Early algorithms use for ranking loss the number of errors on crucial pairs

- For a query q

$$PairLoss(f) = \sum_{\substack{(x,x') \in Training\ set \\ x < x'}} [|f(x') < f(x)|]$$

With $[|f(x') < f(x)|] = 1$ if $f_w(x) > f_w(x')$, 0 otherwise

- For multiple queries

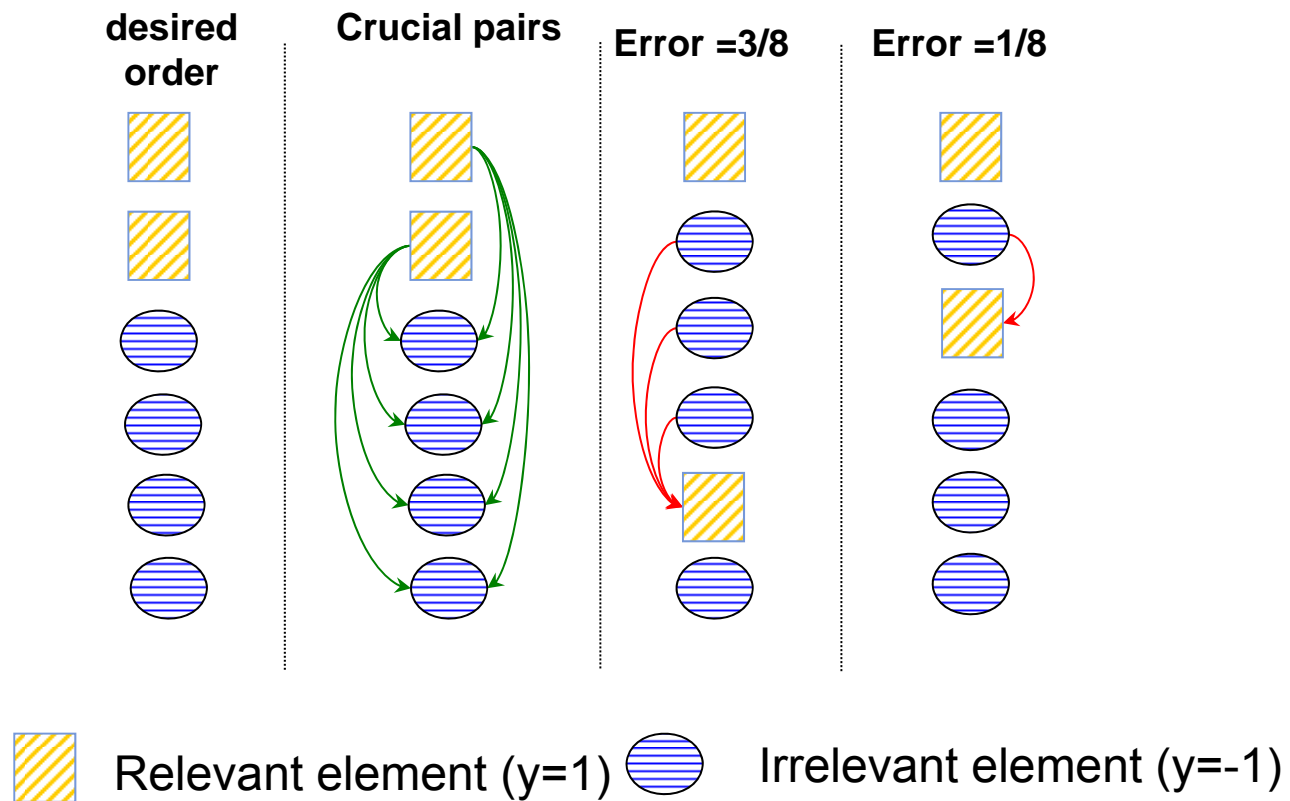
- Example

$$PairLoss(f) = \sum_q \sum_{\substack{x \in Training\ set \\ x < x'}} [|f(x') < f(x)|]$$

- PairLoss

- Measures how well the ranking function f_w satisfies the order requirement
- Estimates the probability $P(f(x') > f(x))$: AUC criterion

Example



Ranking as classification

- Ranking is often formulated as a classification problem on crucial pairs
- For each couple (x, x') , two examples are created
 - $(x' - x, +1)$ and $(x - x', -1)$
 - Remember f is linear
- Any classification algorithm can be used on this new set
 - Same number of $+$ and $-$ examples in the training set
- PairLoss is also valid for multi-valued judgments

Ranking (aka pair classification) vs direct classification of examples

□ Classification

- Predicts if a document is relevant (+1) or non relevant (-1) $P(C|x)$
- Does not take into account the ranking of items
- Minimizes a classification or regression error

□ Ranking

- Considers uniquely the ranking of items $P(x \prec x' | x, x')$
- Minimizes the number of incorrectly ordered couples
- The scale of scores is not important

PairLoss optimization

- PairLoss is non differentiable
 - Algorithms optimize smooth approximations of PairLoss: Hinge loss, Exponential loss, logistic loss,...

- Examples

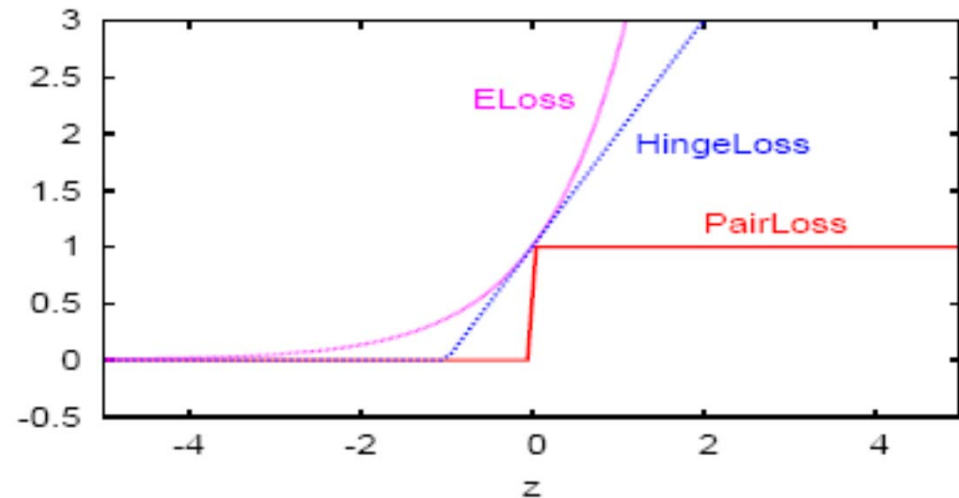
- Hinge loss

$$HLoss(f) = \sum_{\substack{(x, x') \in X^2 \\ x \prec x'}} [1 - f(x') + f(x)]_+$$

$$[z]_+ = \max(0, z)$$

- Exponential loss

$$ELoss(f) = \sum_{\substack{(x, x') \in X^2 \\ x \prec x'}} e^{f_w(x) - f_w(x')}$$



RankSVM (Herbrich et al. 2000)

- Direct adaptation of SVM to the ranking of items
 - Ranking function is linear $f = w \cdot x$
- The ranking problem is transformed into a classification problem
 - For all crucial pair (x, x') with $x < x'$, one wants $w \cdot (x' - x) > 0$
 - Training set $\{(x_i, x'_i)\}_{i=1..N}$ with $x < x'$, RankSVM optimizes

$$\min_w \sum_{i=1}^N [1 - f(x'_i) + f(x_i)] + \lambda \|w\|^2$$

$\lambda \|w\|^2$ is a regularization term

- Quadratic optimization problem
 - Can be solved using classical SVM optimization algorithms

RankBoost (Freund 2003)

□ Based on Adaboost

- Iteratively build a linear combination of base functions $f(x) = \sum_{t=1}^T \alpha_t g_t(x)$
- So as to minimize the ELoss

$$ELoss(f) = \sum_{x \in X_{Tr}^-, x' \in X_{Tr}^+} D(x, x') \exp(f(x) - f(x'))$$

- At each iteration the distribution D_t on crucial pairs is adapted in order to focus on “difficult” pairs
- The corresponding ELoss is minimized in order to learn the t^{th} base function f_t

$$\text{Let } f_t = \sum_{k=1}^t \alpha_k g_k$$

$$ELoss(f_t) = \sum_{x \in X_{Tr}^-, x' \in X_{Tr}^+} D_t(x, x') \exp(\alpha_t (f_t(x) - f_t(x')))$$

- Stops at iteration T

□ Note

- $\text{PairLoss}(f_T) < \prod_{t=1..T} \text{ELoss}(f_t)$
- $\text{PairLoss}(f_T)$ can be minimized by minimizing iteratively $\text{ELoss}(f_t)$

Remarks

- Do it simpler
 - ELoss can be optimized directly using gradient descent to learn the weights of f (e.g. Vittaut et al. 2005)

- Complexity
 - PairLoss and ELoss are quadratic in the number of examples
 - Sum over all crucial pairs
 - RankSVM is also quadratic in the number of items
 - ELoss complexity can be made linear in the number of examples

$$ELoss(f) = \left(\sum_{x \in assessments(1)} e^{f_w(x)} \right) \left(\sum_{x' \in assessments(-1)} e^{-f_w(x')} \right)$$

- All these training criteria are micro-criteria (all pairs weight the same) - Macro versions also available

Adapting ranking to IR

- Ranking with AUC behaves much better than e.g. classification for IR applications, but ...we could still do better
- AUC
 - All errors have the same weight
 - For IR applications we are rather interested into the top results
- Rather optimize directly IR measures: MAP, nDCG, ...
 - no pair decomposition like for AUC: difficult to optimize directly
- Lot of papers on this topic over last years:
 - Optimize AUC using gradient methods using a validation set and MAP or nDCG to stop learning
 - Adapt AUC for considering top of the list
 - Learn approximations of RI measures
 - E.g. approximations of MAP, nDCG, ...
 - Use distances on lists rather than on pairs

Bibliographie

□ Ouvrages généraux

- Baeza-Yates R, Ribeiro-Neto B., 1999, Modern Information Retrieval, Addison-Wesley
- Manning D., Schütze H., 1999, Foundations of statistical natural language processing, MIT press
- **Christopher D. Manning, Prabhakar Raghavan and Heinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.**
- Amini M., Gaussier E., Recherche d'information, Eyrolles 2012

□ Articles

- Hiemstra. D., Using language models for information retrieval. PhD thesis, University of Twente, 2001.
- Hofmann T., Probabilistic latent semantic indexing, SIGIR 1999
- Kazai G. Lalmas M., Inex 2005 evaluation metrics, Inex proceedings.
- Kazai G., Lalmas M., de Vries A.P., 2004, the overlap problem in content-oriented XML retrieval evaluation, Sigir'04
- Mass Y. and Mandelbrod. M., 2005, Component ranking and automatic query renement, In INEX 2004, Lecture Notes in Computer Science, volume 3493, pages 154 - 157. Springer-Verlag 2005.
- Miller D.H, Leek T., Schwartz R., 1999, A hidden Markov model information retrieval system, Sigir'99
- Ogilvie P. Callan J., 2003, Combining document representations for known item search, Sigir'03
- Piwowarski B. Gallinari P, A bayesian framework for xml information retrieval : Searching and learning with the inex collection. *Information Retrieval* , 8:655{681, 2005.
- Piwowarski B. Gallinari P. Dupret G., 2005, Precision recall with user modelling : Application to xml retrieval. To appear Transactions on Information Systems.
- Robertson S., Zaragoza H., Taylor M., 2004, Simple BM extension to multiple weighted fields, CIKM'04
- Sigurbjornsson B., Kamps J., and Rijke M. University of amsterdam at inex 2005. In Pre- Proceedings of the 4th workshop of the initiative for the evaluation of XML retrieval (INEX), pages 84-94, 2005.
- Vittaut N., Gallinari P., Machine Learning Ranking for Structured Information Retrieval, in Proc. ECIR'06
- Zaragoza H., Crasswell N., Taylor M. Saria S. Robertson S., Microsoft Cambridge at TREC-13 : Web and Hard tracks, NIST