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
- Corpus et documents structurés

Introduction

Problèmes de l'accès à l 'information

- Représentation - indexation, cf W3C, MPEG7
  - non structuré (texte, image), semi structuré (BD, balises, métadescription)
- Techniques d'accès - modèle de recherche
  - présenter un ensemble de documents selon un ordre de pertinence, présenter un unique document, ..
- Interaction utilisateur
  - feedback, recherche interactive
- Adaptation à l 'utilisateur
  - modéliser le comportement, mémoire des demandes,
- Stockage
  - quantité de données (tera), stockage distribué

Diversité des sources d ’information

- Texte
  - articles (pdf, ps, …)
  - livres
  - pages HTML, XML
- Images, Video, Son, Musique
- Web (pages, sites, blogs etc), Messageries - fils de discussion, etc

Diversité des demandes d'accès à l 'information

- Consultation (browsing)
- Requêtes booléennes
- Recherche par le contenu
- Recherche interactive
- Recherche automatique (e.g. robots)
- Recherche BD

Information textuelle

- Bases de données textes
  - Bibliothèques électroniques
  - Journaux (Le Monde, Wall Street Journal, ...)
  - Bases d'articles …
- Web
  - Moyen d'accès, e.g. BD textes
  - Source d'information dynamique : sites, pages, ...
  - 95 % de l'information présente sur le Web est sous forme textuelle
- Information majoritairement non structurée, mais structures exploitables (HTML, SGML, XML), hiérarchies, ...

Lois de puissance

- **Loi de Zipf**
  - Caractérise la fréquence d'occurrence en fonction du rang
  - Empiriquement : fréquence.rang = cte
  - Le 1er mot est environ 2 fois plus fréquent que le 2nd qui est 2 fois plus fréquent que le 3e 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 :

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

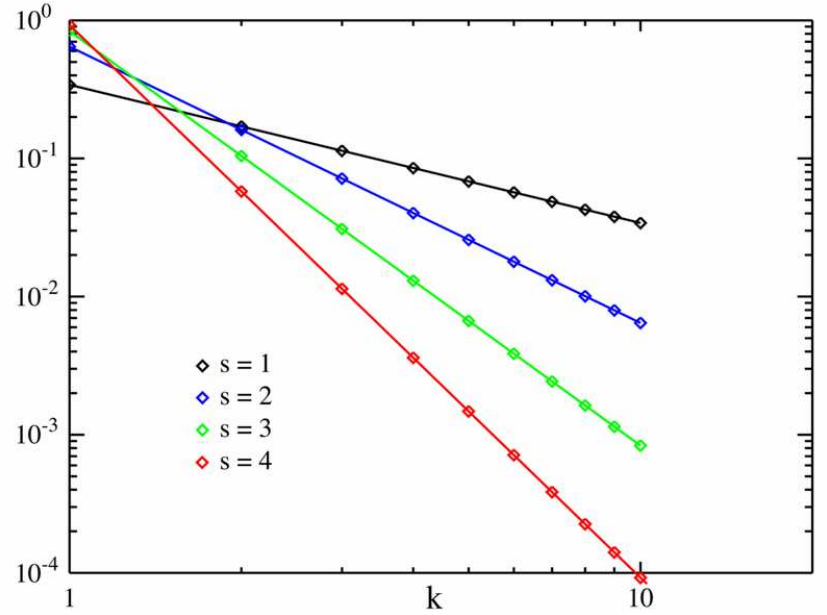
$$\log(f) = -\log r^s - \log \sum_{n=1}^{N} 1/n^s$$

r : rang

N : taille du corpus

s : paramètre qui dépend du corpus

En anglais s ≈ 1, i.e. *f.r* ≈ 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

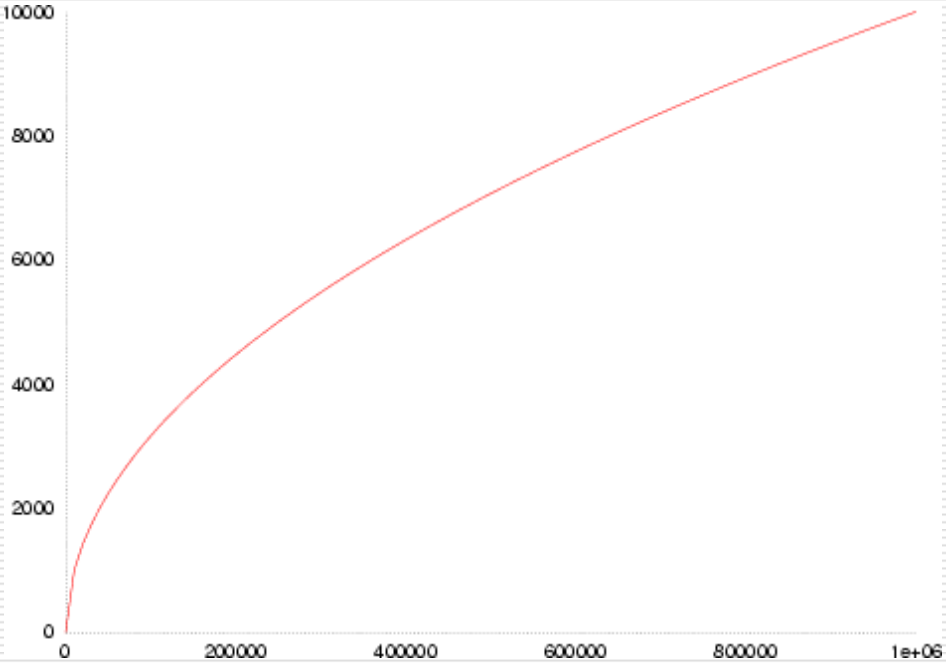
n : taille du texte

K,B paramètres dépendant du texte

Anglais

K entre 10 et 100 et B 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)

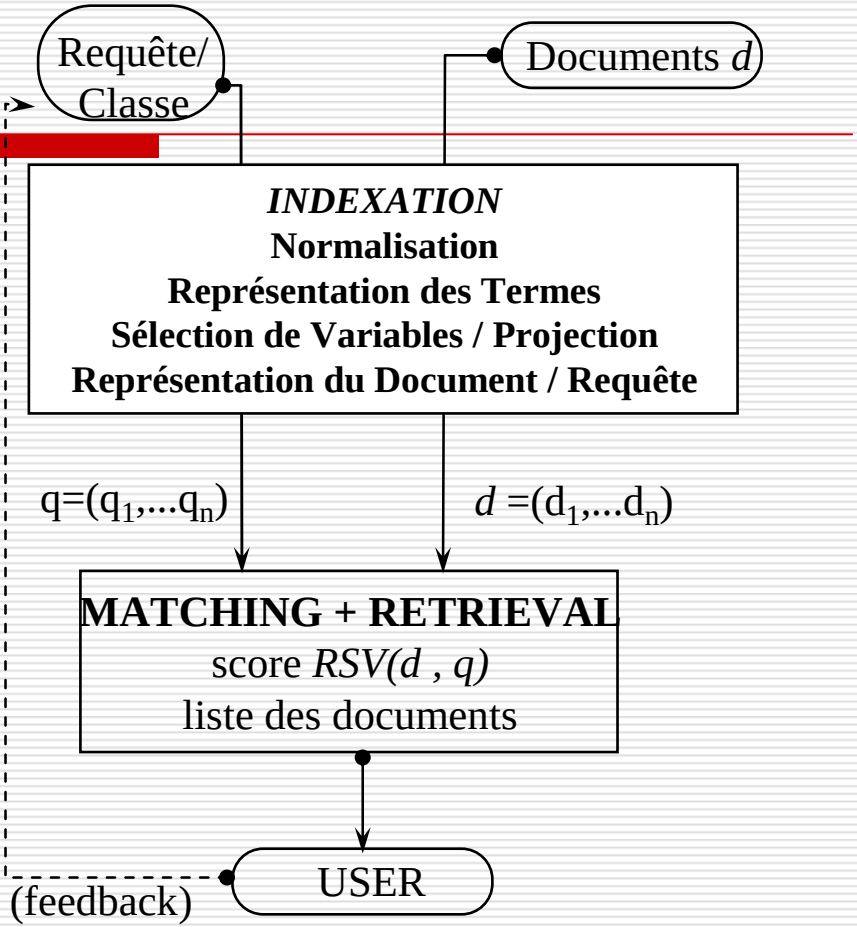
Exemples de tâches

- 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, ...) ....

Recherche d’information textuelle

Recherche d'information
Requêtes ouvertes

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



Text Retrieval Conferences-TREC TREC 2006

- **Blog Track**: 2006. explore information seeking behavior in the blogosphere.
- **Enterprise Track**: 2005. study enterprise search: satisfying a user who is searching the data of an organization to complete some task.
- **Genomics Track**: 2003. study retrieval tasks in a specific domain(include not just gene sequences but also supporting documentation such as research papers, lab reports, etc.)
- **Legal Track**: 2006. develop search technology that meets the needs of lawyers to engage in effective discovery in digital document collections.
- **Question Answering Track**: take a step closer to information retrieval rather than document retrieval.
- **SPAM Track**: 2005. standard evaluation of current and proposed spam filtering approaches, laying the foundation for the evaluation of more general email filtering and retrieval tasks.
- **Terabyte Track**: 2004. investigate whether/how the IR community can scale traditional IR test-collection-based evaluation to significantly larger document collections than those currently used in TREC.

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 i**nvestigate 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.

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 : 10^3, 10^6, 10^9
- Catégorie : liste de mots clé

RD : Prétraitement et représentation des textes : le processus d'indexation

- Analyse lexicale
  - Conversion du texte en un ensemble de termes
    - Unité lexicale ou radical
    - Espaces, chiffres, ponctuations, etc
- 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 dépendant 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
  whomever
  whose
  why
  with
  within
  without
  would
  yes
  your
  yours
  yourself
  yourselves

Prétraitement et représentation des textes *(2)*

- *Normalisation (lemmatisation)*
  - Utilisation d'une forme canonique pour représenter les variantes morphologiques d'un mot
  - e.g. dynamic, dynamics, dynamically, ...seront représentés par un même mot, naviguer, naviguant, navire .....idem
  - 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 établi 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)
- *Regroupement*
  - de mots similaires au sens d'un critère numérique

Prétraitement et représentation des textes *(3)*

- 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 (4) : Implémentation des index

- 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 le 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

Prétraitement et représentation des textes (5)

- 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, ...)

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 φ_i, i = 1...n i.e. termes sélectionnés pré-traités
- Représentation des documents - requêtes : vecteur de poids dans l'espace des caractéristiques

  document: $d=(x_0,..., x_{n-1})$

  requête: $q =(y_0,, ..., y_{n-1})$
- 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
  - 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 - inverse document frequency (tf-idf)

$$x_i = tf(\varphi_i, d) idf(\varphi_i)$$

  - tf(φ_i,d): # occurrences de φ_i dans le document d
  - df(φ_i) : # documents contenant φ_i
  - idf(φ_i) : fréquence inverse
  - idf décroit vers 0 si φ_i apparaît dans tous les documents

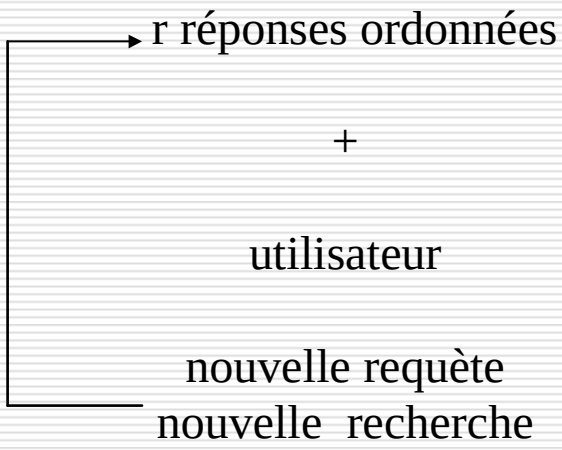
$$idf(\varphi_i) = \log\left(\frac{1+N}{1+df(\varphi_i)}\right)$$

- Mesure de similarité entre q et d (e.g. Salton)
  - Nombreuses autres pondérations et similarités.

$$RSV_{\cos}(d,q) = \frac{\sum_{i=0}^{n-1} x_i y_i}{\sqrt{\sum_{i=0}^{n-1} x_i^2 \sum_{i=0}^{n-1} y_i^2}}$$

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

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

- 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 :

$$\underline{q'} = \alpha \frac{\underline{q}}{\|\underline{q}\|} + \frac{\beta}{|D_r^{rel}|} \sum_{d_j \in D_r^{rel}} \frac{\underline{d_j}}{\|\underline{d_j}\|} - \frac{\gamma}{|D_r^{nonrel}|} \sum_{d_j \in D_r^{nonrel}} \frac{\underline{d_j}}{\|\underline{d_j}\|}$$

  - améliorations allant de 20% a 80 % par rapport à sans RF.
  - Différentes variantes :
    - considérer $D_r^{nonrel} = \emptyset$
    - 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 …

Recherche interactive – Exemple 2

- Reestimation de Robertson et Sparck-Jones (1976) (codage binaire)

$$RSV\ (q,d) = \sum_{i=0}^{n-1} x_i y_i$$

$$x_i = \begin{cases} 1 \text{ si } \varphi_i \in d_j \\ 0 \text{ sinon} \end{cases}$$

$$y_i = \begin{cases} \log \dfrac{p_i(1-q_i)}{q_i(1-p_i)} \text{ si } \varphi_i \in q \\ 0 \text{ sinon} \end{cases}$$

- avec :

$$p_i = \frac{\#\text{ documents } d \text{ dans } D_r^{rel} \text{ contenant } \varphi_i}{\#\text{ documents } d_j \text{ dans } D_r^{rel}}$$

$$q_i = \frac{\#\text{ documents } d \text{ dans } D_r^{nonrel} \text{ contenant } \varphi_i}{\#\text{ documents } d_j \text{ dans } D_r^{nonrel}}$$

p_i : P(doc pertinent contient le terme φ_i de la requête)

q_i : P(doc non pertinent contient le terme φ_i de la requête)

Recherche interactive

- Justification Robertson et Sparck-Jones

$d = (x_1, \ldots, x_n)$, x_i = 0 ou 1 (présence / absence du terme i dans d)

$p_i = P(x_i = 1 / R)$

$q_i = P(x_i = 1 / \neg R)$

$$p(d / R) = \prod_i p_i^{x_i} (1 - p_i)^{1 - x_i}$$

$$p(d / \neg R) = \prod_i q_i^{x_i} (1 - q_i)^{1 - x_i}$$

fonction de décision : $$\log \frac{p(d/R)}{p(d/\neg R)} = \sum_i \log \frac{p_i(1-q_i)}{q_i(1-p_i)} + cte$$

Modèle probabiliste

Modèle probabiliste

- Probability Ranking Principle (Robertson 77)
  - présenter les documents à l 'utilisateur selon l 'ordre décroissant de leur probabilité de pertinence P(R/q,d) est optimal (pour le coût, la précision, le rappel..)
- 2 événements
  - R : d est pertinent pour q
  - ¬R : d n'est pas pertinent pour q

$$P(R/d) = \frac{P(d/R)P(R)}{P(d)}$$

- Calcul de P(R/d)

$$\log \frac{P(R/d)}{P(\neg R/d)} = \log \frac{P(d/R)}{P(d/\neg R)} + \log \frac{P(R)}{P(\neg R)}$$

$$RSV(d,q) = \log \frac{P(d/R)}{P(d/\neg R)}$$

- Indépendance des caractéristiques

$$P(d/R) = \prod_i p(\varphi_i/R)$$

$$RSV(d,q) = \sum_i \log \frac{p(\varphi_i/R)}{p(\varphi_i/\neg R)}$$

Modèle probabiliste

- Ne pas tenir compte des attributs absents :

$$RSV(d,q) = \sum_i \left(\log \frac{p(\varphi_i / R)}{p(\varphi_i / \neg R)} - \log \frac{p(\varphi_i = 0 / R)}{p(\varphi_i = 0 / \neg R)} \right)$$

$$RSV(d,q) = \sum_{\varphi_i \in q \cap d} \log \frac{p_i(1-q_i)}{q_i(1-p_i)}$$

$$p_i = P(\varphi_i / R)$$

$$q_i = P(\varphi_i / \neg R)$$

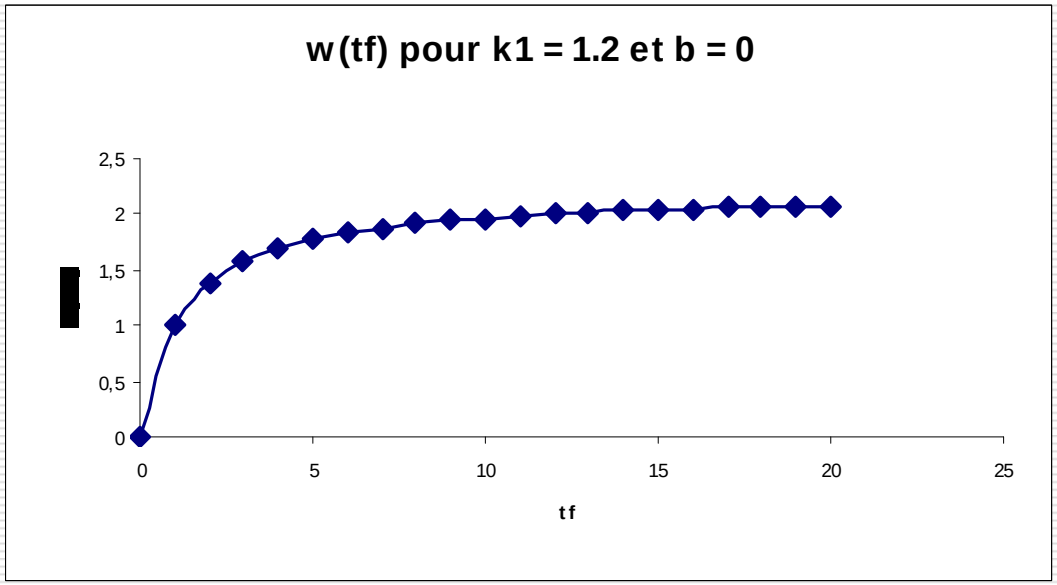
- Nombreuses variantes / extensions¨Problèmes
  - longueur des documents (hypothèse implicite d'égale longueur)
  - expansion des requêtes
  - # doc pertinents considérés (e.g. cas recherche on line <> off line
  - cooccurrence de termes, prise en compte de « phrases » ...

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 langage

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

- Variables
  - d : document que l'utilisateur a en tête
  - t_i : i^{eme} terme de la requête
  - $I_i \in \{0,1\}$ importance du i^e terme de la requête
    1 : important, 0 : pas important
- Considérons une requête de n termes t_1, …, t_n
  - Les documents seront ordonnés selon la pertinence du document pour la requête : $P(d|t_1,..,t_n)$
  - Score d'un document : $P(t_1,..,t_n|d)$
  - On a alors un modèle statistique par document

- Hypothèse : indépendence des termes de la requête conditionnellement à leur importance
- La fonction de score devient

$$P(t_1,..,t_n|d) = \prod_{i=1}^{n} \sum_{k=0,1} p(I_i = k)\, p(t_i / I_i = k, d)$$

$$P(t_1,...,t_n|d) = \prod_{i=1}^{n} (1-\lambda_i)\, p(t_i) + \lambda_i\, p(t_i / d)$$

Avec $p(t/d) = p(t/I = 1,d)$ et $p(t) = p(t/I=0)$

Modèles de langage – Apprentissage

- Différents estimateurs possibles, le plus courant : maximum de vraisemblance
- exemple : $p(d) = \frac{1}{\#documents}$

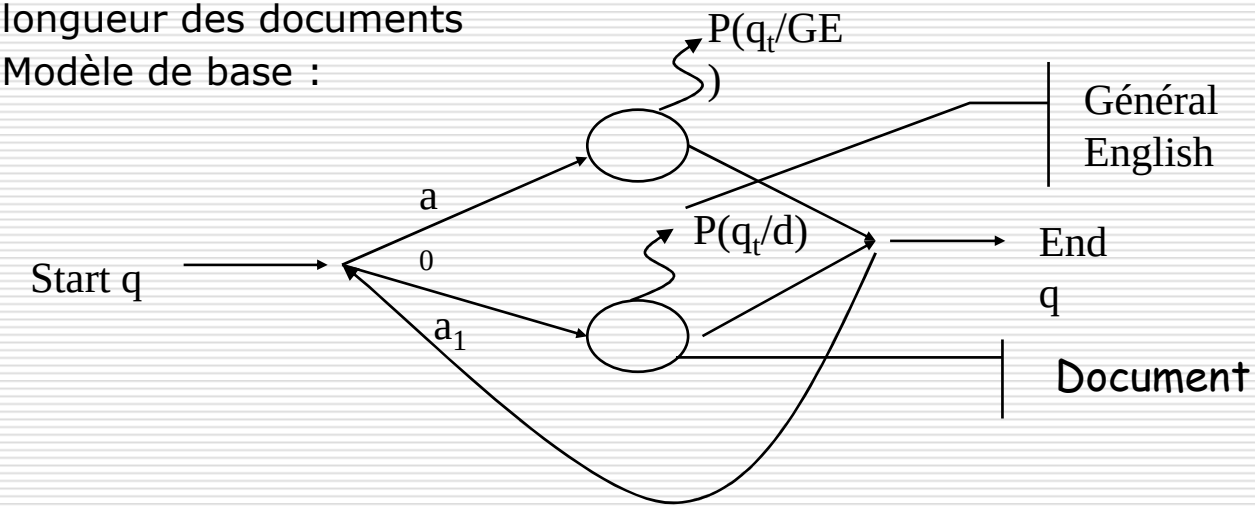
$$p(t_i / I_i = 1, d) = \frac{tf(t_i, d)}{\sum_t tf(t, d)} \qquad p(t_i / I_i = 0) = \frac{\sum_{d'} tf(t_i, d')}{\sum_{d', t} tf(t, d')}$$

- $\lambda_i = p(I_i = 1)$

Les λ_i sont estimés par EM en maximisant la vraisemblance des documents pertinents et des requêtes associées.

HMMs (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 :

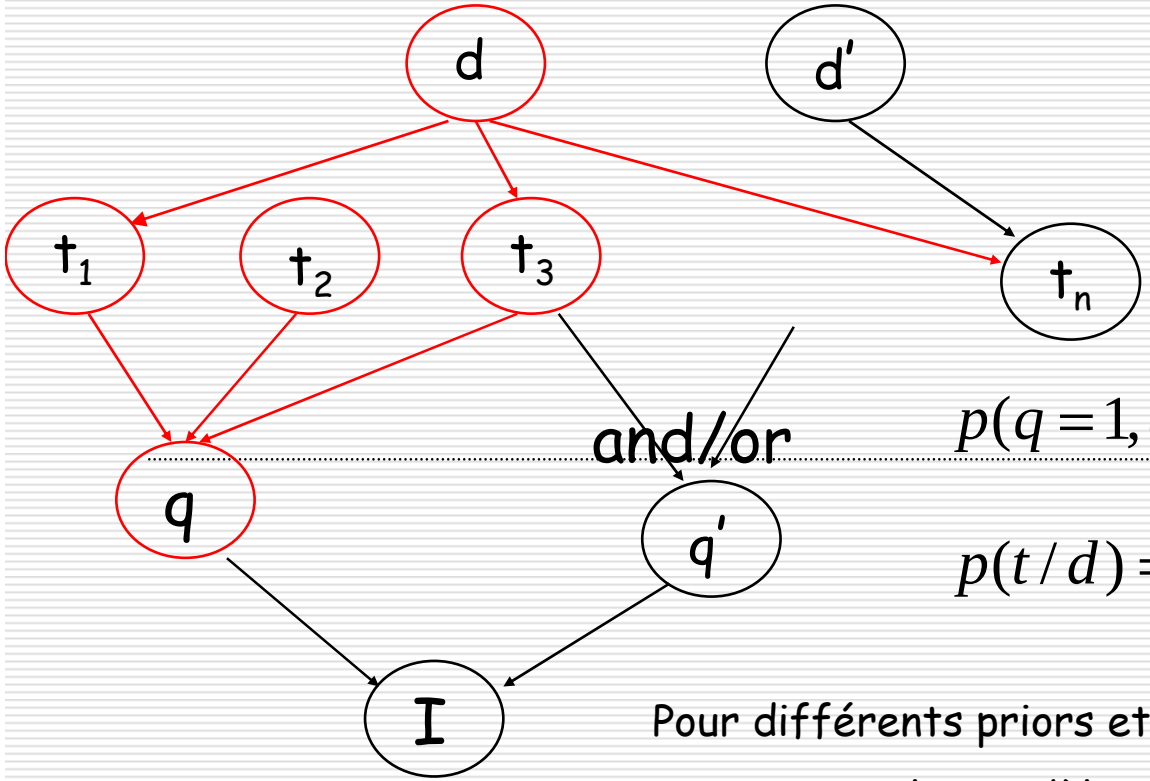


$$p(q/d\ relevant) = \prod_t (a_0\, p(q_t/GE) + a_1\, p(q_t/d))$$

Réseaux Bayesiens

Bayesian Inference Network (Turtle, Croft 91 – Inquery)

- variables binaires, Relevance : p(q = 1, d = 1)



d = 1 evenement "on observe d"

q = 1 la requête est satisfaite

$$p(q=1,d=1)=\sum_{all\ t} p(q/t).p(t/d).p(d)$$

$$p(t/d)=\prod_{i/t_i=1} p(t_i/d)\prod_{i/t_i=0} p(\neg t_i/d)$$

Pour différents priors et P(node/parents)

on retrouve les modèles booleens ou probabilistes

Réduction de dimension
Latent Semantic Indexing (LSI/LSA)

- Décomposition en valeur singulière d'une matrice A m*n de rang r

$$A = U \Sigma V^T$$

Σ: diagonale, racines carrées des valeurs propres de AA^T

U : vecteurs propres de AA^T

V : vecteurs propres de A^TA

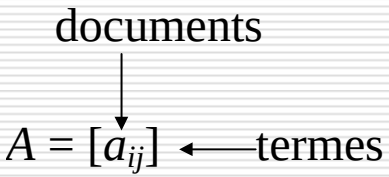
- Propriétés
  - Im (A) : span(u1, ..., ur), Ker (A) : span(vr+1, ..., vn)
  - Soit k < r

$$\underset{B/rk(B)=k}{Min} \|A-B\| = \|A-A_k\| \qquad A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T = U_k \Sigma_k V_k^T$$

  - U et V sont orthogonales

LSI

- Matrice terme documents : terme * docs



- a_{ij}= tf-idf, ou 0/1
- projection de la matrice terme - documents: A_k
- Interprétation
  - U : base des termes dans le nouvel espace
    - Vecteurs propres de la matrice de cooccurrence des termes
  - V : base des documents dans le nouvel espace
    - Vecteurs propres de la matrice de cooccurrence des documents

- Représentation d'une requête ou d'un document dans l'espace des termes :

$$q' = q^T U_k \Sigma_k^{-1}$$

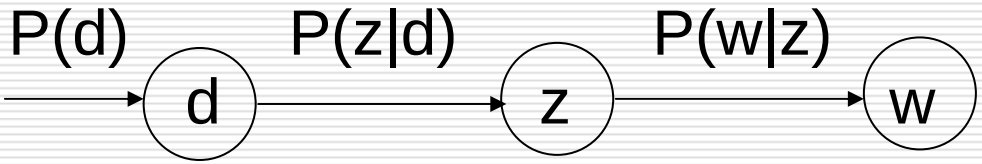
  - Les termes qui cooccurent fréquemment sont projetés au même « endroit »

- idem pour la projection dans l'espace des documents avec V
- Calcul de la similarité : $RSV_{\cos}(q', d')$

Probabilistic Latent Semantic Analysis - PLSA (Hofmann 99)

- Modélisation stochastique de LSA -
  - Modèle à variable latente
  - Une variable latente est associée à chaque occurrence d'un mot dans un document
  - Processus génératif
    - Choisir un document d ,P(d)
    - Choisir une classe latente z, P($z|d$)
    - Choisir un mot w suivant P($w|$ z)

Modèle PLSA



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

- Hypothèses
  - # valeurs de z est fixé
  - Indépendence des observations (d, w), i.e. sac de mots
  - Connaissant z, w ne dépend pas de d
- Apprentissage
  - MV et EM

Applications

- Extraction de concepts
  - Z_k : concept
  - $P(w_i|z_k)$ représentation du concept z_k
  - $P(z_k|d_i)$ importance du concept dans le document
  - Un concept sera commun à plusieurs mots
  - Un même mot peut être associé à différents concepts

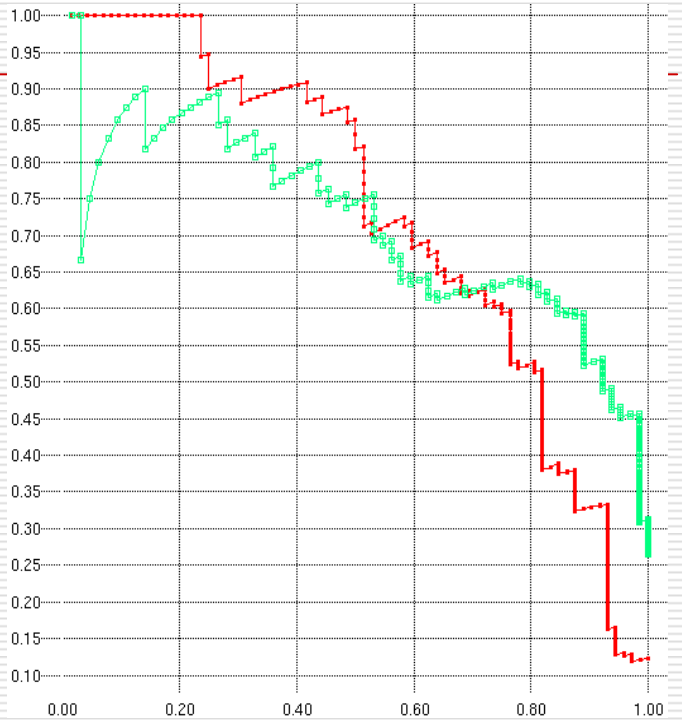
Applications (autres)

- Segmentation thématique
- Construction de hiérarchies de documents (# modèles plsa hiérarchiques)
- Recherche d'information
- Annotation d'images
  - Pour une image inconnue : P(w|image)

Evaluation en RD

- 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 ¬R pour une requête.

Evaluation en IR : mesures de rappel - précision



- ***Rappel*** à r:

$$r_r(q) = \frac{\#\text{ documents pertinents découverts}}{\#\text{ documents pertinents dans la collection}}$$

- ***Précision*** à r:

$$p_r(q) = \frac{\#\text{ documents pertinents découverts}}{\#\text{ documents decouverts}}$$

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

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 interpôléee 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 veleur de précision la plus forte rencontrée après le point i (interpolation)

Evaluation en RI

- Autres mesures d ’évaluation
  - Précision moyenne = 1/3*(précision(0.25) + précision(0.5) + précision(0.75))
  - F mesure $$F = \frac{2 * P * R}{P + R}$$
  - etc

Recherche Web

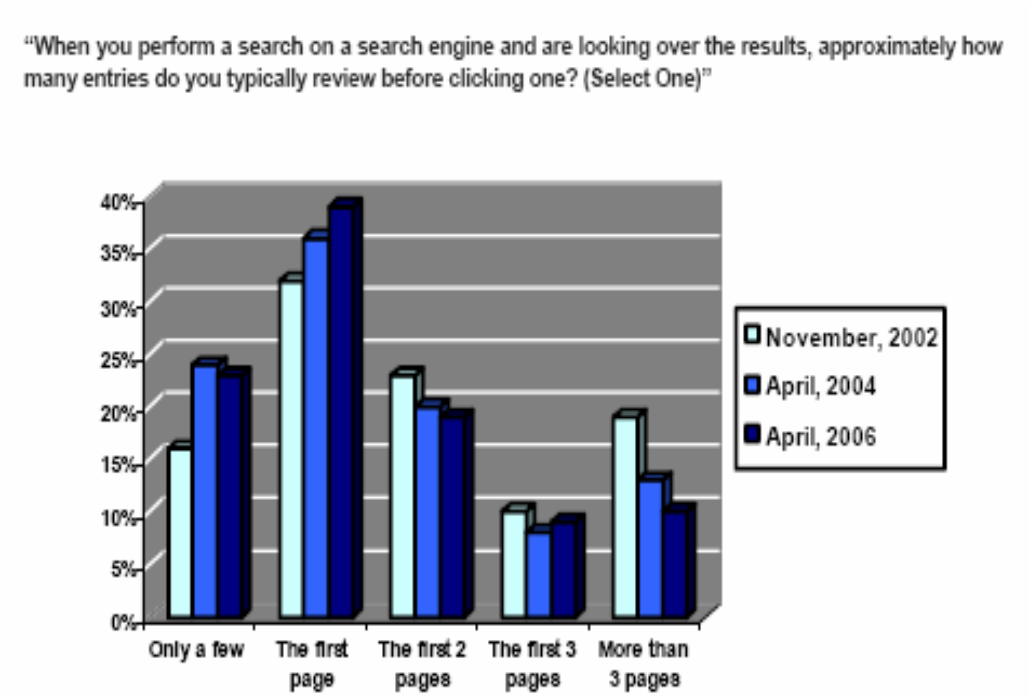
RI Web vs RI classique

- 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
    - Acceder à une ressource
      - Musique, livre, …
  - Informationnel
    - Consultation
    - Se renseigner sur un sujet
  - Navigation
    - Joindre une page donnée
- Interaction
  - Recall souvent peu important, precision mise en avant

Individus - exemple



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

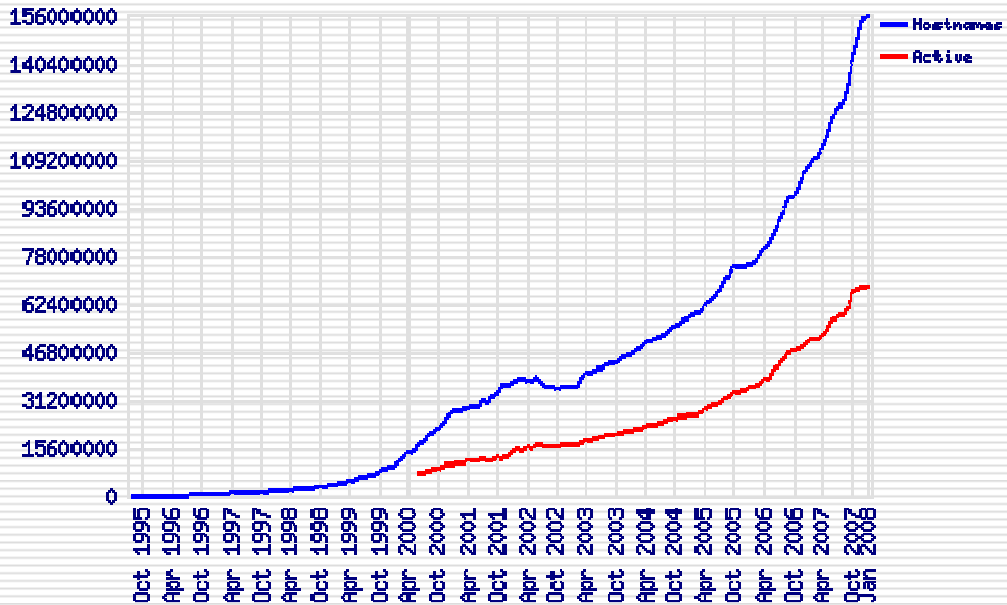
Individus

- Requêtes
  - Loi de puissance
  - Beaucoup de requêtes populaires
  - Taille moyenne requêtes < 3 mots
  - Besoins d'information dynamiques

Corpus

- Croissance désordonnée
  - Pas de coordination
- Nature des informations
  - Contient des informations obsoletes, mensongères, etc
  - Texte, html, images, structuré (XML), BD,…
  - Statique vs dynamique
    - Le web dynamique n'est pas indexé
      - Quelques travaux
    - Web caché
  - Multilingue
    - Difficulté des analyses lexicales
- Forte croissance
  - Double tous les mois
  - La taille du web réel n'est pas connue
  - Etudes sur l'estimation du nombre de pages
    - Plusieurs méthodes : marches aléatoires, etc
    - Nombre de sites (cf Netcraft)
  - Nombre de pages indexées
    - Yahoo! Annonce 20 M en 2005 ?

Croissance du web



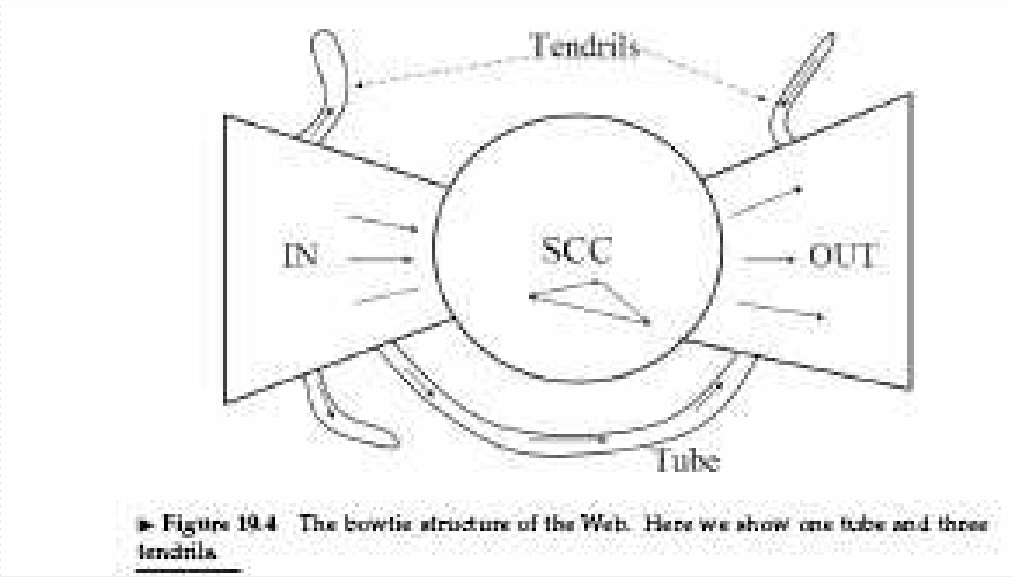
- http://news.netcraft.com/archives/web_server_survey.html
- Total Sites Across All Domains August 1995 - January 2008

Structure globale du Web

- Connexions
  - Loi de puissance
    - Le nombre de pages web de in-degree 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.

- From Manning et al. 2007

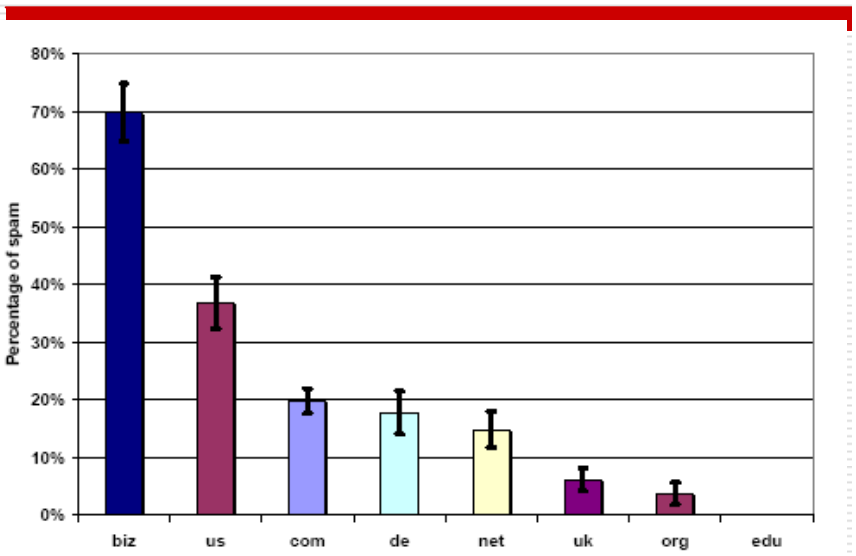
Spam sur le Web

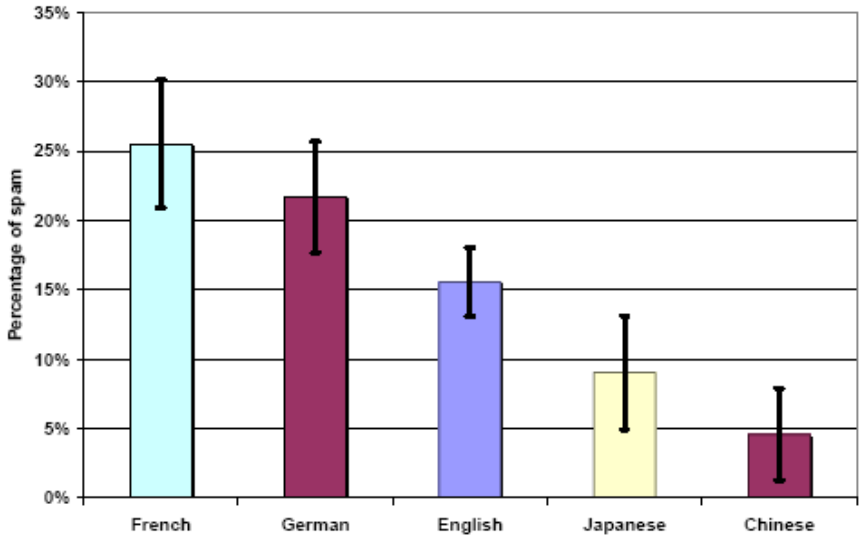
- Référencement
  - Search Engine Optimization
    - Mettre en avant ses pages / son site dans les résultats des moteurs de recherche
    - Motivations
      - Diverses : commerciales, politiques, etc
    - Devenu une industrie
    - Les moteurs essaient de faire respecter des règles aux SEO
  - Très lié au SPAM

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 fréquement demandées, etc
    - Visait les 1ers moteurs de recherche (tf-idf), facilement detecté 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

- Basés sur les liens
  - 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 sponsoriés (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)





- [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.

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/

Evolution des moteurs de recherche

- 1994 – 97
  - Excite, Lycos, etc
    - Contenu
- 1998 –
  - Google, Yahoo
    - Liens
    - Click through
    - Anchor text
- 2002 –
  - Money
  - Multiplication des services
  - Prise en compte contexte et utilisateur
  - Autres sources d'information
    - Web 2.0 etc

Analyse de lien

- Popularisée par Google avec PageRank
- Actuellement une composante parmi beaucoup d'autres des moteurs de recherche
  - Entre 10 et 100 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

- Idée
  - Marche aléatoire dans le graphe du web
  - Au bout d'un certain temps, on atteint un état stationnaire qui donne la probabilité d'atteindre chaque page visitée
  - Modélisation : chaine de Markov

  - Les pages les plus visitées lors de la marche aléatoire sont celles qui ont de nombreux in-links provenant de sites externes

PageRank

- On démarre d'une page du web
- On effectue une marche aléatoire
  - On suit un lien sur cette page avec une certaine probabilité
    - Dans le modèle de base tous les liens sont équiprobables
  - On saute à une page quelconque avec une probabilité q (0.15) : téléportation
    - Permet d'éviter de rester bloquer sur une page sans lien
    - Permet de visiter l'ensemble des pages
- On atteint un état stationnaire
  - Le taux de visite des pages dans cet état sert de score PageRank (valeur entre 0 et 1)
  - Rq : pas la peine de calculer la solution exacte, seul l'ordre entre les pages est important

Pagerank - modèle

- On modélise la M.A. par une chaîne de Markov
  - N états
    - Un état = une page
  - Une matrice de transition A_{ij}
    - A_{ij} = P(j| i) : probabilité d'aller en j quand on est en i
- Définition
  - Une chaîne de Markov est ergodique si il existe un entier k > 0 / pour toute paire d'états i, j, si le processus démarre à 0 en i, alors pour t > k, on a P(j) > 0
- Propriété
  - Toute chaîne de Markov a une distribution d'états stationnaire unique
  - Sur une période de temps suffisement longue, chaque état est visité en proportion de ce taux de visite

- État de la chaine
  - X = (x_1, …, x_n) vecteur ligne
    - x_i = P(on se trouve dans l'etat i)
  - État suivant
    - X' = X.A
- Etat stationnaire
  - Résoudre X.A = X
    - X est le vecteur propre de A associé à sa plus grande valeur propre
    - Une matrice stochastique ($0 < A_{ij} < 1$ et $\Sigma_j A_{ij} = 1$) a une valeur propre ppale égale à 1.
- Un algorithme simple
  - Algorithme de la puissance itérée
    - Partir d'un état aléatoire X
    - Itérer X.A, $X.A^2$… $X.A^k$ jusqu'à stabilité

Pagerank résumé

- Requête Q : on considère les pages qui sont pertinentes pour Q
- On les ordonne en fonction de leur score Pagerank
- Cet ordre est indépendant de la requête

- Remarques
  - Variantes
    - Marches aléatoires plus sophistiquées (bouton back, bookmarks, selection des liens non uniforme, prise en compte des intérets de l'utilisateur, PageRank topic specific, etc)
    - Prise en compte du spam sur les liens

Hits

- 2 notions à la base de la méthode
  - Hubs
    - Pages qui pointent vers des pages pertinentes pour un sujet (liens sortants)
  - Authorities
    - Pages qui sont de bonne références sur un sujet
      - qui sont donc pointées par les hubs
- Adapté à des recherches assez large
  - E.g. “voiture”
- Idée
  - Chaque page va avoir 2 scores H et A
  - On aura 2 listes ordonnées par H et A
- Algorithme
  - Itératif
    - Partir d’un petit ensemble initial de pages qui peuvent être de bons hubs ou authorities (obtenu par un moteur contenu)
    - Calculer les scores h et a pour toutes les pages de cet ensemble et pour celles qui pointent sur cet ensemble et hors de cet ensemble (c’est l’ensemble de base)

Hits – l'algorithme

- But
  - Calculer pour chaque page x dans le base set h(x), a(x)
- Initialiser
  - h(x) = 1, a(x) = 1
- Repéter
  - $$h(x) = \sum_{x \to y} a(y)$$

    $$a(x) = \sum_{y \to x} h(y)$$

- Après convergence
  - Sortir les 2 listes
    - Meilleurs h()
    - Meilleurs a()

L'algorithme - suite

- Pour un ensemble de pages web
  - **h** : vecteur des hubs de ces pages
  - **a** : vecteur des authorities de ces pages
  - L'algorithme revient à répeter :

$$\begin{cases} \mathbf{h} = A\,\mathbf{a} \\ \mathbf{a} = A^T\,\mathbf{h} \end{cases} \qquad \begin{cases} \mathbf{h} = AA^T\,\mathbf{h} \\ \mathbf{a} = A^TA\,\mathbf{a} \end{cases}$$

  - On retrouve un pb de valeur / vecteur propre
    - La méthode précédente est simplement l'algorithme de la puissance itérée pour les matrices AA^T et A^TA

- Remarques
  - L'algorithme converge
    - En pratique quelques itérations suffisent (5)
  - Indépendant du contenu effectif des pages
    - Prise en compte indirecte via les liens
  - Dérive possible vers des pages qui ne sont pas pertinentes pour la requête
  - Les sites affiliés se renforcent ce qui n'est pas l'effet voulu
  - Plusieurs solutions proposées pour ces problèmes

Corpus et documents structurés

Modèles de RI pour le Web et les corpus XML

- Les modèles standards de la RI considèrent des documents plats
- L'information aujourd'hui est largement structurée
  - Web, corpus XML, blogs, fils de discussion, etc
  - Video, multimédia, web sémantique, ontologies
- Evolution des modèles de RI pour prendre en compte les nouveaux média et les nouveaux besoins
  - Remise à « plat » des principes de base de la RI
  - En pratique : passe souvent par une adaptation des concepts et modèles existant

Modèles de RI et Web

La Webtrack de TREC (2004)

- Tâches
  - Topic distillation
    - Q décrit une requête générale, le système retourne des pages pertinentes
  - Homepage finding
    - Q est le nom d'un site e.g." togo embassy", le systeme retourne l'url du site dans les top r
  - Name page finding
    - Q correspond à une page e.g. "services sociaux de la mairie de Paris", le système renvoie l'url de cette page dans les top r
  - HP et NP : on ne cherche pas tous les docs pertinents mais un site ou une page

Documents Web

- Structure
  - Présente au niveau du web, des sites, des pages HTML, XHTML
  - Nombreux algorithmes pour prendre en compte la structure du web
    - Pagerank, Hits etc
    - on n'en parle pas dans l'exposé
  - Modèles de RI
    - 2 grandes familles
      - Fusion de scores des différentes composantes du document ou sources d'information (heuristiques ou par apprentissage)
      - Fusion des information au niveau de la représentation même des documents

Okapi pour des documents structurés BM25F (Robertson et al 2004)

- Document structuré avec différents champs
  - (page ou site web)
- Question
  - comment combiner ces différentes informations ?
- Constat
  - La combinaison de scores apporte peu lorsqu'il faut combiner de nombreux champs
- Proposition
  - Prendre en compte les différents champs directement dans la représentation des documents
- La méthode obtient les meilleurs résultats à TrecWeb 2004
  - Utilisée également pour plusieurs autres tâches

Rappel Okapi

- Rappel : Okapi BM25 classique

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

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

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

- Critique de la combinaison de scores
  - Difficile de combiner les scores de champs de nature très différente
    - Robustesse des statistiques dans les différents champs, confiance dans les scores pour les différents champs etc
  - La non linéarité (saturation) de la fonction perd son sens dans cette combinaison
    - E.g. un document contenant un terme de la requête dans différents champs peut avoir un meilleur score qu'un document contenant plusieurs termes de la requête dans un seul champs
  - etc

BM25F

- Un document comprends différents champs
- A un terme *t*, on associe le vecteur de ses fréquences dans les différents champs
- La représentation fréquentielle d'un document est définie par l'ensemble des vecteurs de fréquence de ses termes.
- La fréquence d'un terme *t* est alors définie comme une combinaison des fréquences de *t* dans les différents champs
- Le score est calculé de manière analogue à OKAPI classique

BM25F

- $F = (F_1,\ldots,F_n)$ un ensemble de champs
- A chaque champ F_i est associé un poids v_i
- $tf_{F,t}$ fréquence de t dans le champs F de d
- DL_F *et* $AVDL_F$ sont la longueur du champs F dans d et la longueur moyenne du champs F dans le corpus

$$\overline{tf}_t = \sum_i v_i tf_{F_i,t}$$

$$w_F(tf_t) = \frac{\overline{tf}_t (k_1 + 1)}{K_F + \overline{tf}_t}$$

$$K_F = k_1 * ((1-b) + b(DL_F / AVDL_F))$$

$$DL_F = \sum_i v_i . DL(F_i)$$

$$AVDL_F = \sum_i v_i . AVDL(F_i)$$

$$Score\ BM25F = \sum_{t \in q} w_F(tf_t) * "idf(t)"$$

- Les différents paramètres sont choisis séquentiellement et séparément de façon à optimiser une mesure e.g. precision@10
- Les formules utilisées effectivement (TREC) diffèrent un peu de celle donnée ici
- Application au Web
  - Combinaison des champs des documents
    - Titre, corps, ancre liens hypertexte
    - Combinaison avec d'autres sources d'information (e.g. score page rank pour le web) par de simples combinaisons linéaires
- L'idée est utilisable avec d'autres méthodes

Combinaison dans des modèles de langage (Ogilvie, Callan 2003)

- Les modèles de langage permettent naturellement de combiner différentes sources d'information
- Dans le cas du web par exemple

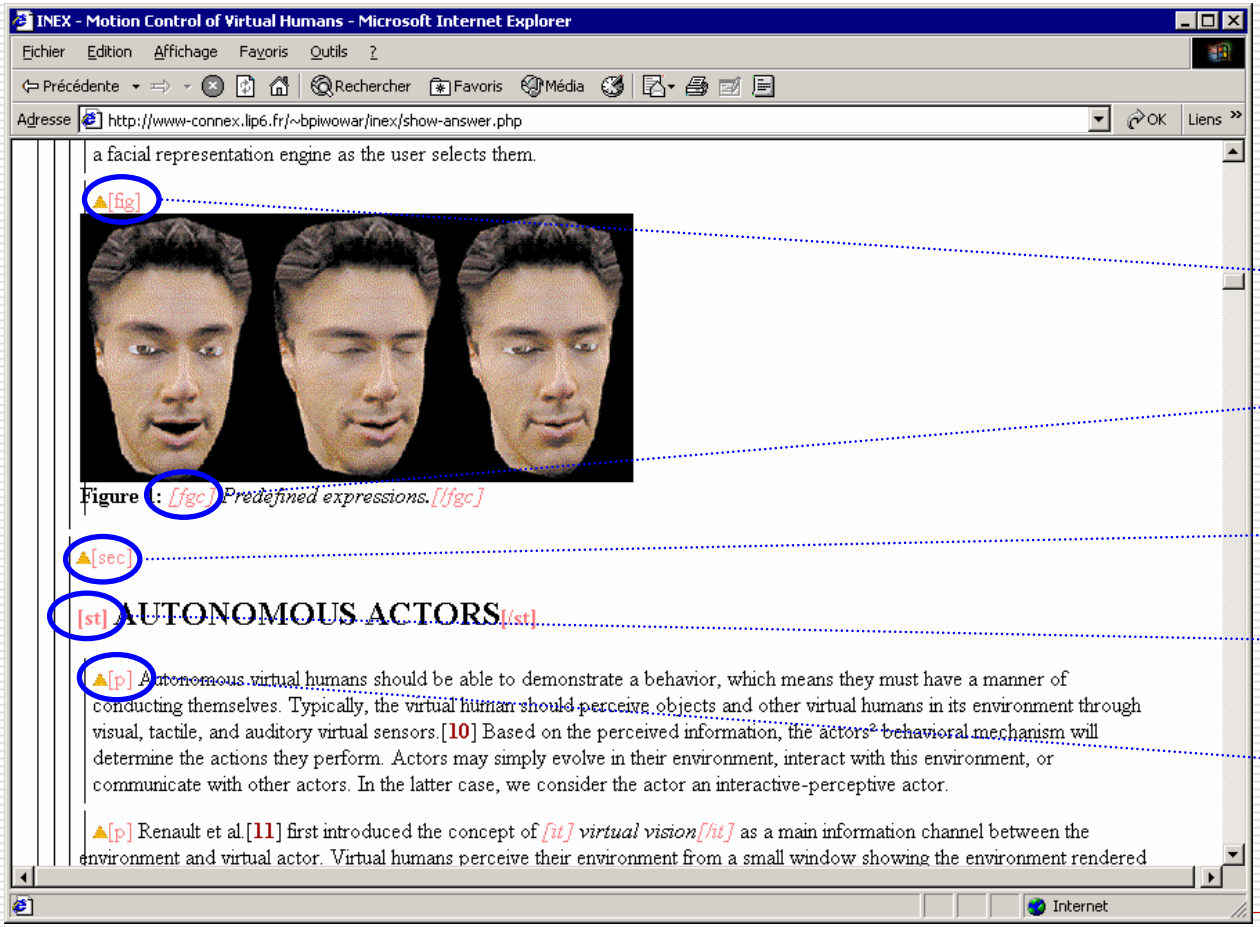
$$P(d|q) = P(d)\prod_{t\in q} \lambda_1 P(t|Corpus) + \lambda_2 P(t|d, ancre) + \lambda_3 P(t|d, body) + \lambda_4 P(t|d, titre)$$

$$\sum_i \lambda_i = 1$$

- Les λ_i sont estimés sur les données
- Rq : par rapport à des combinaison classiques, ici, la combinaison des scores est faite séparément pour chaque terme et non pour chaque composante – équivalent à combiner les composantes dans la représentation des documents

La recherche dans les documents structurés XML

Documents XML



Recherche d'information structurée

- But
  - RI dans les corpus de documents structurés e.g. XML
  - Considerer simultanément la structure logique et le contenu

- Change la perspective sur la RI
  - Requêtes
  - Eléménts à rechercher
  - Evaluation
  - Interaction

« INEX » E. U.

- Corpus:
  - 2002-2005 - 500 Mo de documents XML + requêtes + jugements de pertinence, 16 000 documents (IEEE journals), 10 millions de doxels
  - 2006 Wikipedia XML : textes anglais de Wikipedia, 659,388 articles couvrant une hierarchie de 113,483 categories, > 60 Gigabytes, 5000 tags differents. En moyenne an article contains 161.35 nœuds XML par article, profondeur moyenne d'un élément 6.72.

Inex - requêtes

- Requêtes
  - Content Only CO
  - Content and Structure CAS – VCAS
  - Constitution d'une requête
    - Title
      - Expression du besoin d'information
      - CO : mots clés, CAS :

//article[about(.,interconnected networks]//p[about(.,crossbar networks)]

    - Topic description
      - 1 ou 2 phrases en langage naturel
    - Narrative
      - Descriptif plus complet
  - Exemple en 2004
    - 30 CO, 30 CAS
    - 37 000 doxels judged for CO (1500 per question)
    - 34 000 doxels judged for CAS (1137 per question)
  - Coût de l'assessment : 20 h / requête !

Inex – tâches (2005)

- Tâche de base : Focused
  - Retrouver les éléments pertinents au bon niveau de granularité
- Pour analyser le comportement des systèmes, 2 autres tâches
- Thorough
  - Retrouver tous les éléments pertinents sans prise en compte de la dépendance entre éléments (i.e une section et ses paragraphes)
- Fetch and browse
  - Fetch : identifier les articles pertinents
  - Browse : identifier les éléments dans ces articles

INEX 2002 assessments

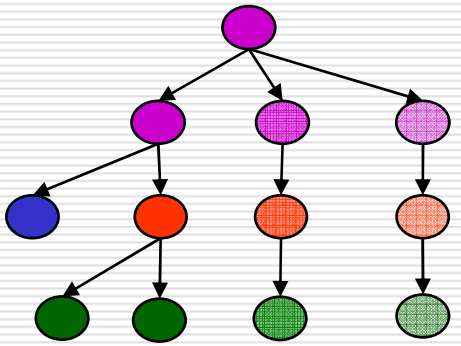
- Deux dimensions
  - Exhaustivité
    - Un doxel « exhaustif » contient l'information
    - Echelle 4 valeurs
  - Specificité
    - Il ne contient pas d'autre information
    - 4 valeurs 2004, continu [0,1] 200510 valeurs indépendantes sur 16

Too large (G)

Exact (E)

Too small (P)

Non exhaustive (I)



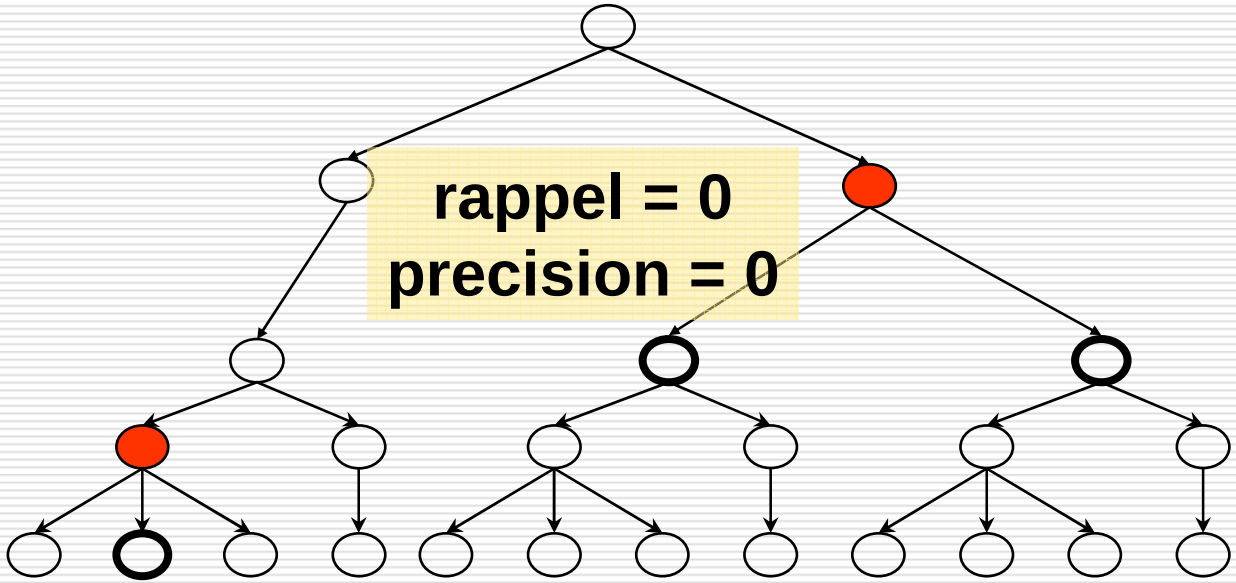
Evaluation

- Difficultés
  - Prise en compte des relations entre éléments
    - Near misses : on retourne un élément « voisin » d'un élément recherché
    - Overlap : la même information est retournée plusieurs fois (paragraphes, section, etc)
  - Prise en compte de l'echelle graduelle à 2 dimensions (E,S)
- Problème difficile
  - Différents métriques

Precision - rappel

Pas adapté

e.g. système retournant systèmatiquement un doxel plus petit



Mesure utilisée à inex 2005 : Gain cumulé (Kazai et al 2005)

- Base de rappel
  - Base de doxels pertinents dans le corpus
- Liste de documents retournés par le système
- Gain d'un doxel de rang *i* dans la liste
  - A un doxel *x* on associe un score *xG[i]* qui mesure le « gain » d'information apporté par ce doxel (dépend des jugements et de la liste elle-même)
- Gain idéal *xI[i]*
  - La liste idéale est composée de l'ensemble des doxels pertinents de la base ordonnés par leur degré de pertinence calculé en fonction des « assessments »
- Gain cumulé au rang *i* : $xCG[i] = \sum_{j=1}^{i} xG[i]$ $CI[i] = \sum_{j=1}^{i} xI[i]$
- Métrique : gain normalisé : $nxCG[i] = \frac{xCG[i]}{xCI[i]}$

Precision-Recall with User Modeling (Piwowarski et al. 2005)

- Mesure de précision-rappel probabiliste qui prend en compte overlap – near misses – navigation utilisateur

$$PRUM(i) = P(Lur|Retr, L = l, Q = q)$$

i = niveau de rappel dans [0,1]

Lur = évènement : l'élément conduit à un doxel pertinent

Retr = évènement : l'élément est dans la liste consultée

l = pourcentage d'éléments pertinents que l'utilisateur veut voir

q = requête

Modèles

- Différentes adaptations des modèles classiques de RI
  - Modèle vectoriel
  - Modèle de langage
  - Réseaux Bayesiens

- Remarques
  - Nombreux essais sur l'indexation, les pondérations, etc : pas de consensus général sur ce qui est le mieux
  - Importance du lissage des estimations,
  - Requêtes CAS : différentes méthodes pour la prise en compte des contraintes (index de la structure)
  - Overlap souvent traité en post-processing

Modèle vectoriel (Mas et al. 05)

- Adaptation directe du modèle vectoriel
  - 1 index par type d'élément « significatif »
    (articles, section, sous section, paragraphe)
  - Modèle vectoriel appliqué séparément à chaque type
- Algorithme
  - Appliquer le modèle vectoriel sur le type *i*
  - RF sur les sorties de type *i*
  - Normaliser les scores dans [0,1] en divisant le score par RSV(q,q)
  - Interpolation avec le score de l'article
  - Ordonner la liste globale comprenant tous les types avec les scores normalisés
- Leçon : importance de l'interpolation (+30 %) et du RF

Modèle de langage (Kamps et al. 2005)

- Index pour les articles, et les types de doxels

(redondance : l'index de la section contient les termes du paragraphe)

  - Un modèle de langage par type de doxel
  - Lissage
  - Priors sur la taille des éléments : incorporation d'information de contexte
  - RF par type de doxel
  - Comparaison directe des scores des différents éléments

$$P(e|q) \propto P(e)P(q|e) = P(e)\prod_{i=1}^{n} P(t_i|e)$$

$$P(t_i|e) = \lambda_{elt} P_{ml}(t_i|e) + \lambda_{doc} P_{ml}(t_i|doc) + \lambda_{corpus} P_{ml}(t_i|corpus)$$

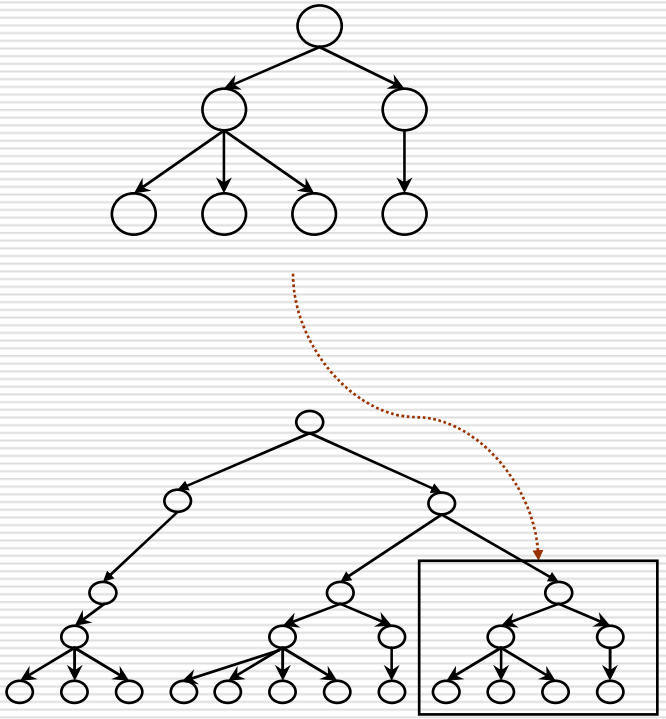
$$P(e) = \frac{|e|}{\sum_{e'}|e'|}$$

Réseaux Bayesiens (Piwowarski et al. 2003)

- Les documents structurés sont considérés comme des arbres
- Modèle : RB arborescent
- Les scores sont calculés par inférence dans le RB
- Probabilités conditionnelles du RB estimées sur le corpus

Corpus modeling

Modèle de document

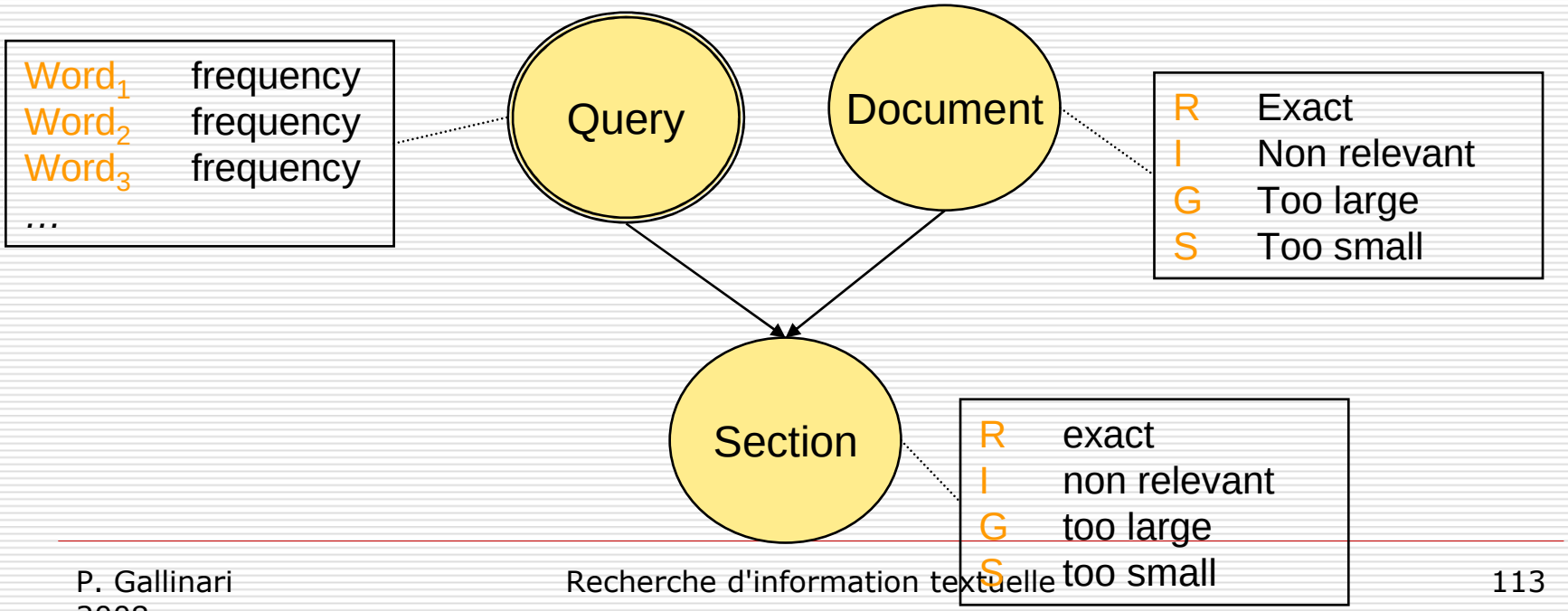


Modèle du corpus : un RB construit à partir des RB des documents

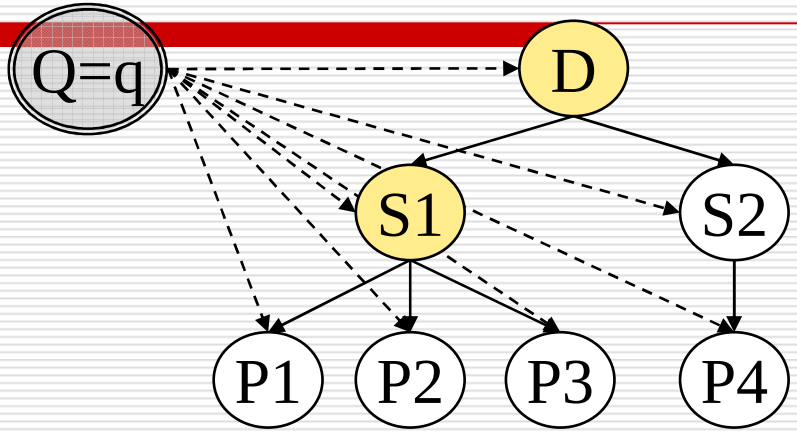
La structure du réseau reflète celle du corpus

Doxel

- La pertinence des doxel depend de
  - Celle du parent
  - La requête



Modèle de Document



- La pertinence du doxel est calculée par inférence dans le réseau
  - $P(D = R / Q = q)$, $P(S_1 = R / Q = q)$, $P(S_2 = R / Q = q)$, …
- Pour cela il faut connaitre

$$P(\text{ doxel relevance} = x / \text{parent relevance} = y, \text{query} = q)$$

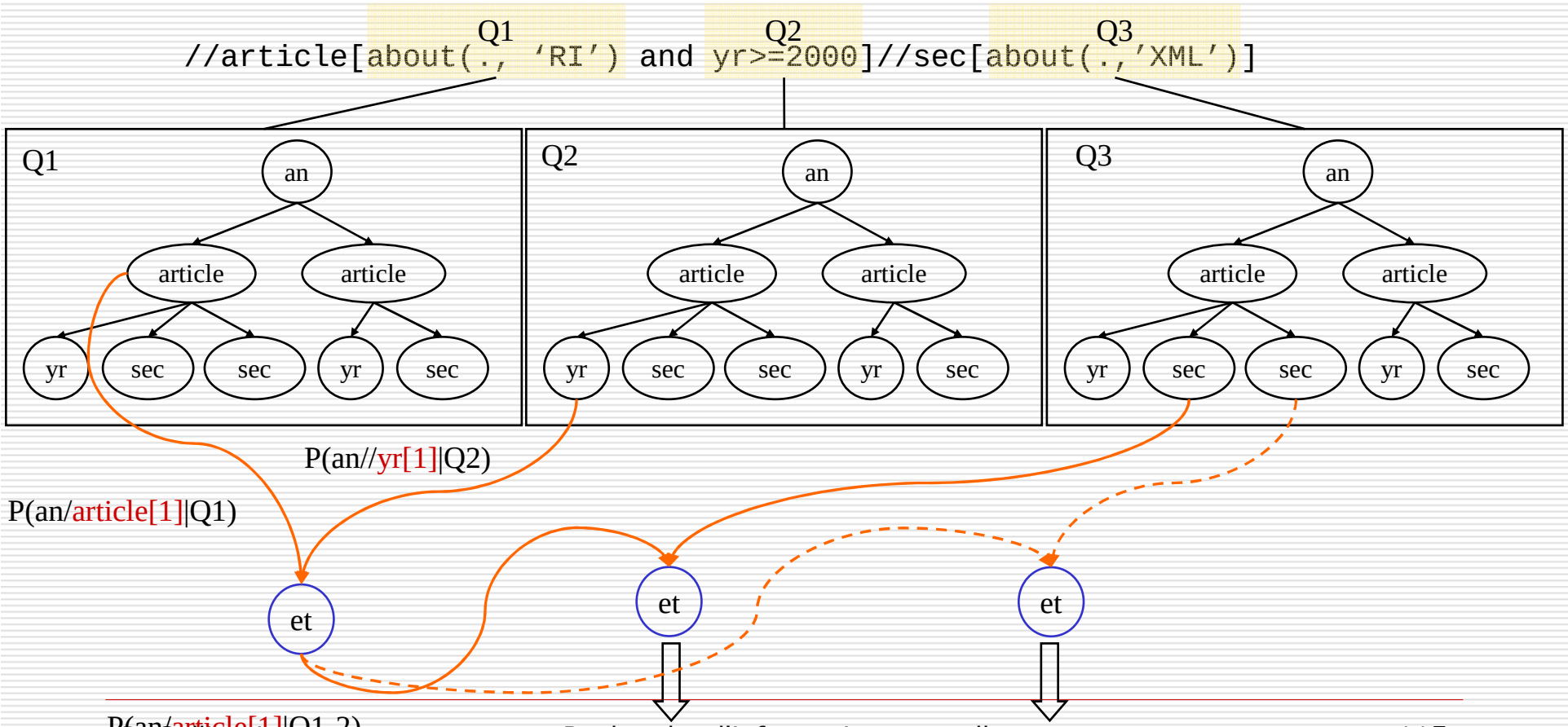
$$=$$

$$\mathbf{F}(\text{doxel},x,y,q,\Theta)$$

- Cette fonction est calculée pour chaque requête
- Apprentissage : parmètres du modèle Θ par gradient.

Requêtes « CAS »

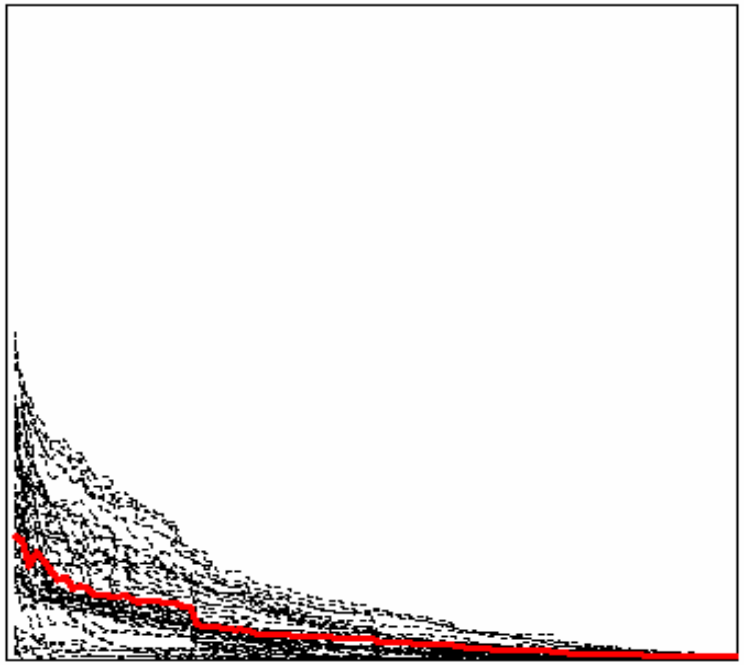
« I want a section on XML in an article about RI published in 2000 »



Precision recall on INEX 2003

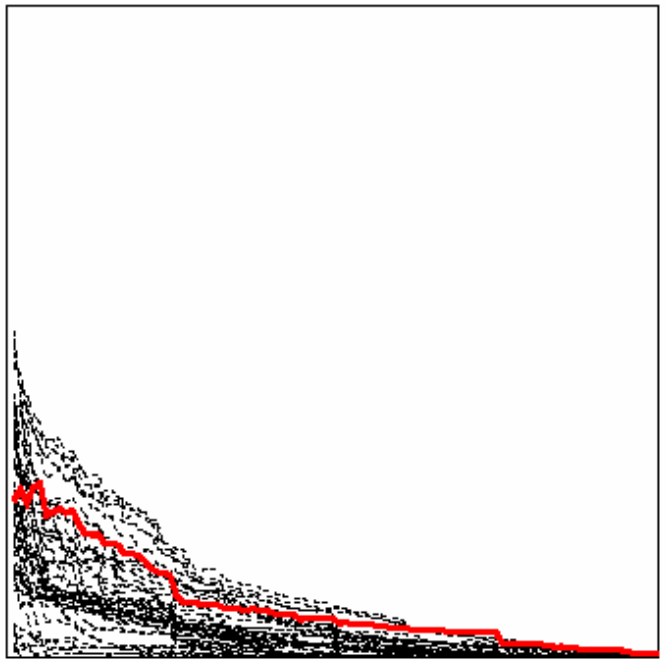
INEX 2003: t3-g-simple,trivial-ef-lf

quantization: strict; topics: CO
average precision: 0.0454
rank: 19 (56 official submissions)



INEX 2003: local-okapi-element,list,ef

quantization: strict; topics: CO
average precision: 0.0809
rank: 7 (56 official submissions)



Apprentissage de fonctions d'ordonnancement pour la recherche structurée (Vittaut 2005)

Apprentissage de fonctions d'ordonnancement

- Apprendre automatiquement une fonction d'ordonnancement
- Utilisé pour combiner des caractéristiques, des scores ou des relations de préférence dans différentes tâches :
  - meta search, Résumé automatique, RI, Pooling, etc
- Les algorithmes d'ordonnancement combinent des caractéristiques des éléments à ordonner.
  - Possibilité d'incorporer des informations de nature différente
- En SIR, les caractéristiques vont dépendre :
  - Du doxel lui même (contenu)
  - De son contexte structurel (etiquette, parent, etc)
- Peut être utilisé avec toute méthode qui fournit des scores pour les doxels et pour combiner ces méthodes entre elles

Principe

- Apprendre un ordre total sur un ensemble X, qui permette de comparer tout couple d'élément de cet ensemble.
- Etant donné cet ordre total, on peut ordonner tout sous ensemble de X

Exemple

En IR, X peut être un ensemble de couples (document, requête), et l'ordre total est l'ordre naturel sur les scores.

Comment apprendre ?

- L' ensemble d'apprentissage consistera en paires d'exemples ordonnés.
- Il n'est pas nécessaire d'ordonner l'ensemble des paires
- Cela va fournir un ordre partiel sur les élements de X.
- L'algorithme d'ordonnancement va utiliser cette information pour apprendre un ordre total sur les éleéments de X : la fonction d'ordonnancement.
- Celle ci va permettre d'étendre l'ordre partiel à tous les éléments du corpus (ordre total).

Exemple

- Pour SIR, X sera l'ensemble de tous les couples (doxel, requête) dans la collection de documents.

- Cet ensemble est partiellement ordonné selon la pertinence des jugements pour chaque requête

Notations

- Un élément de X sera représenté par un vecteur de caractéristiques réelles

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

- Dans notre cas, les caractéristiques seront les scores locaux calculés sur différents éléments contextuels d'un doxel.
- La fonction d'ordonnancement sera une combinaison linéaire des caractéristiques de x

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

- $w = (w_1, \ldots, w_n)$ sont les paramètres à apprendre

Coût d'ordonnancement

- le coût d'ordonnancement mesure à quel point f_w respecte l'ordre

$$R(X, w) = \sum_{\substack{(x,x') \in X^2 \\ x \prec x'}} X(x, x')$$

avec $X(x, x') = 1$ si $f_w(x) > f_w(x')$ et 0 sinon

- R est non différentiable
- Les algorithmes d'ordonnancement optimisent un coût exponentiel :

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

Ranking vs Classification

- Classification
  - Prédit quel doxel est pertinent ou non pertinent
  - Ne s'intéresse pas à l'ordre des doxels
  - Minimise l'erreur de classification $P(C|x)$
- Ordonnancement
  - Considère uniquement l'ordre des doxels
  - Minimise le nombre de couples mal ordonnés $P(x \prec x' | x, x')$
  - L'échelle des scores n'est pas importante

Combinaison

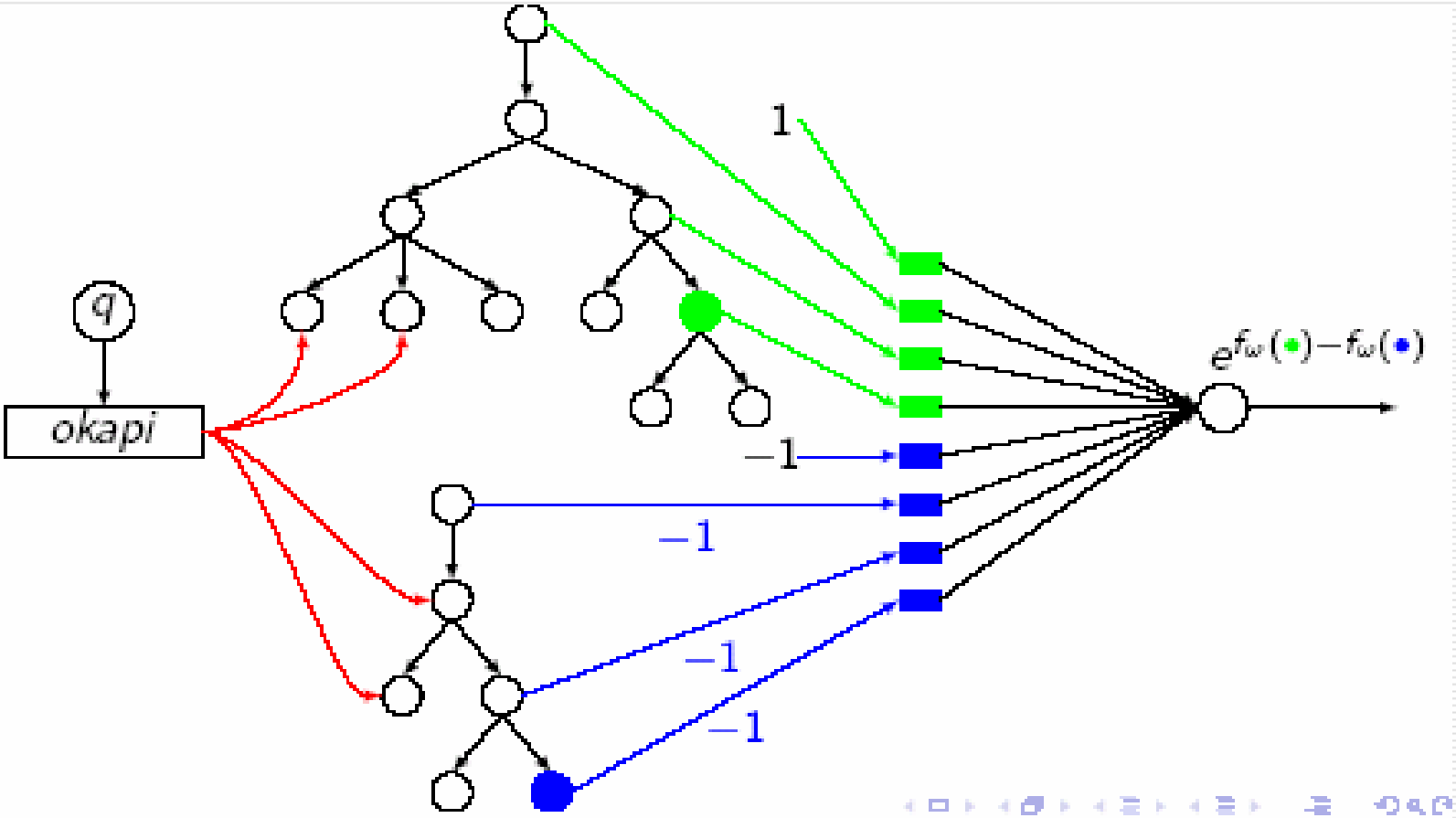
- Nous avons utilisé la combinaison suivante

$$f_w(x) = w_1^l + w_2^l Okapi(x) + w_3^l Okapi(pa(x)) + w_4^l Okapi(doc(x))$$

- Okapi est un modèle Okapi adapté à SIR

- Le paramètre w_i^l dépend
  - De la caractéristique *i*
  - Du type du noeud *l*

Combinaison



Reduction de la complexité

- Comparer des éléments pour différentes requêtes n'a pas de sens
- Pour chaque sous ensemble, les préférences entre les doxels sont exprimées suivant plusieurs dimensions
- Il n'y a pas de préférence entre éléments partageant la même valeur d'exhaustivité - spécificité

Reduction de complexité

- La fonction de coût initiale est quadratique pr aux assessments

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

- Elle se réécrit sous une forme qui est linéaire pr aux nombre d'assessments

$$R_e(X,w) = \sum_{\substack{(ES,E'S') \\ E'S' \prec ES}} \left(\sum_{x \in assessments(ES)} e^{f_w(x)} \right) \left(\sum_{x' \in assessments(E'S')} e^{-f_w(x')} \right)$$

Assessments

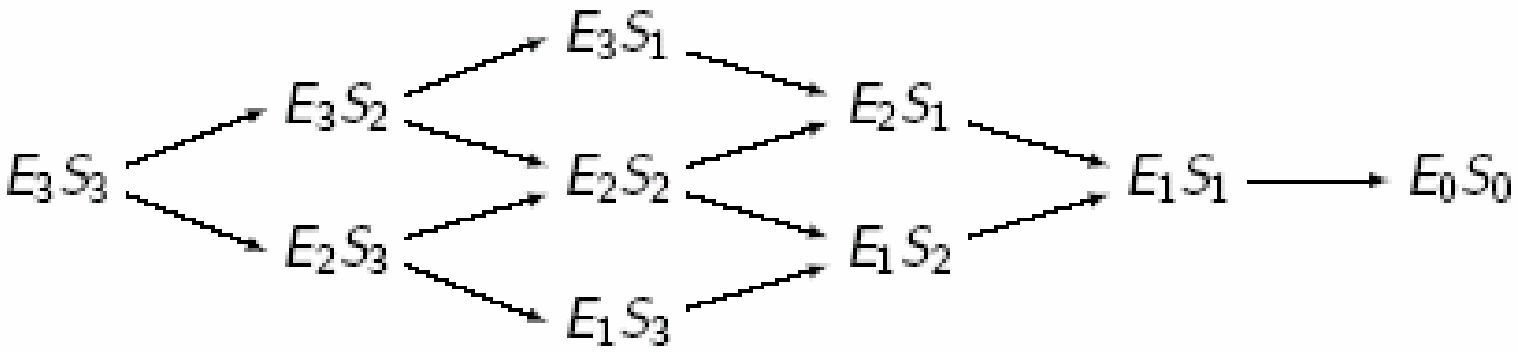
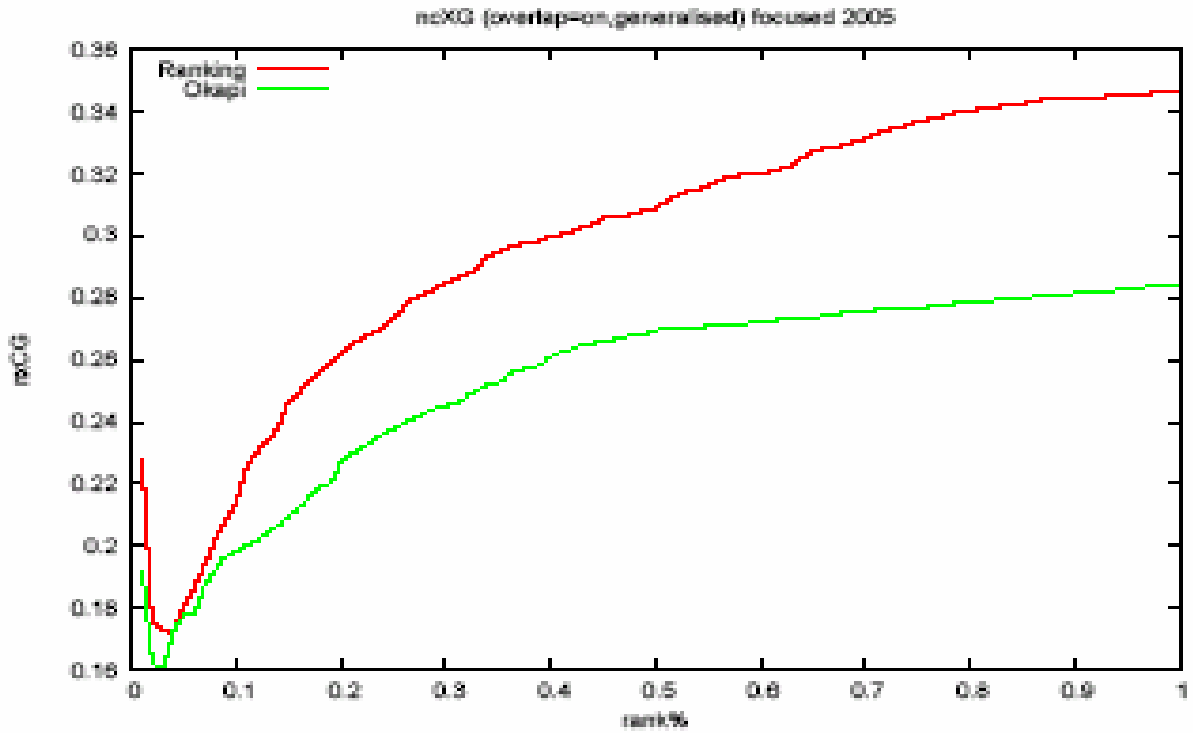


Figure: Lattice representing the order between elements for a given query, according to the two dimensional discrete scale of INEX. Doxels labeled E_3S_3 must be the highest ranked, and doxels labeled E_0S_0 the lowest ranked.

CO Focused

Topics et assessments de Inex 03, 04 pour l'apprentissage
Inex 05 pour le test

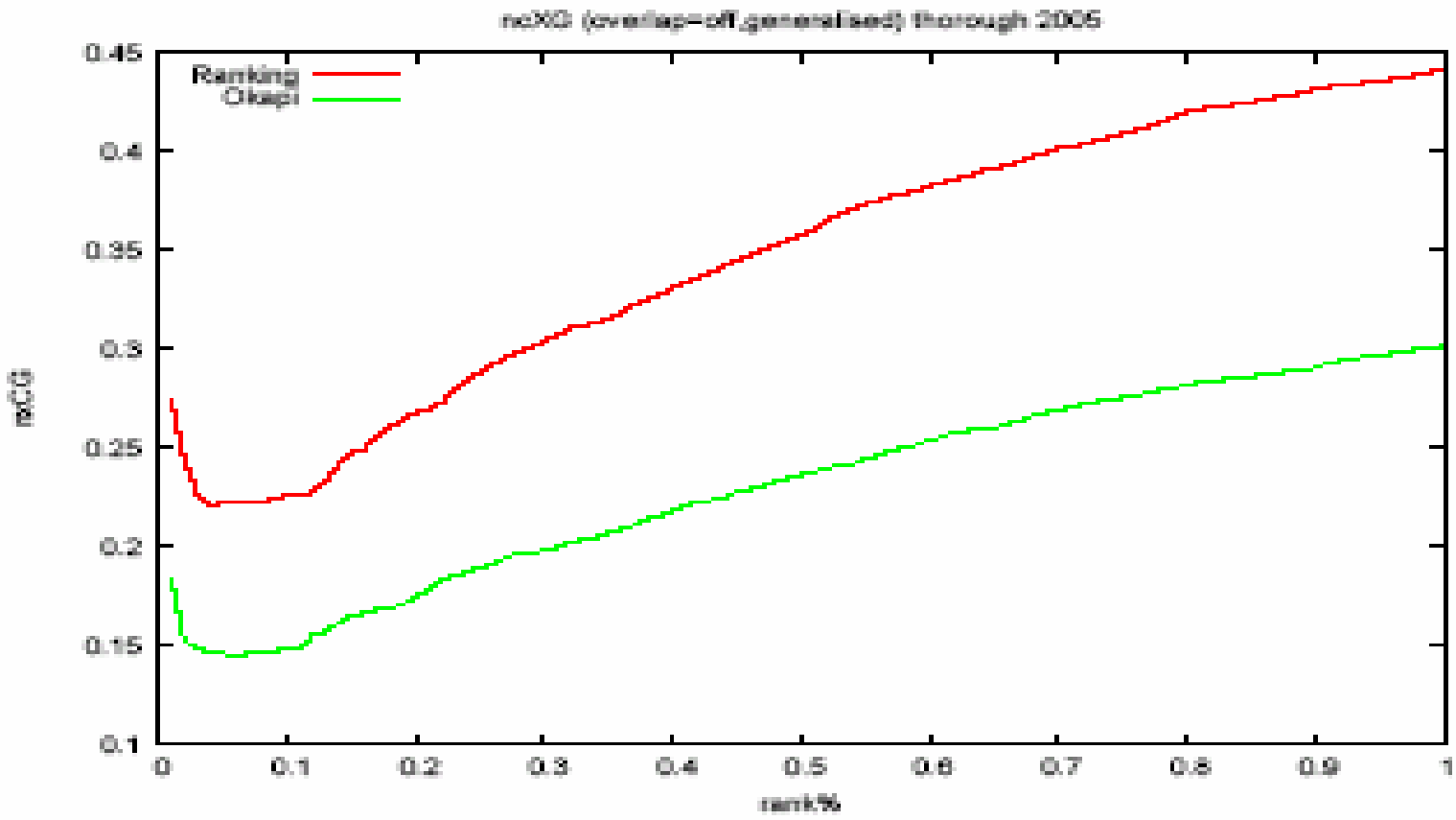


CO-Focused

Table: Rank of Okapi and ranking models among all participant submissions using MAncXG metric for CO-Focussed task

	@1	@2	@3	@4	@5	@10	@15	@25	@50	@100	@500	@1000	@1500
Okapi	21	20	19	19	18	18	19	19	19	18	20	20	20
Ranking	1	1	1	1	2	7	11	13	15	14	10	14	13

CO-Thorough


nxCG (overlap=off,generalised) thorough 2005
Ranking
Okapi
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
nxCG
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
rank%

CO-Thorough

Table: Rank of Okapi and ranking models among all participant submissions using MAncXG metric for CO-Thorough task

	@1	@2	@3	@4	@5	@10	@15	@25	@50	@100	@500	@1000	@1500
Okapi	26	22	26	26	26	31	34	37	38	38	35	32	32
Ranking	1	1	1	2	2	3	3	4	11	12	5	5	6

Extraction d'Information

Extraction examples

- Q/A
  - What was W. Shakespeare occupation before he began to write plays
  - Who is Tom Cruise married to
    - “married actors T. Cruise and Nicole Kidman play dr. William and Alice Hartford, a N.Y. couple who think their eight year marriage is very good”
- On line adds
  - Capital Hill
    - 1 br twnhme. Fplc D/W/W/D Undrgrnd pkg incl $675
    - 3 B, upper flr of turn of ctry HOME. Incl. gard, grt N. Hill, loc $995….

Information Extraction

- Unstructured text
  - Newspapers, scientific articles, etc
  - Closed extraction - MUC: Message Understanding Conferences
  - Open extraction - Question/Answering in TREC
- Structured text
  - HTML pages – Regular structures
- Specific approaches for each task

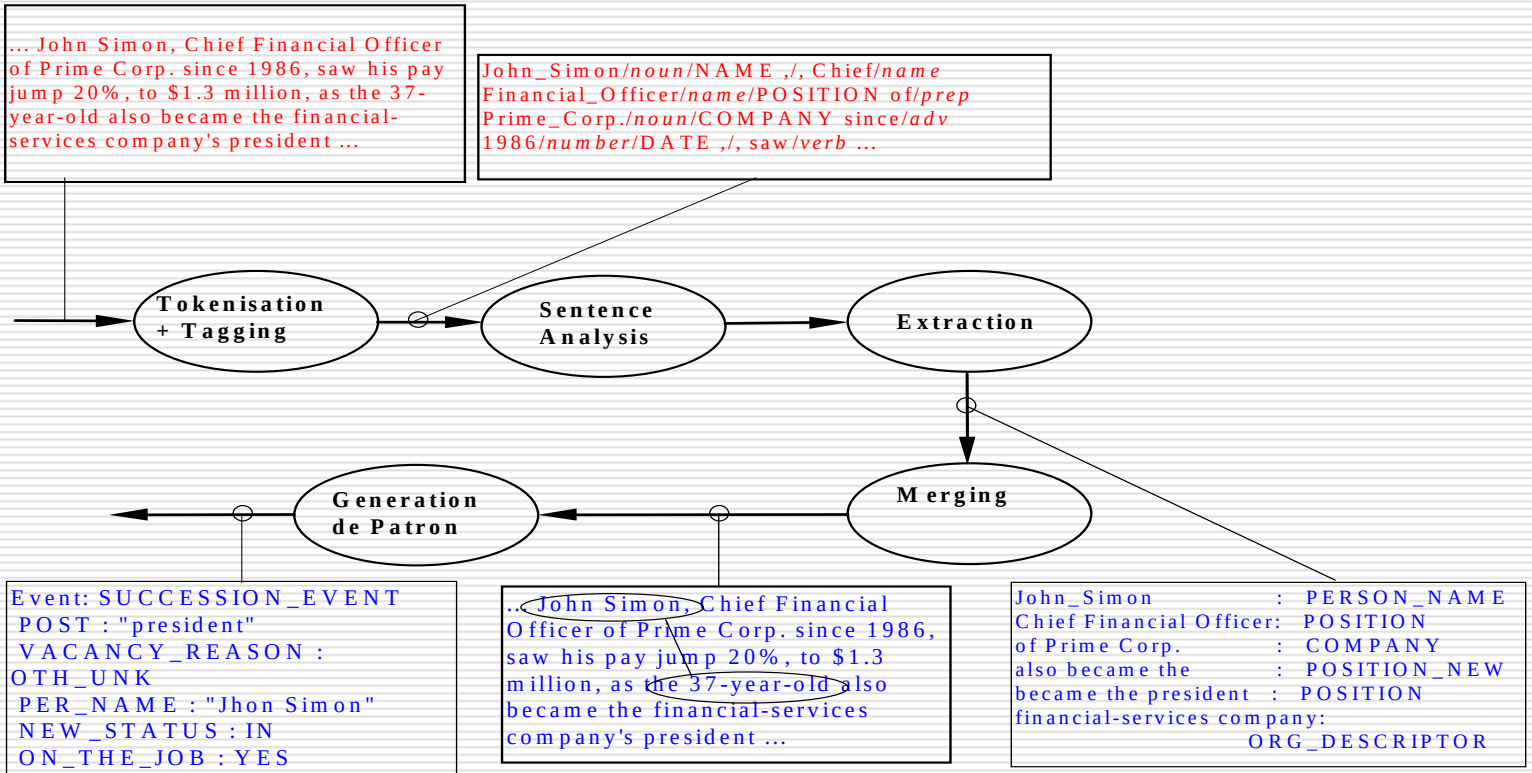
Message Understanding Conferences (MUC)

- Message Understanding Conferences : évaluations sur des tâches "pratiques" d'analyse et de compréhension de texte. MUC 1 (1987) à MUC7 (1998)
- 3 tâches :
  - développer des composants IE immédiatement utilisables, indépendants du domaine, automatiques
    - e.g. identifier tous les noms d'organisation et d'individus dans des textes.
  - portabilité des systèmes d'extraction
    - e.g. retrouver dans un texte les informations concernant les mouvements d'individus dans les compagnies.
  - évaluation sémantique
    - Coréférences, Désambigüation, Structure : constituants d'une phrase

Extraction d ’information (MUC)

- Une tâche TLN spécifique du domaine
  - Données : texte libre
  - Sortie : « résumé » du texte concernant des sujets d'intérêt spécifique : codé sous une forme structurée
  - Comment :
    - analyse superficielle du texte complet
    - détection des sections pertinentes au sujet
    - analyse de ces sections pour extraire l'information

Vision MUC de l'IE



Apprentissage et EI

- 2 Approches
- EI : Automatiser les étapes d'une chaîne d 'EI :
  - Extraction de références et résolution de co-références
- Apprentissage
  - Extraction d 'information de surface

Extraction : Autoslog (Riloff 1993)

- Forme linguistique = Concept node
- Exemple :
- CN
  - Name : target-subject-passive-verb-bombed
  - Trigger : bombed
  - Variable slots : (target (*Subj* 1))
  - Constraints : (class phys-target *S*)
  - Constant Slots : (type bombing)
  - Enabling Conditions ((passive))
- Phrase : "in la oroya, ..., public buildings were bombed and a car-bomb was detonated"
- Le CN extrait : "public buildings"

Algorithme Autoslog

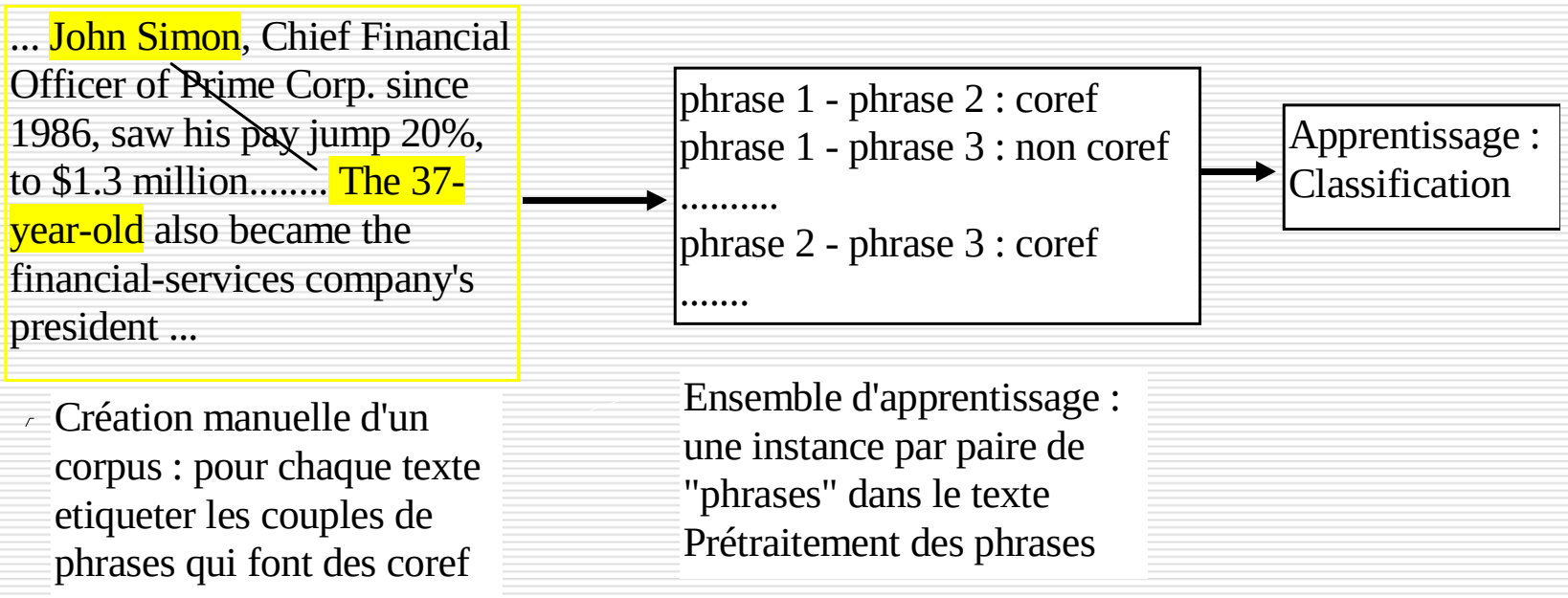
- Données : textes + termes à extraire dans chaque texte
  - Trouver la 1ère « phrase » contenant 1 expression recherchée e.g. « public buildings » ...
  - Analyse syntaxique de la phrase : sujet, verbe, ...
  - Application de règles heuristiques (13 formes linguistiques simples)
    - e.g <target - sujet> verbe passif → cf exemple
    - verbe actif <target - obj-dir>
    - prep <gpe nominal>

(Suite)

- Si une règle s'applique, générer 1 CN définie à partir des instances auxquelles elle s'applique → dans l'exemple, déclencheur = bombed
- Générer les formes linguistiques qui généralisent le mieux possible → apprentissage inductif sur les CN, e.g. CRYSTAL 1995 (algorithme à la Foil)

Merging : résolution de co-références

Déterminer quand deux "phrases" se réfèrent à la même entité



Surface Information Extraction

- HMMs
  - Generic tasks & documents
    - Bikel et al. 98 (BBN) : named entities
    - Zaragoza et al. 98 (LIP6) : Classification + extraction
    - Freitag et al. 99 (CMU) : entities
    - Seymore et al. 99 (CMU) : text structure
  - Specific tasks
    - Leek 98 Biology - HHM 64 states
- Other stochastic models
  - SVM, CRFs since 2005
- Rule learners
  - Freitag 98, Naïve Bayes + rule learner
  - Craven 99, Naïve Bayes + Foil

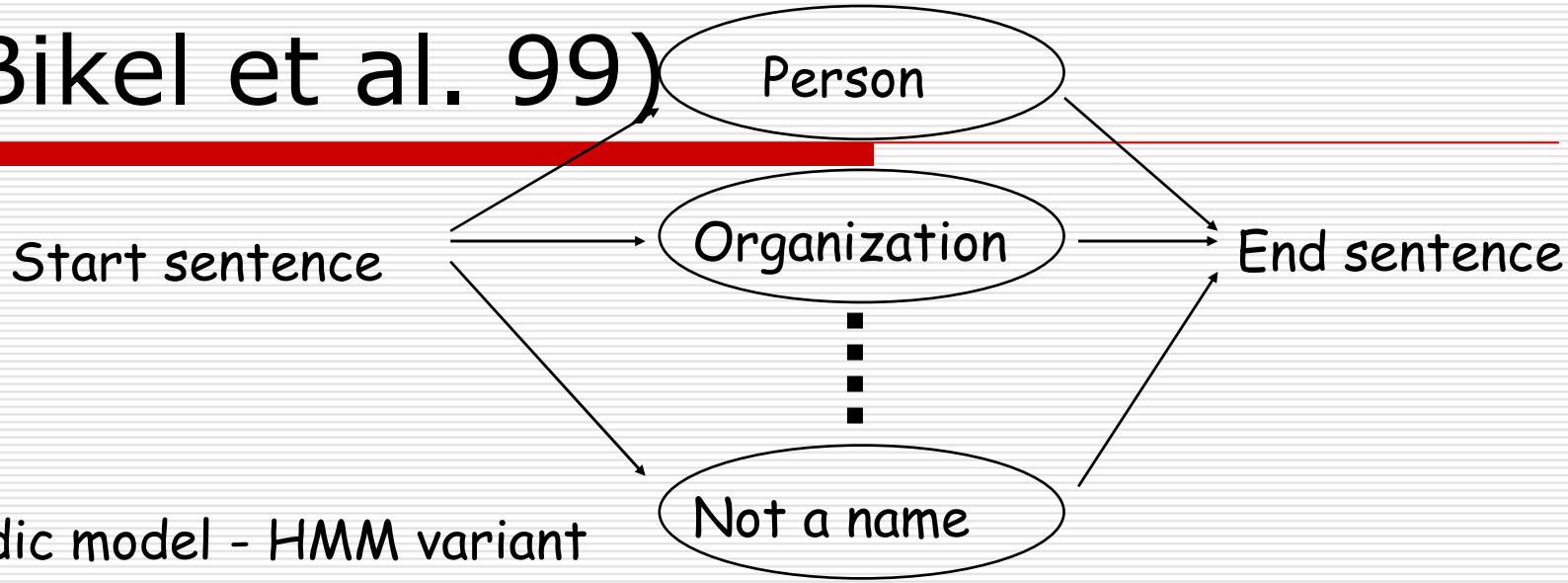
Named Entities (Bikel et al. 99)

- Processing step in many natural language / extraction tasks
- Evaluation in MUC 6, 7
- MUC: Identify all named
  - Entities: locations, persons, organizations
  - Time expressions: dates, times
  - Numeric expression: $ amount, percentages
    Representative of different extraction tasks
- Usual techniques: Handwritten Finite state machines

- NEW YORK (Reuters) - Goldman Sachs Group Inc. agreed on Thursday to pay $9.3 million to settle charges related to a former economist …. Goldman's GS.N settlement with securities regulators stemmed from charges that it failed to properly oversee John Youngdahl, a one-time economist …. James Comey, U.S. Attorney for the Southern District of New York, announced on Thursday a seven-count indictment of Youngdahl for insider trading, making false statements, perjury, and other charges. Goldman agreed to pay a $5 million fine and disgorge $4.3 million from illegal trading profits.

- LONDON (Reuters) - Veteran quarterback Vinny Testaverde, just two months short of his 40th birthday, will lead the New York Jets in the NFL's season-opening game against the Washington Redskins on Thursday. Testaverde, just 442 yards short of becoming the ninth NFL quarterback to throw for 40,000 yards, amassed just 499 yards last season after he was benched when the Jets went 1-3 under his leadership.

(Bikel et al. 99)



Ergodic model - HMM variant

Labeling: p(w sequence, NC sequence)
Input w = word + feature
Features: 4 digits nb (2003), digits and period (5.6) Cap period (M.), Init cap (John)

Transitions	$P(NC/ NC_{-1}, w_{-1})$
Emission	$P(w_{first}/ NC, NC_{-1})$
	$P(w/ NC, w_{-1})$

About 95 % on MUC WSJ with 600 k words

Hand labeled Training set !!

Hand labeling 100 k words needs about 33 hours

learning approach to building domain specific search engines, IJCAI'99

- Machine learning for specialized search engines (Computer Science)
  - Spidering: reinforcement
  - Classifying documents in hierarchies - naïve Bayes
  - Information Extraction: HMMs
- Principe
  - Scrawl the web for CS labs
  - Convert .ps docs into text
  - Extract header and bibliographic information for indexing
  - Build a citation graph
  - Build a hierarchy

Cora : Information Extraction

- Extract relevant fields for paper identification:
- Authors, titles, affiliation, address, keywords, web, etc
- Hand made and learned HMMs structures

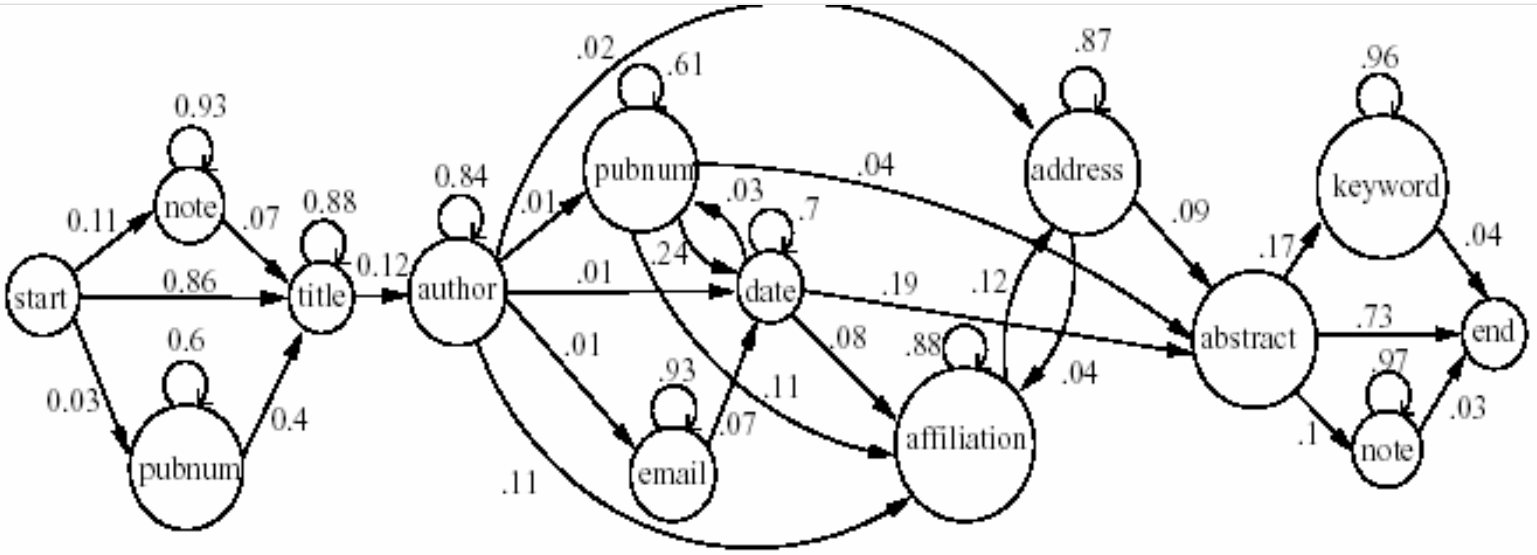
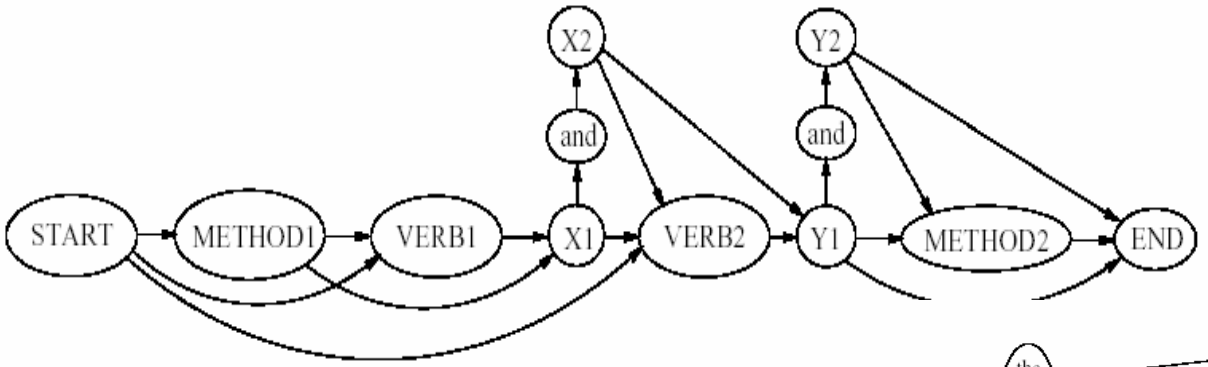
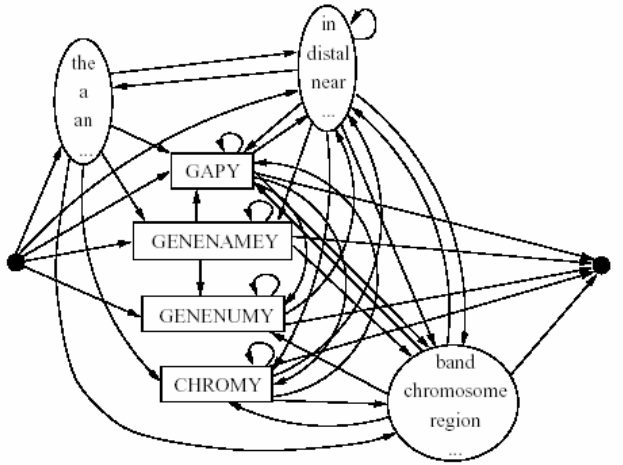


Figure 7. Example header HMM. Each state emits words from a class-specific multinomial distribution.

Extraction of relations (from Leek 97)

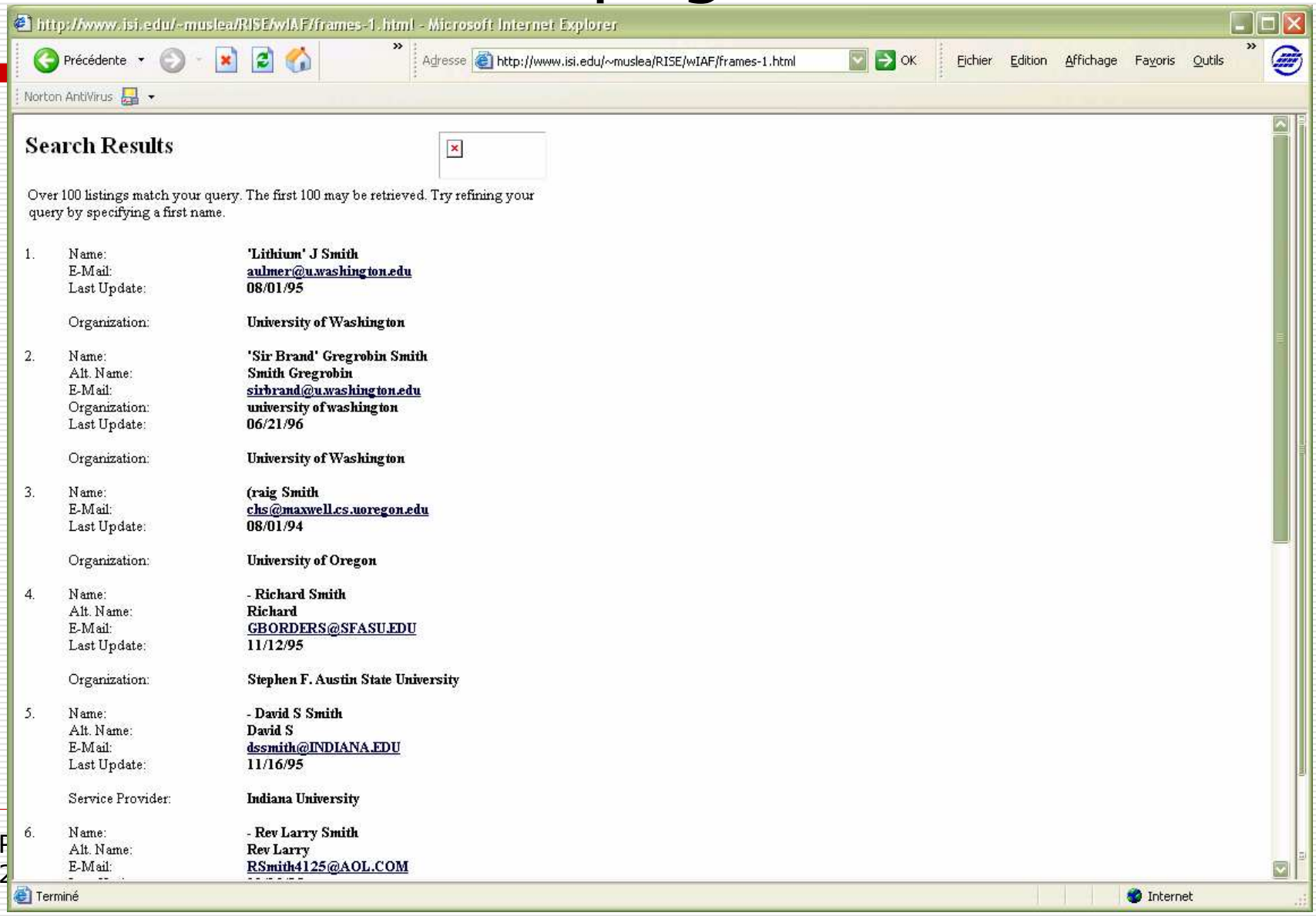
- Extract relations
- E.g. gene location: The gene encoding Bark2 mapped to mouse chromosome 5...





(a) Y (location)

Structured web pages



IE- Wrapers for Structured Web Pages -> Finite State machines

- Finite state automata
- Probabilistic Finite state automata
- Language = strings of tags (+sometimes text) from pages

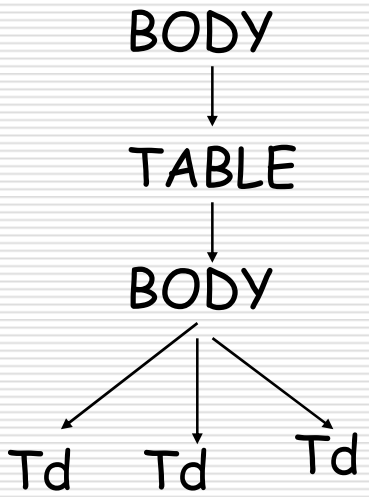
<HTML><BODY><TABLE>

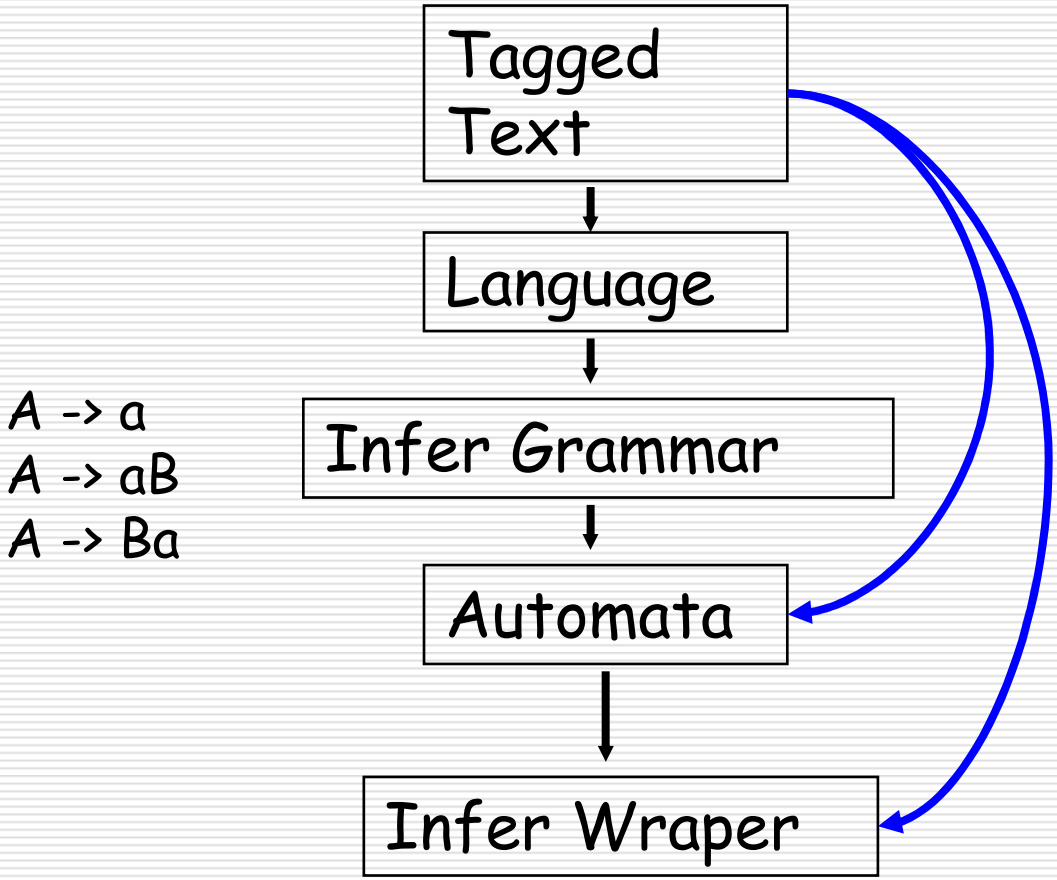
<TR>

<TD>Name1</TD><TD>Tel1<TD><TD>Ad1</TD>

<TR>

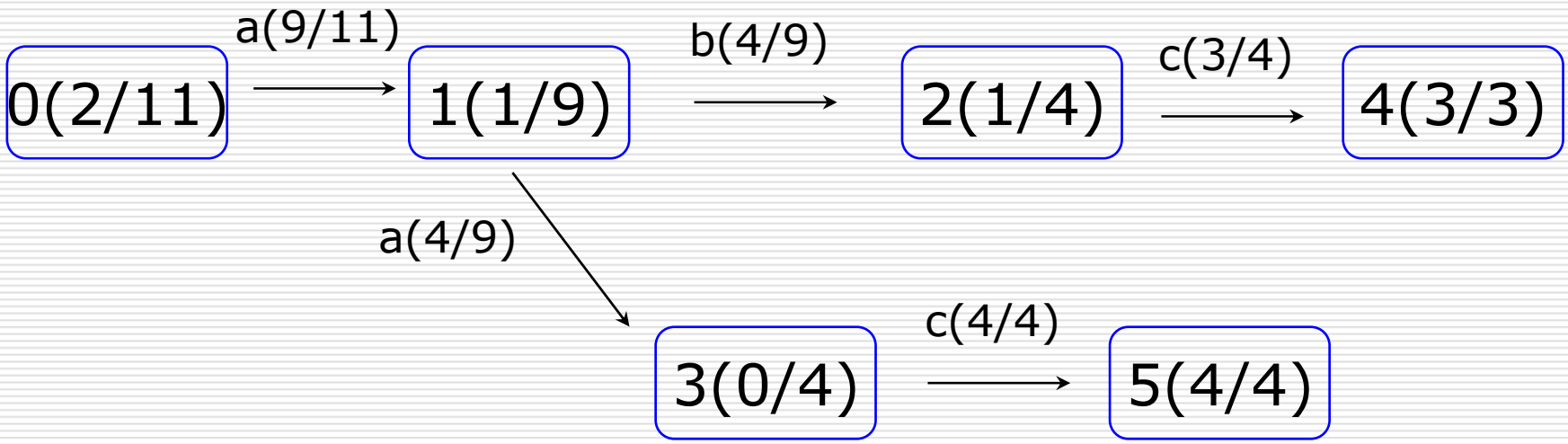
<TD>Name2</TD><TD>Tel1<TD><TD>Ad1</TD
>




Tagged Text
Language
Infer Grammar
Automata
Infer Wraper
A -> a
A -> aB
A -> Ba

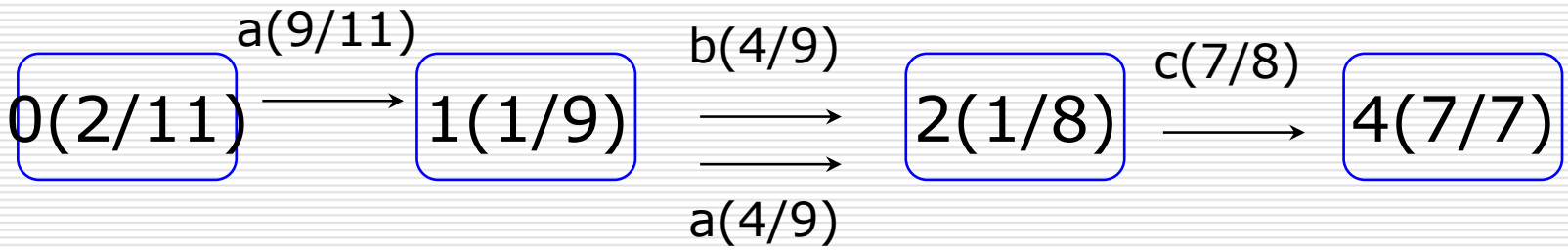
PFSA

- Probabilistic Prefix Tree Acceptor for language:
  - {λ,aac,aac,abc,aac,aac,abc,λ,a,ab,abc}



Fusion

- 2 states are equivalent if they have the same transition probability whatever the symbol is and if the destination states are also equivalent.



Learning PFSA – Alergia (Carrasco, Oncina 94)

- Start from A = PTA(I+)
  - covers all examples from training set I+
  - No generalization ability
- Generalize A
  - Explore all node pairs in PTA
  - Merge Similar nodes (α)
    - statistical test based on node statistics (transition proba., termination proba. are equal), destination stats are equivalent (recursive)
    - May change the language accepted

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 Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.

- 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