

---

# Moteur de recherche (SE)

---

Présentée par : **Quoc Anh LE**  
quocanh263@gmail.com

# Problématique

- On arrive 29,7 billion de sites web jusqu'à Février, 2007
- Informations deviennent plus diverses
- On aime commencer Internet à partir des sites connus comme Yahoo, Bbc,.. ou les SEs. (on maintient une liste des sites préférés mais pas très efficace)
- Augmentation du besoin de la recherche d'informations sur Internet (de 1500 requêtes reçues par jour en 1994 par les moteurs de recherche, le nombre de requêtes était 61 milliards menées au mois d'Août dans le monde)
- L'explosion d'informations continue encore sur Internet dans le futur...

# Préliminaire

- Ce rapport se restreint aux moteurs de recherche sur Internet.
- L'architecture général ainsi que les algorithmes présentées dans le rapport sont basées sur les publications concernant deux moteurs de recherche Google – le SE plus connu dans le monde et Nutch – un grand projet de SE source ouvert.
- Les algorithmes présentés sont à partir du domaine de Recherche d'Information restreindre à « web keywords searches ». Il y a aussi des autres algorithmes tels que le traitement d'image, la reconnaissance optique de caractères (ROC), etc.



# Réponse à deux questions

- Comment construit-on un SE très intelligent comme Google?
  - Réponse: Puissance des ordinateurs + Algorithmes Intelligents. Cependant, je présenterai seulement le deuxième aspect.
  
- Quelles sont donc les techniques d'IA qui seront appliquées aux SEs?
  - Réponse: Dans cette présentation, vous verrez un grand nombre de techniques d'IA qui sont appliquées aux SEs.



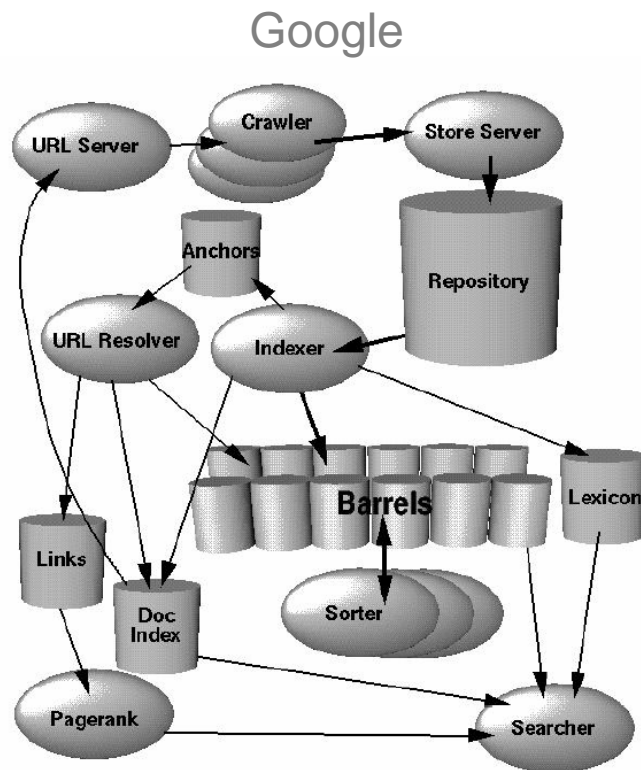
# Composition de la présentation

La présentation se compose de six parties principales:

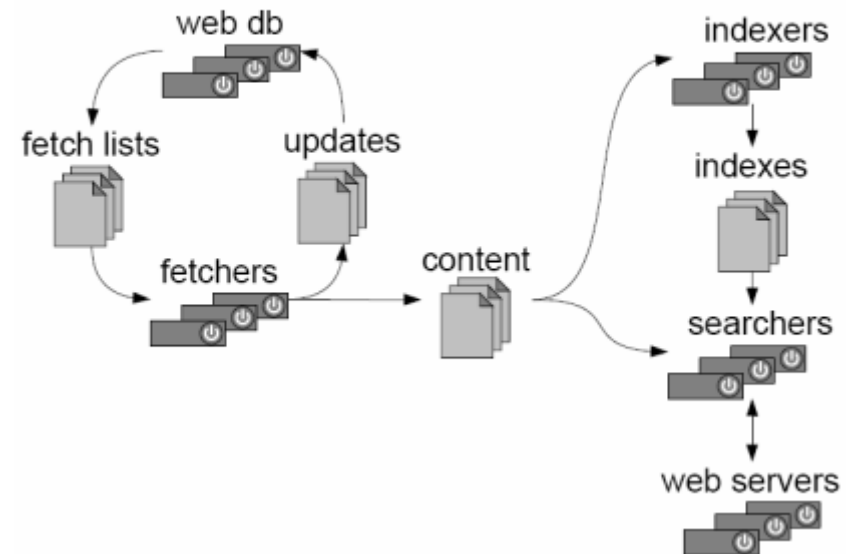
1. **Architecture générale**
2. **Technique de récupération des données**
3. **Technique d'indexation des données**
  1. *Indexation de textes*
  2. *Indexation de documents scannés*
  3. *Indexation d'images, vidéos*
  4. *Indexation d'informations sonores*
4. **Technique de la recherche**
  1. *Compréhension des requêtes*
  2. *Facilitation d'entrée des requêtes*
  3. *Classement des résultats*
  4. *Affichage des résultats (résumé automatique)*
5. **Implémentation**
6. **Conclusion**



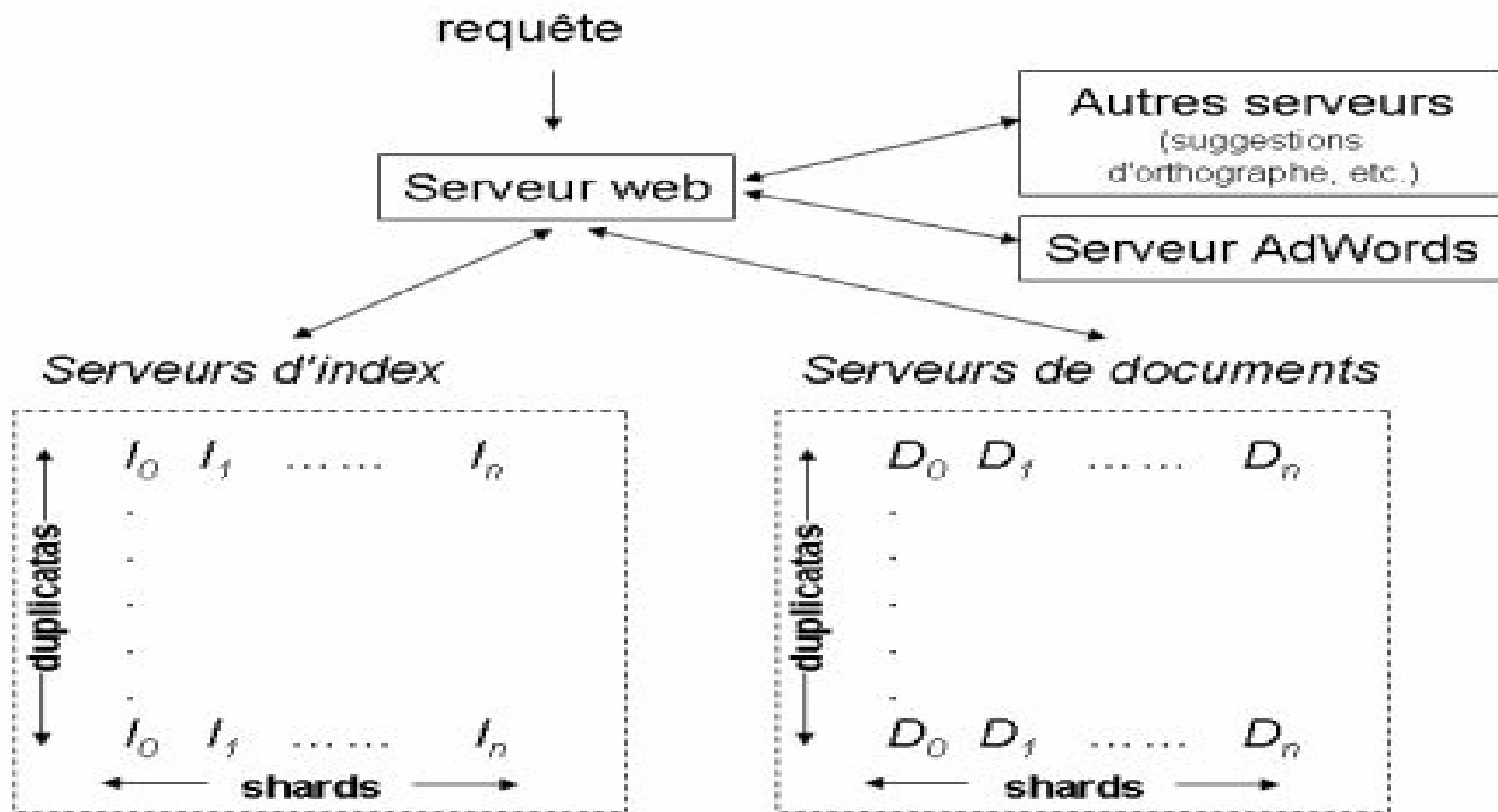
# Architecture



## Nutch Architecture



# Architecture



Duplicates pour le but des systèmes distribués et aussi des sauvetages

# Récupération des données

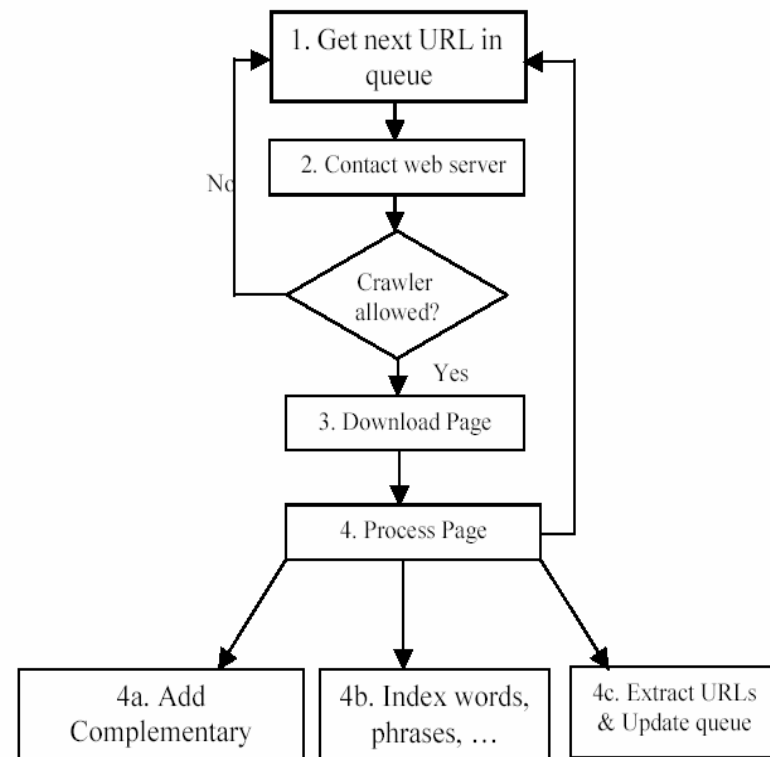
- Les SEs utilisent un « crawler » ou « spider » pour récupérer les données, plus exactement, les sites web. Je ne sais pas comment on dit en français ce mot, peut-être « exploration ».
- Une question: Les SEs peuvent-ils couvrir tous les sites web existant? La question est que non. Pourquoi? Je vous expliquerai tout de suite.





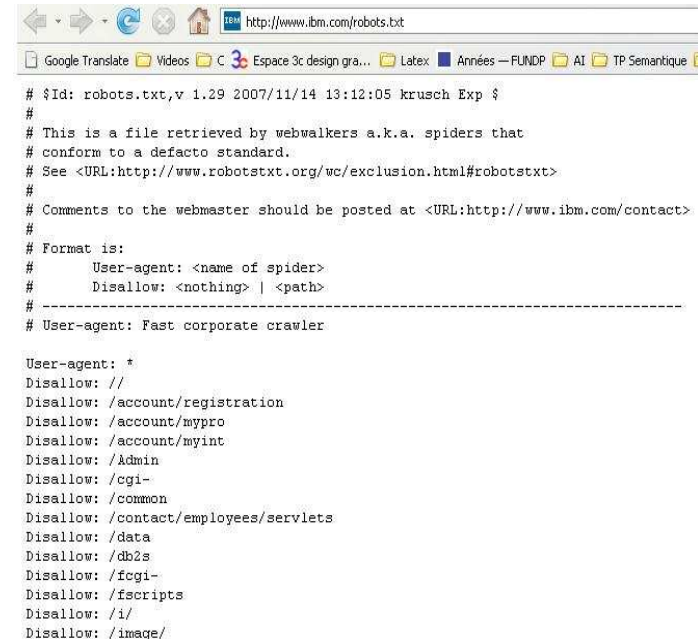
# [Récupération]: Crawler

- A partir des sites web connues, le SE extrait les liens dans ces sites et arrive à nouveaux sites à l'aide de l'algorithme de parcours en largeur BFS.
- Grâce aux liens dans les nouveaux sites, il va couvrir plusieurs sites possibles. Donc, il y a encore les sites web que le SE ne peut arriver.



# [Récupération]: règles

- Les SEs doivent respecter les règles lorsqu'ils se passent sur les sites web (normalement, les règles est stockés dans le fichier robots.txt). Pourquoi?
- Il y a deux raisons:
  - ❑ Eviter l'Attaque DDos
  - ❑ policy



```
# $Id: robots.txt,v 1.29 2007/11/14 13:12:05 krusch Exp $
#
# This is a file retrieved by webwalkers a.k.a. spiders that
# conform to a defacto standard.
# See <URL:http://www.robotstxt.org/wc/exclusion.html#robotstxt>
#
# Comments to the webmaster should be posted at <URL:http://www.ibm.com/contact>
#
# Format is:
#   User-agent: <name of spider>
#   Disallow: <nothing> | <path>
# -----
# User-agent: Fast corporate crawler

User-agent: *
Disallow: //
Disallow: /account/registration
Disallow: /account/mypro
Disallow: /account/myint
Disallow: /Admin
Disallow: /cgi-
Disallow: /common
Disallow: /contact/employees/servlets
Disallow: /data
Disallow: /db2s
Disallow: /fcgi-
Disallow: /fscripts
Disallow: /i/
Disallow: /image/
```

Le fichier **robots.txt** est un fichier texte contenant des commandes à destination des robots d'indexation des moteurs de recherche afin de leur préciser les pages qui peuvent ou ne peuvent pas être indexées. Ainsi tout moteur de recherche commence l'exploration d'un site web en cherchant le fichier *robots.txt* à la racine du site



# [Récupération]: règles

**CNIL - Les « CNIL » européennes précisent les règles applicables aux moteurs de recherche - Mozilla Firefox**

File Edit View History Bookmarks Tools Help

http://www.cnil.fr/index.php?id=2419

Google Translate Videos C Espace 3c design gra... Latex Années — FUNDP AI TP Semantique Travel RFI - Journaux franç... RADIO.BLOG Google Agenda SBB: Home - Online-F...

Gmail - Boîte de réception (304) - quoc... Faq.pdf (application/pdf Object) Google Docs - All items CNIL - Les « CNIL » européennes ...

**CNIL** La CNIL Découvrir Approfondir Agir Déclarer

► **Accueil** > **La CNIL** > **Actualité** > **En bref** > Les « CNIL » européennes précisent les règles applicables aux moteurs de recherche

Rechercher  OK

**Actualité**

- Agenda
- Communiqués
- Echos des séances
- En bref
- Tribune

**L'institution**

**Publications**

**Lettre InfoCNIL**

**Rencontres régionales**

**Recrutement**

**Marchés publics**

**Version Imprimable**

**Les « CNIL » européennes précisent les règles applicables aux moteurs de recherche**

11/04/2008 - En bref

Le 4 avril 2008, le groupe des 27 « CNIL » européennes, a adopté à l'unanimité un avis précisant les règles applicables aux moteurs de recherche. Cet avis résulte d'une concertation avec les acteurs majeurs du secteur. Il précise notamment que les données personnelles enregistrées par les moteurs de recherche, doivent être effacées au plus tard au bout de 6 mois.

Après avoir procédé à la consultation des principaux moteurs de recherche, le groupe des « CNIL » européennes dit « G29 » a adopté, le 4 avril 2008, un avis sur la protection des données à caractère personnel applicable aux moteurs de recherche.

Cet avis précise les conditions d'application des règles juridiques communautaires et formule des recommandations, qui doivent améliorer la protection et le droit des utilisateurs des moteurs de recherche.

**La loi européenne s'applique aux moteurs de recherche**

Le G29 souligne que les règles européennes de protection des données telles que définies par la directive 95/46, qui protègent les citoyens européens, s'appliquent aux moteurs de recherche, même si leur siège social se trouve en dehors de l'Union européenne.

**Les données enregistrées par les moteurs de recherche ne doivent pas être conservées plus de 6 mois**

Le G29 considère que les données personnelles enregistrées par les moteurs de recherche doivent être effacées dès que possible, et au plus tard au bout de 6 mois. Il rappelle à cet égard que les moteurs de recherche étant des « services de la société de l'information », ils ne sont pas concernés par la directive 2006/24/CE relative à la conservation des données, contrairement aux fournisseurs d'accès internet ou aux opérateurs de télécommunications. Ceci signifie que les moteurs de recherche ne sont pas légalement obligés de conserver des informations sur les connexions des utilisateurs.

En pratique, un moteur ne devrait pas conserver indéfiniment l'historique des requêtes effectuées et des sites consultés par un utilisateur. Cet historique peut révéler des informations très intimes, comme par exemple des problèmes conjugaux ou une opinion politique, à partir desquelles il est possible de déduire des habitudes de vie supposées ou un certain comportement. Il en va de la protection de notre vie privée.

**Les européens doivent être informés de leurs droits**

L'avis du G29 rappelle que les internautes doivent être clairement informés de l'ensemble de leurs droits en application de la directive 95/46. En particulier, l'information doit préciser les finalités des traitements, c'est-à-dire leur objectif, ainsi que les modalités d'exercice des droits d'accès, de rectification et de suppression des données.

**La CNIL**

**SUR LE SITE DU G29**

**L'avis du G29 sur les moteurs de recherches**  
(version anglaise)

Done

start Ou... Ox... Inb... CN... Art... Ex... Wi... nh... Acr... Re... L... W... 6:48 PM



# [Récupération]: Mis à jour

Pourquoi?

- Existence non stable des sites web
- Ajout, modification des sites web

Difficile:

- Grand nombre des sites, ex: jusqu'à le 27 juin 2006, Google a 450,000 serveurs placés dans tous les pays mais il a besoin encore au moins une semaine pour re-visiter tous les sites.

Solution:

- Priorité des sites: par exemple, pour les sites web des journaux en ligne, les blogs,...les informations sont mis à jour plus souvent que les autres sites donc ils doivent être re-visités plus tôt.



# [Récupération]: Priorité des sites web

Utilisant deux méthodes:

- Expérience accumulé après des fois de récupérations (par exemple, comparer des versions différentes d'un site après des fois consécutives de récupération pour estimer la vitesse de mis à jour sur ce site)
- Les caractères d'un site comme les mots de clé, la structure pour déterminer si un site est un journal ou pas (mais le SE doit être intelligent pour distinguer les sites réels et les sites déguisés)
- Autres méthodes: Espion, Statistiques de Alexa, google-statistic, etc

En fait, les journaux en ligne connus tels que bbc.com ou cnn.com sont mis à jour aux résultats de recherche de Google très rapidement, peut-être après des heures d'être mis sur Internet.



# [Récupération]: téléchargement

- Il n'a pas besoin de télécharger toute le page lorsque le SE veut vérifier les changements d'un page Web
- Les sites web seront stockés dans locaux hôtes (caches) et normalement compressés afin de diminuer l'espace de stockage

Voici donc un exemple de réponse HTTP :

```
HTTP/1.0 200 OK
Date : Sat, 15 Jan 2000 14:37:12 GMT
Server : Microsoft-IIS/2.0
Content-Type : text/HTML
Content-Length : 1245
Last-Modified : Fri, 14 Jan 2000 08:25:13 GMT
```

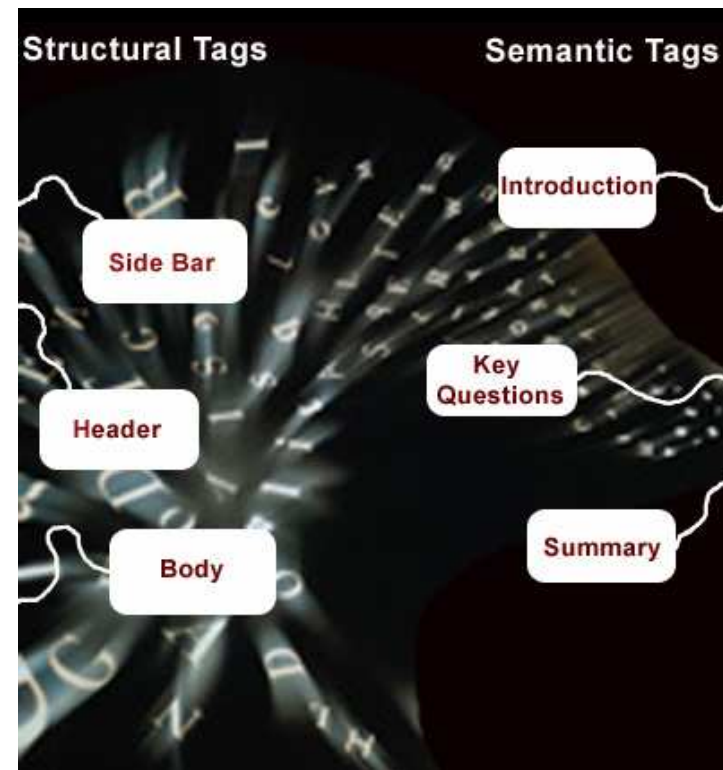






# Indexations des textes

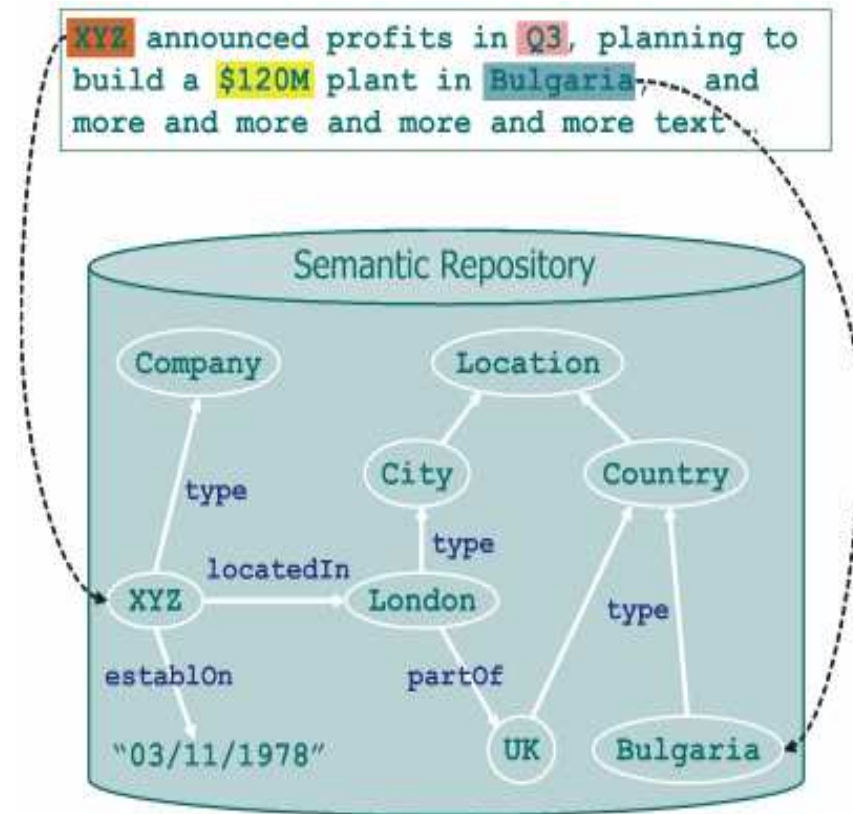
- Classement des textes selon leurs caractères (capitalisation, taille, italique, gras, font, position,...et grâce aux tags pour déterminer les mots de clé, les phrase de tête,...)





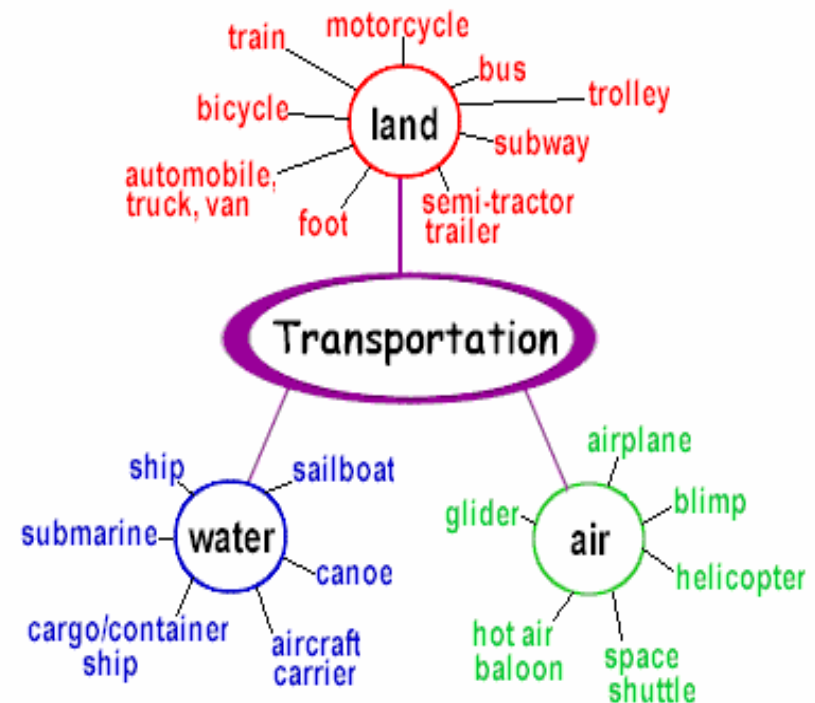
# Indexations des textes

- Construire un sémantique texte pour citer les relations entre les mots de recherche et entre les pages qui contiennent les mots de recherche.
- En utilisant un ensemble des dictionnaires (WordNet par exemple,...)



# Indexations des textes

## ■ Selon les thèmes



# [Index] Algorithme de Latent (LSA)

- Parce qu'il y a plusieurs manières pour qu'un utilisateur exprime son souhait par une phrase de la requête en utilisant les mots différents (synonyme) et parce qu'il y a aussi la plupart des mots ont plusieurs sens (polysémie)

Words:

	<u>Keyword</u>	<u>LSA</u>
Doctor—Doctor	1.0	1.0
Doctor—Physician	0.0	0.8
Doctor—Surgeon	0.0	0.7

Passages:

Doctors operate on patients  
Physicians do surgery.

Keywords 0, LSA .8

- Relier les mots avec les pages
- Déterminer quelle page est correspondante d'une requête concrète. Par exemple: Une page d'ordinateur Apple semble naturellement qu'elle se compose les termes tels que iMac ou iPod



Latent semantic indexing adds an important step to the document indexing process. In addition to recording which keywords a document contains, the method examines the document collection as a whole, to see which other documents contain some of those same words. LSI considers documents that have many words in common to be semantically close, and ones with few words in common to be semantically distant. This simple method correlates surprisingly well with how a human being, looking at content, might classify a document collection. Although the LSI algorithm doesn't understand anything about what the words mean, the patterns it notices can make it seem astonishingly intelligent. [source](#)

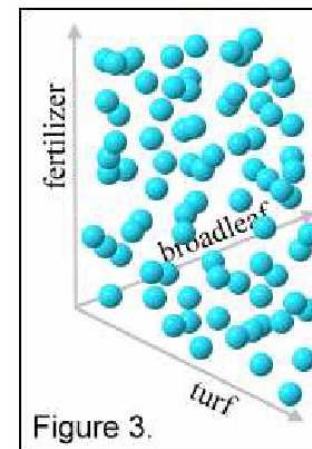
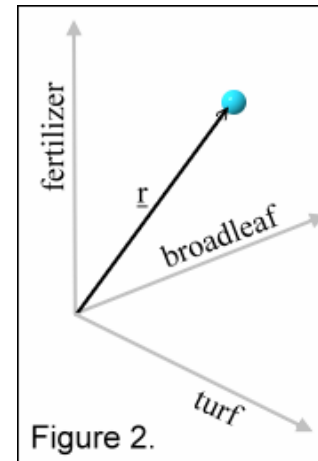
*L'idée principale: Le sens d'une phrase est approximatif de sommer les sens de ses mots*

$$m(\text{phrase}) = m(\text{mot1}) + m(\text{mot2}) + .. + m(\text{motn})$$



# [Index] Algorithme de Latent[1]

- On peut localiser un objet dans l'espace 3D ( $x, y, z$ )
- Appliqué cette idée pour localiser une page web dans une espace 3D. Imaginez que cette page contient trois mots *fertilizer*, *broadleaf* et *turf* qui se trouvent certains fois dans cette page. On peut représenter la position de la page dans l'espace *fertilizer-broadleaf-turf* par le vecteur  $r$  comme le Figure 2.
- On peut aussi présenter la position de toutes les pages qui contiennent ces trois mots comme le Figure 3 - « espace de termes ». Donc, chaque vecteur détermine combien de fois ces trois mots se trouvent dans la page correspondante.
- Maintenant, on n'utilise plus l'espace 3D mais l'espace  $nD$  ( $n$  est le nombre des mots possibles dans un grand dictionnaire) pour représenter tous les mots et toutes les pages web dans la base de données.



## [Index] Algorithme de Latent [2]

- Calculer les fréquences des mots dans les pages (millions de mots et billions de pages).
- Construire une table  $a(i,j) = 1$  si le mot  $i$  se trouve dans le document  $j$  et égal zéro dans l'autre cas. C'est clair que la table contient le note 0 beaucoup plus fois que le note 1.
- Dans ce modèle, tous les format (capitalisation,...) sont laissés tomber. Les prépositions, conjonction, les verbes communs, pronoms, articles et les adjectifs communs sont aussi laissés tomber. Enfin, les fins communes des mots sont supprimés (ce processus est appelé stemming, présenté plus tard).
- Un facteur local de pondération d'un mot est appliqué de façon que ce mot apparaît plusieurs fois dans une page va peser plus l'autre mot qui apparaît une fois.
- Un facteur global de pondération d'un mot est appliqué de façon que ce mot apparaît dans un petit nombre des pages va peser plus l'autre mot qui apparaît dans un grand nombre des pages. (donc, plus significantes).

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	.....	pn
word 1	0	0	0	0	0	0	0	0	0	0	0		0
word 2	0	0	0	0	0	0	0	0	0	0	0		0
word 3	0	0	0	0	0	0	0	0	0	0	0		0
word 4	0	1	0	0	0	0	0	0	0	0	0		0
word 5	0	0	0	0	0	0	0	0	0	0	0		0
word 6	0	0	0	0	0	0	0	0	0	0	0		0
word 7	0	0	0	0	0	0	0	0	0	0	0		0
word 8	0	0	0	0	0	0	1	0	0	0	0		0
word 9	0	0	0	0	0	0	0	0	0	0	1		0
word 10	0	0	0	0	0	0	0	0	0	0	0		0
word 11	0	0	0	0	0	0	0	0	0	0	0		0
⋮													
word n	0	0	0	0	0	0	0	0	0	0	0		0

Figure 6.

# [Index] Algorithme de Latent [3]

d1: *Shipment of gold damaged in a fire.*

d2: *Delivery of silver arrived in a silver truck.*

d3: *Shipment of gold arrived in a truck.*

**Problem:** Use Latent Semantic Indexing (LSI) to rank these documents for the query *gold silver truck*.

**Step 1:** Score term weights and construct the term-document matrix **A** and query matrix:

Terms ↓	d1 ↓	d2 ↓	d3 ↓	q ↓
a	1	1	1	0
arrived	0	1	1	0
damaged	1	0	0	0
delivery	0	1	0	0
fire	1	0	0	0
gold	1	0	1	1
in	1	1	1	0
of	1	1	1	0
shipment	1	0	1	0
silver	0	2	0	1
truck	0	1	1	1

**A =**

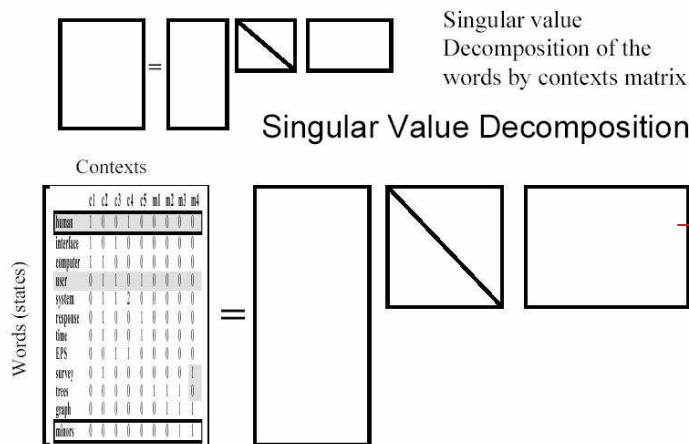
**q =**



# [Index] Algorithme de Latent [4]

**Step 2:** Decompose matrix **A** matrix and find the **U**, **S** and **V** matrices, where

$$A = USV^T$$



$$U = \begin{bmatrix} -0.4201 & 0.0748 & -0.0460 \\ -0.2995 & -0.2001 & 0.4078 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.1576 & -0.3046 & -0.2006 \\ -0.1206 & 0.2749 & -0.4538 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.4201 & 0.0748 & -0.0460 \\ -0.2626 & 0.3794 & 0.1547 \\ -0.3151 & -0.6093 & -0.4013 \\ -0.2995 & -0.2001 & 0.4078 \end{bmatrix}$$

$$S = \begin{bmatrix} 4.0989 & 0.0000 & 0.0000 \\ 0.0000 & 2.3616 & 0.0000 \\ 0.0000 & 0.0000 & 1.2737 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.4945 & 0.6492 & -0.5780 \\ -0.6458 & -0.7194 & -0.2556 \\ -0.5817 & 0.2469 & 0.7750 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \\ -0.5780 & -0.2556 & 0.7750 \end{bmatrix}$$

Soit  $M$  une matrice  $m \times n$  dont les coefficients appartiennent au corps  $K$ , où  $K = \mathbb{R}$  ou  $K = \mathbb{C}$ . Alors il existe une factorisation de la forme :

$$M = U \Sigma V^*$$

avec  $U$  une matrice unitaire  $m \times m$  sur  $K$ ,  $\Sigma$  une matrice  $m \times n$  dont les coefficients diagonaux sont des réels positifs et tous les autres sont nuls (c'est donc une matrice diagonale dont on impose que les coefficients soient positifs), et  $V^*$  est la matrice adjointe à  $V$ , donc une matrice unitaire  $n \times n$  sur  $K$ . On appelle cette factorisation la *décomposition en valeurs singulières* de  $M$ .

La matrice  $V$  contient un ensemble de vecteurs de base orthonormés pour  $M$ , dits « d'entrée » ou « d'analyse » ;

La matrice  $U$  contient un ensemble de vecteurs de base orthonormés pour  $M$ , dits « de sortie » ;

La matrice  $\Sigma$  contient les valeurs singulières de la matrice  $M$ .

Une convention courante est de ranger les valeurs  $\Sigma_{i,i}$  par ordre décroissant. Alors, la matrice diagonale  $\Sigma$  est déterminée de façon unique par  $M$  (mais  $U$  et  $V$  ne le sont pas).



# [Index] Algorithme de Latent [5]

A l'aide de la réduction dimensionnelle, on transforme les matrices 3D en les matrices 2D mais tient encore les sens généraux des informations.

**Step 3:** Implement a Rank 2 Approximation by keeping the first columns of **U** and **V** and the first columns and rows of **S**.

$$\begin{aligned}
 \mathbf{U} \approx \mathbf{U}_k &= \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} & \mathbf{S} \approx \mathbf{S}_k &= \begin{bmatrix} 4.0989 & 0.0000 \\ 0.0000 & 2.3616 \end{bmatrix} \\
 \mathbf{V} \approx \mathbf{V}_k &= \begin{bmatrix} -0.4945 & 0.6492 \\ -0.6458 & -0.7194 \\ -0.5817 & 0.2469 \end{bmatrix} & \mathbf{V}^T \approx \mathbf{V}_k^T &= \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \end{bmatrix}
 \end{aligned}$$

$k = 2$





# [Index] Algorithme de Latent [6]

**Step 4:** Find the new document vector coordinates in this reduced 2-dimensional space.

Rows of **V** holds eigenvector values. These are the coordinates of individual document vectors, hence

d1(-0.4945, 0.6492)

d2(-0.6458, -0.7194)

d3(-0.5817, 0.2469)

**Step 5:** Find the new query vector coordinates in the reduced 2-dimensional space.

$$\mathbf{q} = \mathbf{q}^T \mathbf{U}_k \mathbf{S}_k^{-1}$$

Note: These are the new coordinate of the query vector in two dimensions. Note how this matrix is now different from the original query matrix **q** given in **Step 1**.

$$\mathbf{q} = \mathbf{q}^T \mathbf{U}_k \mathbf{S}_k^{-1}$$

$$\mathbf{q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \begin{bmatrix} 1 \\ 4.0989 & 0.0000 \\ 1 \\ 0.0000 & 2.3616 \end{bmatrix}$$

$$\mathbf{q} = \begin{bmatrix} -0.2140 & -0.1821 \end{bmatrix}$$



# [Index] Algorithme de Latent [7]

**Step 6:** Rank documents in decreasing order of query-document cosine similarities.

$$\text{sim}(q, d) = \frac{q \bullet d}{\|q\| \|d\|}$$

$$\text{sim}(q, d_1) = \frac{(-0.2140)(-0.4945) + (-0.1821)(0.6492)}{\sqrt{(-0.2140)^2 + (-0.1821)^2} \sqrt{(-0.4945)^2 + (0.6492)^2}} = -0.0541$$

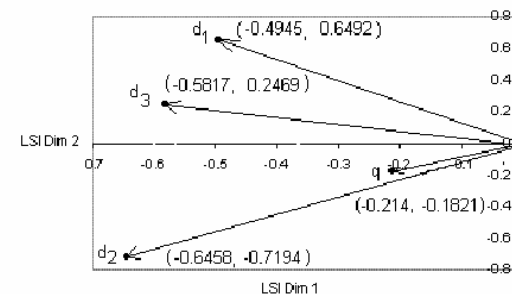
$$\text{sim}(q, d_2) = \frac{(-0.2140)(-0.6458) + (-0.1821)(-0.7194)}{\sqrt{(-0.2140)^2 + (-0.1821)^2} \sqrt{(-0.6458)^2 + (-0.7194)^2}} = 0.9910$$

$$\text{sim}(q, d_3) = \frac{(-0.2140)(-0.5817) + (-0.1821)(0.2469)}{\sqrt{(-0.2140)^2 + (-0.1821)^2} \sqrt{(-0.5817)^2 + (0.2469)^2}} = 0.4478$$

Ranking documents in descending order

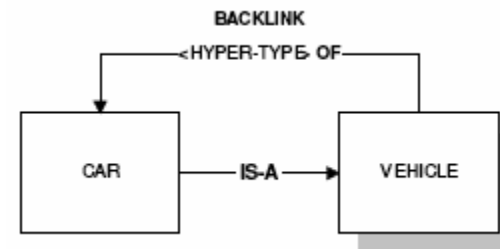
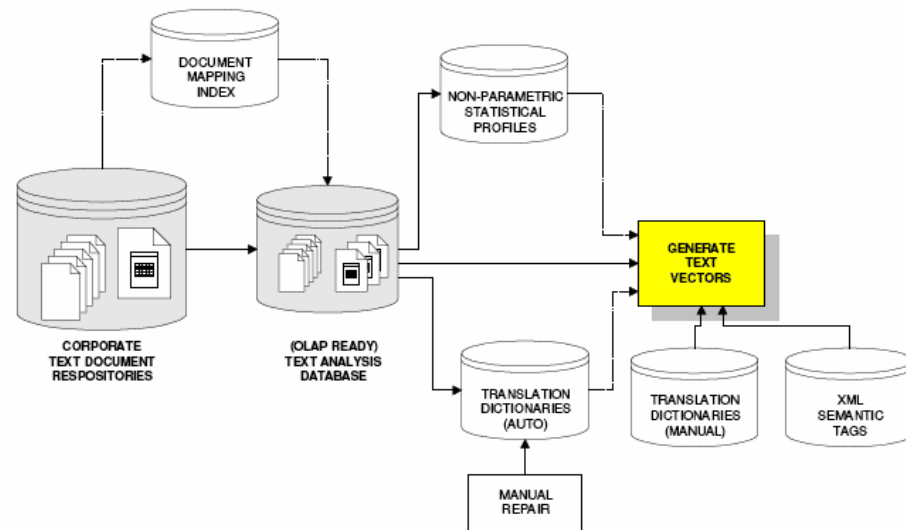
$$d_2 > d_3 > d_1$$

We can see that document d2 scores higher than d3 and d1. Its vector is closer to the query vector than the other vectors. Also note that Term Vector Theory is still used at the beginning and at the end of LSI.



# Indexations des textes

- Classement des textes selon leurs caractères
- Sémantique
- Autres: synonymes, IS-A (CAR est une sous-classe de VEHICLE,...), polysémies, etc
- Résumé automatique (présenté plus tard)



# [Indexations] Documents scannés



Carlos Osorio  
ASSOCIATED PRESS

(enlarge photo)

Google's Book Search project has hired hundreds of librarians to create digital versions of all the world's books. Courtney Mitchel works in dim lighting to protect an antique Bible.



GOOGLE

## Page by page, Google's digital book project is growing

By Natasha Robinson  
ASSOCIATED PRESS

Monday, April 28, 2008

ANN ARBOR, Mich. — In a dimly lit back room on the second level of the University of Michigan library's book-shelving department, Courtney Mitchel helped a giant desktop machine digest a rare, centuries-old Bible.

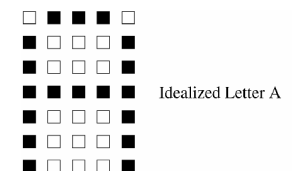
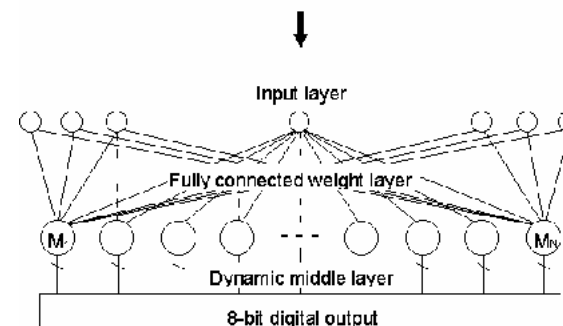
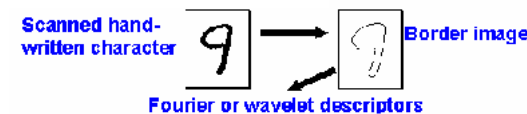
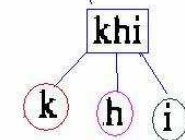
Mitchel is among hundreds of librarians from Minnesota to England making digital versions of the most fragile of the books to be included in Google Inc.'s Book Search, a portal that will eventually lead users to all the estimated 50 million to 100 million books in the world.

A final click of the mouse sends each digitized book to Google for optical character recognition processing, which makes the text searchable. Google returns a copy of the images and data to the library and posts one to the Web.

# [Indexations] Documents scannés

- Reconnaissance des caractères d'imprimerie
  - Un simple algorithme:
    - Texte -> mots -> mot -> lettre -> matrice 7x5 (A)
    - For (i=0;i<24;i++)
      - Temp = différence(matriceA, matrice(i));
      - If(temp > seuil)
        - If(min>temp)
          - Min = temp;
          - K = i;
    - Lettre = matrice(K) => Sortir code d'ASCII
    - Autre algorithme: Réseau de neurones
- Faire l'indexation comme les textes

Hiện nay, trong mảng nghiên cứu tiếng Anh mới chỉ có các sách còn hầu như chưa có cuốn Việt xác cao. Trong **khi** đó, một cuốn Anh là rất cần thiết. Nó làm nhiều thời gian và công sức mỗi người tương đương trong tiếng Anh có vốn từ vựng phong



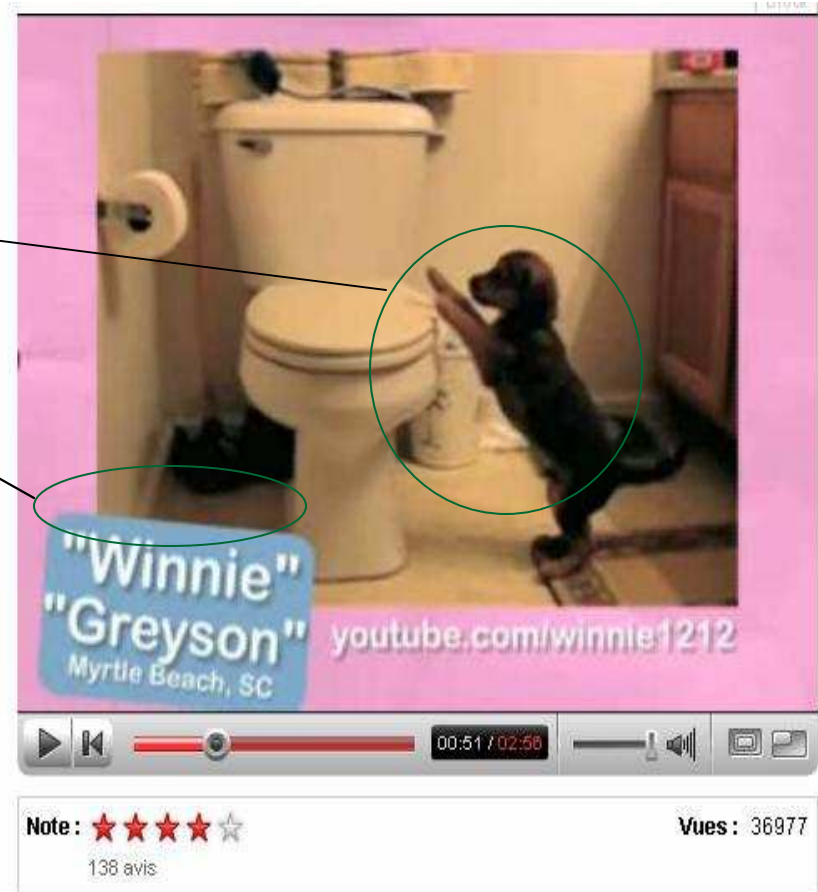
(00111001)<sub>2</sub>=57<sub>10</sub>

ASCII Output

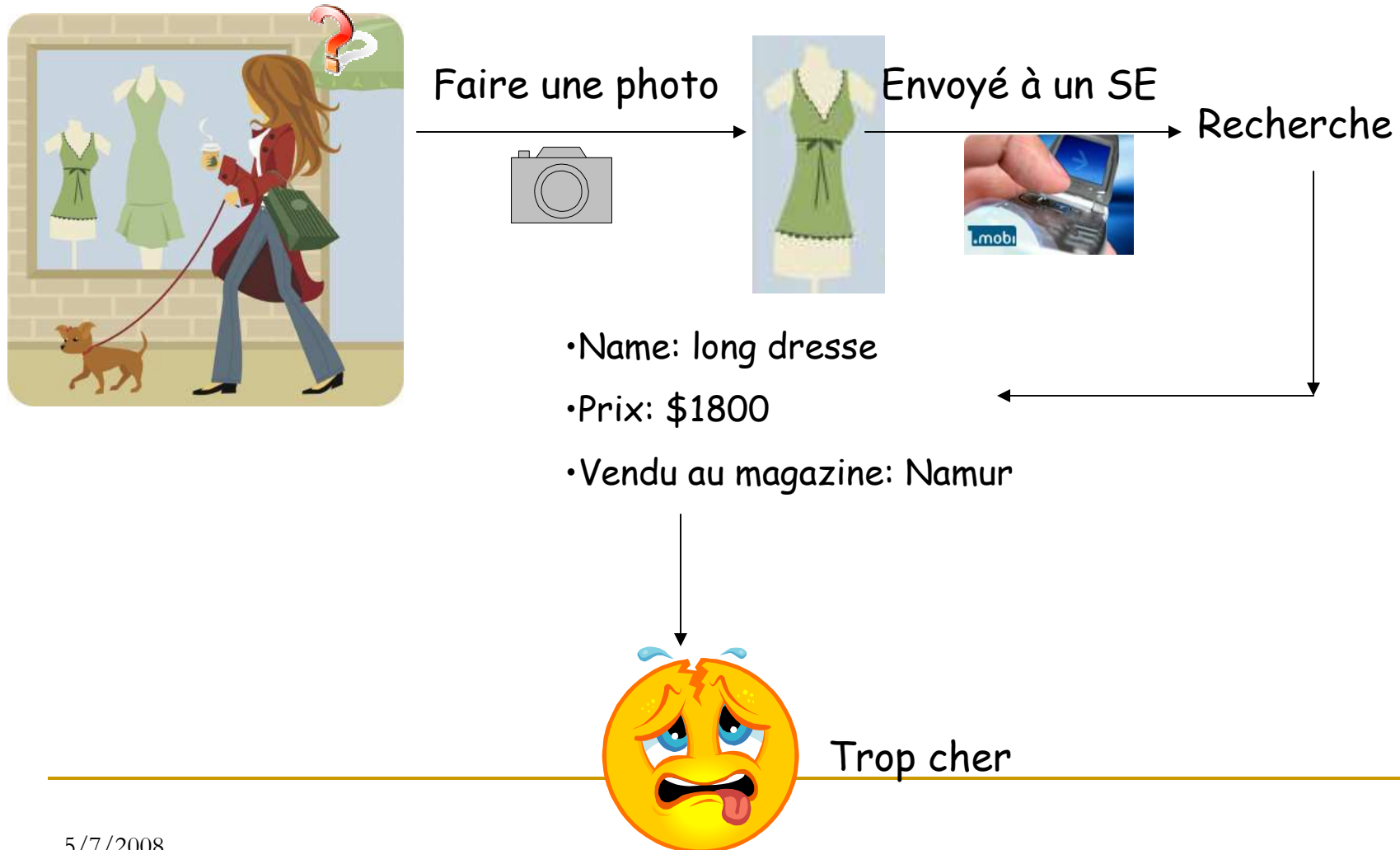
# Un exemple de l'index de vidéos par le contenu

Requête:

« chien + Winnie »

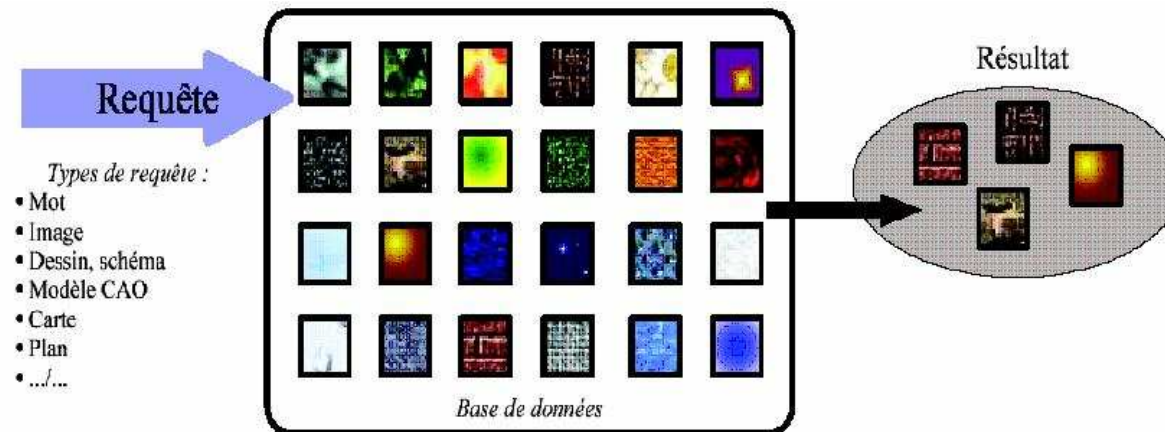


# Un exemple de l'index d'image par le contenu

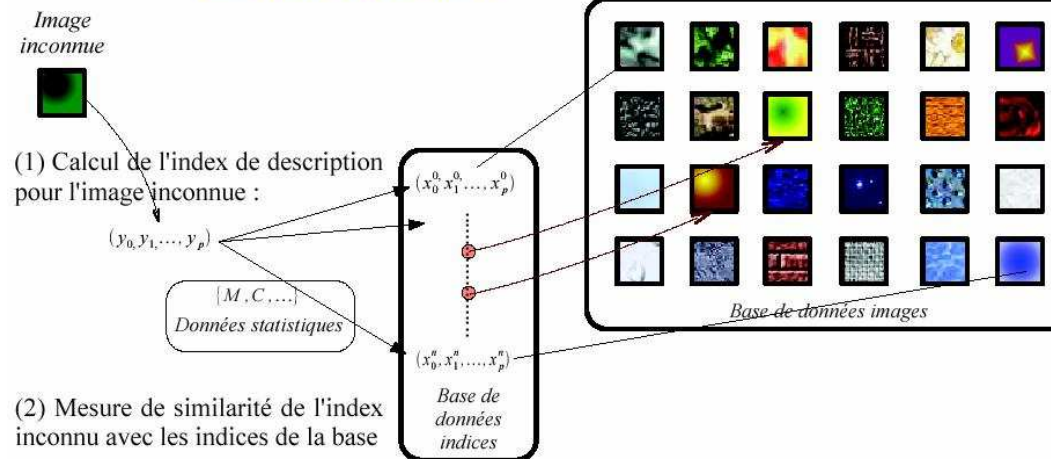




# Index d'images



## La requête est une image

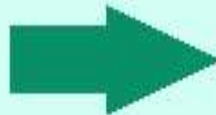


(3) Résultat : adresse des meilleurs images au sens de la mesure de similarité



# Index d'images

- Requête par un exemple : recherche d'images semblables

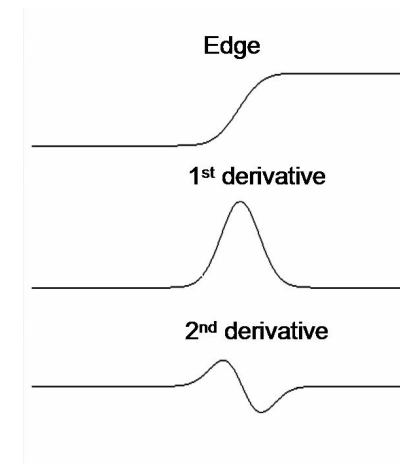
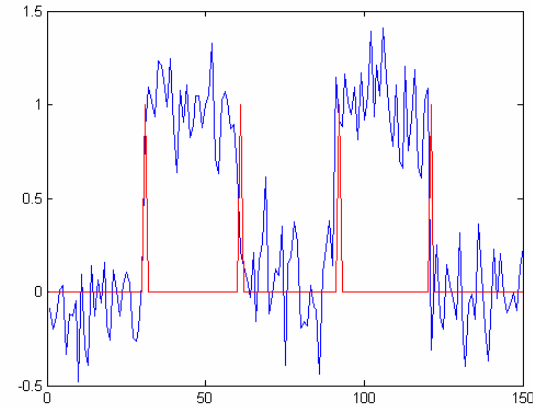


- Recherche d'un objet, ou d'un type d'objets particulier



# Index d'images - Algorithmes

1. A l'aide d'images structures (contours, géométrie, reconstruction 3d <-> 2d,..)
2. Utilisation des points d'intérêt
3. Comparaison des modèles



# Index de séquences vidéos

- Un vidéo est essentiellement une série des images consécutives. Donc, il y a deux approches pour chercher et reconnaître les objets dans un vidéo
  - Traiter chaque image indépendante
  - Traiter la relation entre les images consécutives
- Ce problème fait actuellement l'objet de recherches très abondantes dans le domaine du traitement d'images et de la vision par ordinateur
- Indexation de vidéos, c'est-à-dire qu'il se propose d'attacher à une vidéo un ensemble de descripteurs de leur contenu, dans le but de mesurer la ressemblance avec les descripteurs correspondant à la requête



# Index de vidéos –

Traitement de chaque image indépendante

1. Une variable d'intervalle  $d$
2. Après chaque temps  $d$ , on extrait une image du vidéo
3. Traiter cette image comme l'indexation d'images qu'on a déjà abordé avant

=> Perdre beaucoup de temps, donc on doit chercher l'autre méthode pour extraire les images-clefs, c'est-à-dire d'images « les plus représentatives »



Interface de l'outil de segmentation vidéo développé à l'INRIA Rhône-Alpes – projet MOVI

# Index de vidéos –

Traitement la relation entre les images consécutives

- Basé sur le mouvement des objets dans un vidéo pour extraire les objets souhaités
  
- Algorithmes de la différence des images:
  - Soustraction de l'arrière-plan
  - Soustraction de deux images consécutives
  - Soustraction de trois images consécutives (*double-différence*)

Source: Quoc Anh LE, Proceeding of the First Young Vietnamese Scientists Meeting (YVSM'05), Nha Trang, Vietnam, June 12-16, 2005



# Index de vidéos –

Traitement la relation entre les images consécutives – soustraction de l'arrière-plan

- Appliqué seulement pour les vidéos qui a l'arrière-plan constant ou l'arrière-plan très peu de changement avec le temps

1. Extraire l'image de l'arrière-plan sans objets
2. Soustraire l'arrière-plan de l'image présente (avec un seuil donné)

$$d(i, j) = \begin{cases} 1 & \text{if } |f_1(i, j) - f_2(i, j)| > T, \text{ where } T \text{ is a suitable threshold} \\ 0 & \text{otherwise} \end{cases}$$

3. Reconnaître les objets extraits de façon de comparer avec les objets souhaités avec un seuil donné
4. Mis à jour l'arrière-plan

Source: Quoc Anh LE, Proceeding of the First Young Vietnamese Scientists Meeting (YVSM'05), Nha Trang, Vietnam, June 12-16, 2005



# Algorithme « *double-difference* »



$$I(i,j) = \text{Binaire}(\text{Image\_originale}, \text{Seuil})$$

To explain the *double-difference* algorithm, we can consider the sequence of binary  $\{I_m\}$ : the *difference-image*  $D_m$  is defined as:

$$D_n(i,j) = |I_n(i,j) - I_{n-1}(i,j)| \quad (2)$$

The *double-difference image* is obtained by performing a logical AND between pixels belonging to two subsequent *difference-images*, thresholded by a threshold  $T$ :

$$DD_n(i,j) = \begin{cases} 1 & \text{if } (D_{n+1}(i,j) > T) \wedge (D_n(i,j) > T) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Objects}(i,j) = DD(i,j) * \text{Image\_originale}(i,j)$$

- Extraire chaque objets de l'image Objects
- Comparer ces objets avec l'objet de la requête en utilisant un seuil donné



# Indexation d'Audios

- A l'heure actuelle: Basé sur les textes autour qui décrivent le contenu (très mauvais résultats)
- Dans le futur: Basé sur le contenu de façon de transformer le son aux paroles => index
  - L'algorithme est extrêmement dur
  - Je ne sais pas du tout ce domaine

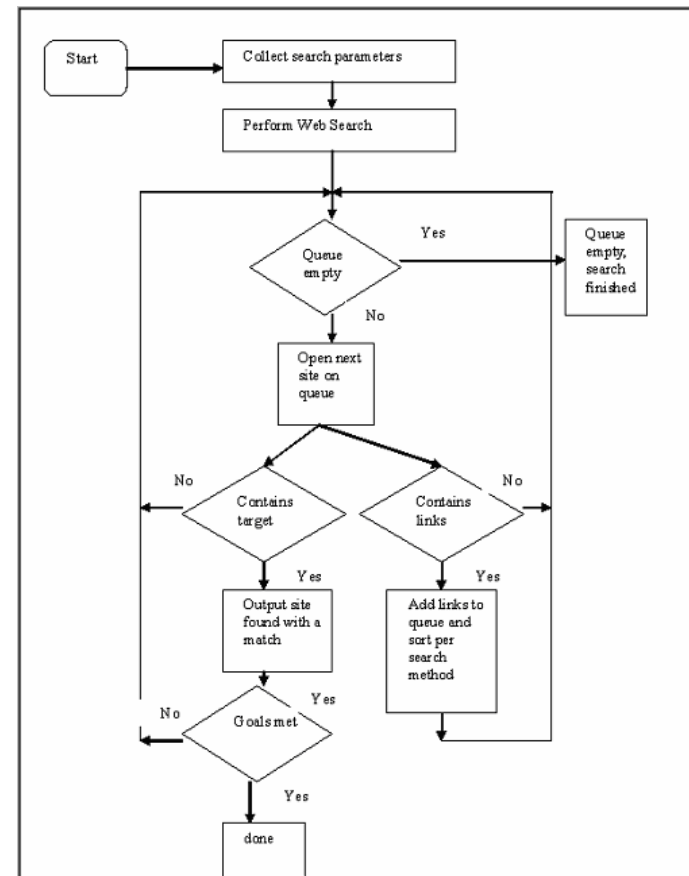




# La recherche

- Le temps de la recherche est au plus 5 seconds « *Les attentes des clients sont devenues plus exigeantes aujourd'hui* ».
- Les utilisateurs regardent seulement dix premiers résultats « *si vous ne trouvez pas exactement ce que vous attendez dans les trois premiers résultats, c'est qu'il y a un problème* ».

⇒ Il faut classer les résultats



# Exemple

1. Parse the query.
2. Convert words into wordIDs.
3. Seek to the start of the doclist in the short barrel for every word.
4. Scan through the doclists until there is a document that matches all the search terms.
5. Compute the rank of that document for the query.
6. If we are in the short barrels and at the end of any doclist, seek to the start of the doclist in the full barrel for every word and go to step 4.
7. If we are not at the end of any doclist go to step 4.  
Sort the documents that have matched by rank and return the top k.

Figure 4. Google Query Evaluation

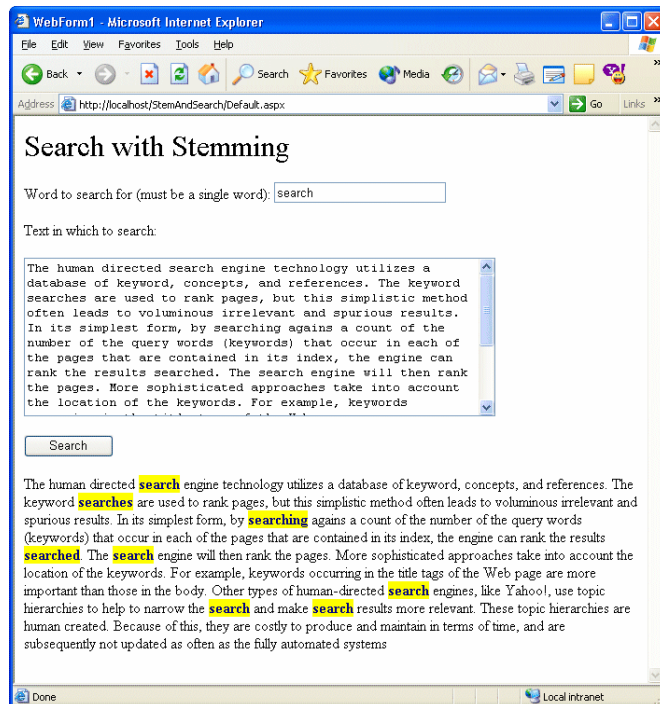


# Analyse des requêtes

- Texte sémantique (suggestion, faute d'orthographe,...)
- Opérateurs AND, OR, NOT
- Synonyme (car = automobile,...)
- Les mots proches (ebook, manual, guide, tips, report, tutorial, etc. )
- Sémantique
- Il faut remplacer « *donnez moi ce que j'ai tapé* » par « *donnez moi ce que je souhaite trouver* »



# Analyse des requêtes – Stemming



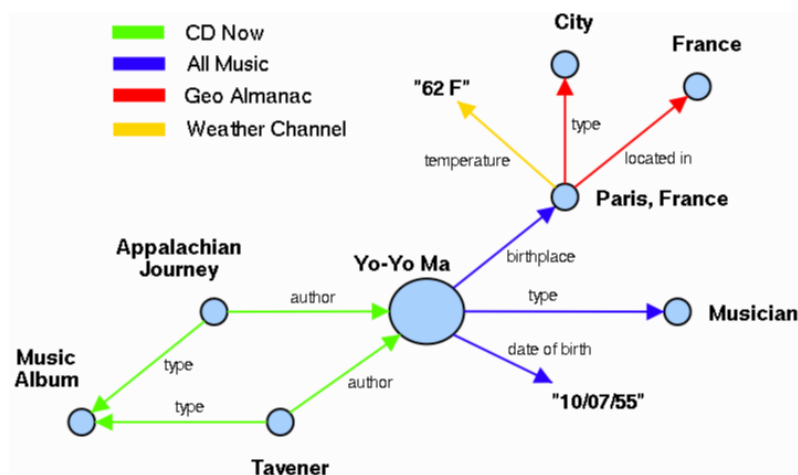
```
information -> inform
presidency -> presid
presiding   -> presid
happiness   -> happi
happily     -> happi
discouragement -> discourag
battles     -> battl
```

And here is our sample story as it appears to the stemmer:

```
o'neill criticizes europe grants treasury
secretary paul o'neill expressed irritation
european countries refused US proposal boost
direct grants rich nations poor countries
bush administration pushing plan increase amount
direct grants world bank poorest nations
assistance loans nations
```



# [Recherche] Sémantique



Text Search Results

Semantic Search Augmentation

Information from AllMusic:

- Top Album: [Soul of the Tango](#)
- [Appalachian Journey](#)
- [Simply Baroque](#)
- [Trombones](#)
- [Portrait of Yo-Yo Ma](#)
- Biography: Yo-Yo Ma was the cellist's foremost contemporary proponent, while primarily a classical performer, he also made a number of highly successful crossover recordings. Born October 7, 1955 to Chinese parents living in Paris, he began playing...
- [See full bio](#)

Related Activities:

- [W3C Semantic Web Activity](#)

Related Recommendations:

- [Rachmaninoff's Lullabies \(CD\)](#)
- [Mozart and Gounod: Symphonies](#) - 12 February 1999, Ralph Smith, Oia Lantz

Related W3C Working Drafts:

- [RDF Model Syntax](#) - 14 February 2002, Frank Hees
- [RDF Primer](#) - 19 March 2002, Frank Hees
- [RDF Test Cases](#) - 19 November 2001, Art Beckett
- [Semantic Interpolation for Search](#) - 16 November 2001, Lutz Van

Related Mailing Lists:

- [www.rdf.org](#)
- Sep 2001 to April 2002 (107 msgs)

Shop@AOL:

- [800 Com Music: Soul Of The Tango...](#)
- [Appalachian Journey / Yo-Yo Ma, Edgar...](#)
- [Yo-Yo Ma: Made In America \\$11.82](#)
- [800 Com Music: Brahms Sonatas For...](#)
- [Grieg's Elégies / Yo-Yo Ma, Aron...](#)
- [More Shop@AOL](#)

Concert tickets from TicketMaster:

- [Silk Road Project With Yo-Yo Ma/Cello](#) On 5/12/02 at Seattle, WA
- [Silk Road Project With Yo-Yo Ma/Cello](#) On 5/13/02 at Seattle, WA
- [Seattle Symphony Silk Road Project](#) On 5/14/02 at Seattle, WA
- [Silk Road Project With Yo-Yo Ma/Cello](#) On 5/15/02 at Seattle, WA
- [Seattle Symphony Silk Road Project](#) On 5/16/02 at Seattle, WA
- [More TicketMaster concerts](#)



# Adwords

- « AdWords » provient de « Ad » pour *Advertising* : Publicité et *Words*: mots (*wiki*)
- Il n'influence pas les résultats retournés aux utilisateurs.
- SEs font apparaître les AdWords lors que le contenu des meilleurs résultats est correspondant au contenu des AdWords (une bonne idée pour que les SEs maintiennent leur qualité de recherche and ils gagnent encore de l'argent)

# Algorithme de suggestion

1. Généralement, les données historiques sont les meilleures prédictions. Enregistrer toutes les requêtes qu'un utilisateurs sont entrées et les utiliser comme une liste des suggestions. Donc, lorsqu'un utilisateur type « A », la machine suggère les plus fréquents réponses commençant avec « A ».
2. La liste de suggestions est récupérée des autres utilisateurs similaires.
3. Certains cas, l'ordre les suggestions dans la liste est prioritaire de façon plus récente or plus fréquente or alphabétique.



The screenshot shows the Google Experimental LABS logo at the top. Below it, a search bar contains the text 'google'. To the right of the search bar, a list of suggestions is displayed, each followed by the number of results. The suggestions are: 'google earth' (114,000,000 results), 'google maps' (121,000,000 results), 'google.com' (82,400,000 results), 'google map' (381,000,000 results), 'google video' (720,000,000 results), 'google talk' (181,000,000 results), 'google scholar' (78,500,000 results), 'google toolbar' (13,800,000 results), 'google images' (218,000,000 results), and 'google mail' (723,000,000 results). On the far right, there are links for 'Gmail', 'Images', and 'Maps'.

Suggestion	Results
google	
google earth	114,000,000 results
google maps	121,000,000 results
google.com	82,400,000 results
google map	381,000,000 results
google video	720,000,000 results
google talk	181,000,000 results
google scholar	78,500,000 results
google toolbar	13,800,000 results
google images	218,000,000 results
google mail	723,000,000 results



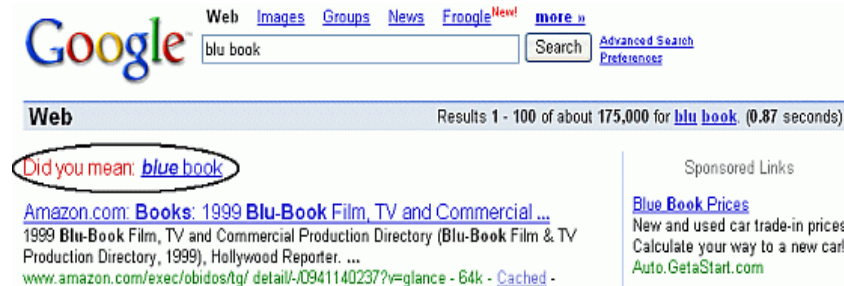
# Algorithme de correction des fautes d'orthographe

## Spelling Correction for Search Engine Queries

Bruno Martins and Mário J. Silva

Departamento de Informática  
Faculdade de Ciências da Universidade de Lisboa  
1749-016 Lisboa, Portugal

bmartins@xldb.di.fc.ul.pt, mjs@di.fc.ul.pt



## ■ Deux types:

- ❑ **Erreur typographique** *lorsque d'appuyer sur la mauvaise clé, d'appuyer sur deux clés, d'appuyer la clé de façon incorrecte d'ordre: Une simple méthode est de calculer la distance entre le mauvais mot et les mots dans un dictionnaire*

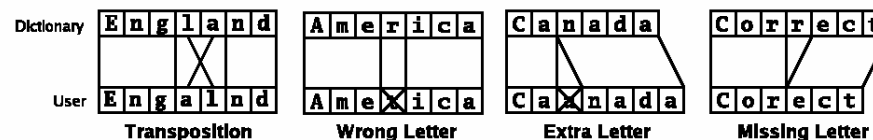
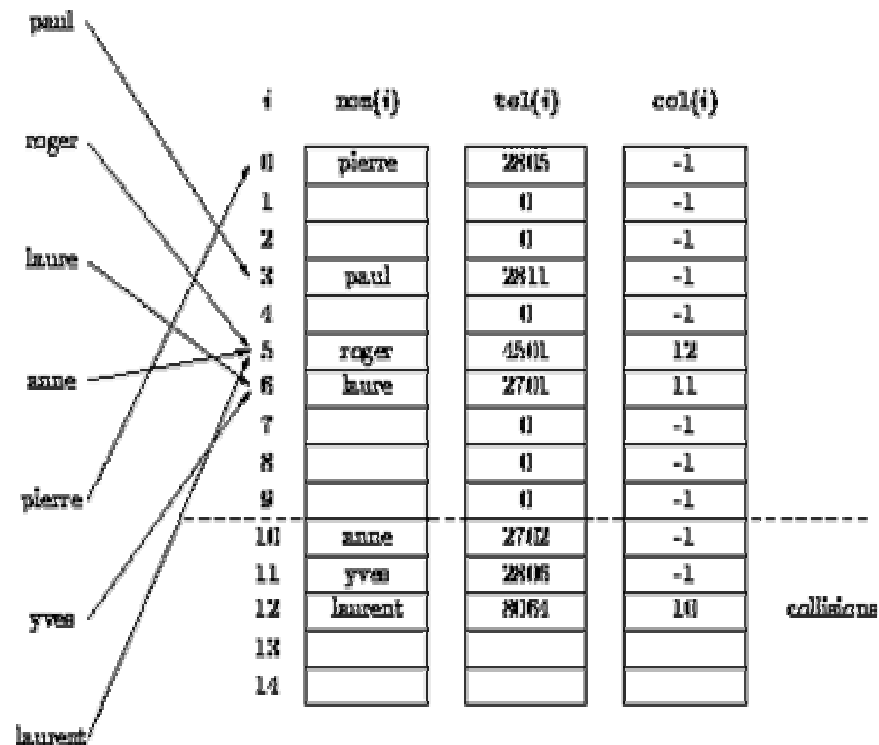


Figure 1. The four most common spelling errors.

- ❑ **Erreur phonétique** *lors de la faute d'orthographe, de grammaire (plus difficile de corriger), surtout la grammaire française. En pratique, on utilise les modèles tels que la machine Automat, les arbres sémantiques,...*



# Fonction de hachage



Utilisant un Lexicon qui contient 14 millions de mots. Il se compose 2 parties: une liste des mots et une table de hachage de pointeur

# Recherche

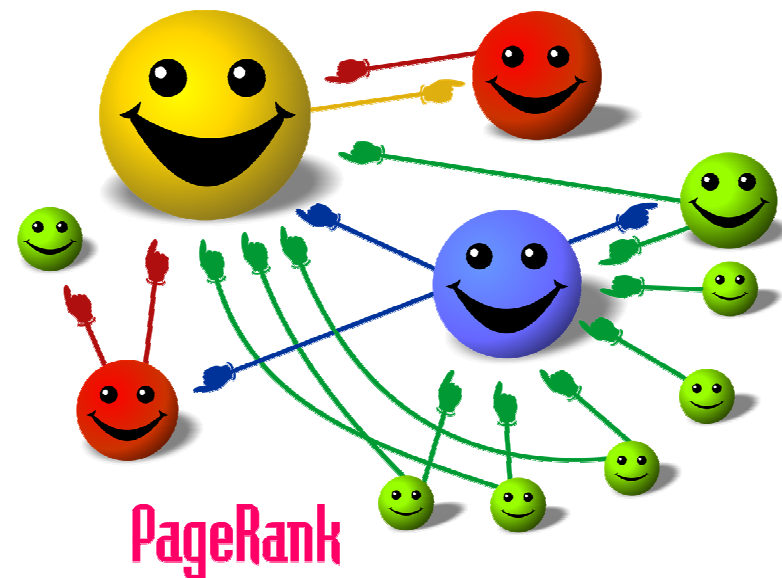
- Barrels se composent deux ensembles: un semble de listes qui contient les titres or anchors et l'autre contient les autres informations.
- Lors de la recherche, le premier ensemble sera recherché au préalable. Si il n'y a pas de correspondances dans le premier, on continue avec l'autre ensemble plus grand.
- Pour une requête qui a plus un mot, l'ordre et la position des mots trouvés sont importants.
- La liste finale des résultats est arrangée par le classement des pages.  
→ Comment peut-on les classer? Page suivant ...



# [Classement] Pagerank

L'idée: un lien émis par une page A vers une page B est assimilé à un « vote » de A pour B. Au plus une page reçoit de « votes », au plus cette page est considérée comme importante par SEs, exactement comme le principe des élections que nous connaissons tous.

La comparaison avec les élections s'arrête là car toutes les pages n'ont pas le même pouvoir de « vote ». Intuitivement, les pages sont bien citées par beaucoup d'autres sites sont bien de regarder. De plus, les citation des sites connus comme Yahoo sont meilleures. Si une page n'est pas haute de qualité, ou est cassée, c'est clair que Yahoo ne la relie plus.



# [Classement] Pagerank

- Il parle d'une page et non d'un site entier.
- $PR(T1)/C(T1)$ : Plus une page a un grand nombre de liens de sortie, moins son « vote » pèse.
- $d$ : facteur de probabilité qu'il y a un « random surfer » sur une page or un groupe de sites.

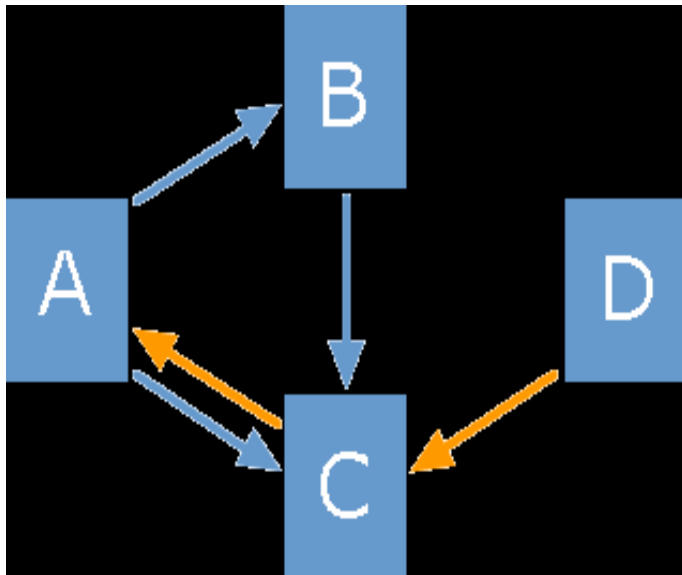


# Initialisation de l'algorithme de PageRank

- Oui, le PageRank d'une page dépend du PageRank des autres pages qui émettent un lien vers lui.
- En fait, le calcul du PageRank d'une page est peut être calculé sans connaître PR des pages concernées (égal 0 dans ce cas). Après certaines itérations, les résultats convergent vers des valeurs de plus en plus précis.
- Comme Google a effectué près de 4 milliards de pages dans sa base, il pourrait nécessiter plusieurs milliards d'itérations.



# Un exemple de Pagerank



Page A 1.49

Page B 0.78

Page C 1.58

Page D 0.15

**Somme des PageRank 4.0**

**Moyenne 1.0**

# [Classement] Analyse des textes dans un page

- Recherche par mots de clé
- Le score des textes est différent. Ça dépend leur rôle dans une page
- Anchor text
- Taille des textes, police de caractères
- => Appelée Hit lists, stocké en utilisant le codage de Huffman
- Number of Pageviews

The screenshot shows a Mozilla Firefox browser window displaying the homepage of Le Quoc Anh's project. The browser's address bar shows the URL <http://www.lequocanh.info/>. The page title is "Project: Applying Computer Vision in Traffic Surveillance System". The page content includes a summary of the project's functions and a link to a report presented at the sixth Vietnam Conference on Automation. A smaller window titled "Applying Computer Vision to Traffic Monitoring System in Vietnam" is also visible, showing a traffic monitoring interface with vehicle detection and statistics.

Annotations and scores:

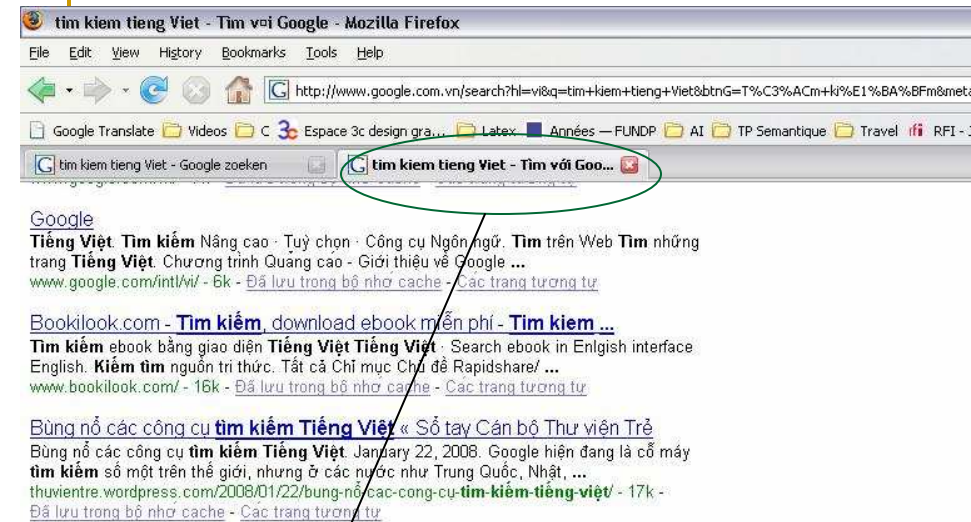
- Titre: 0,7**: Points to the page title "Project: Applying Computer Vision in Traffic Surveillance System".
- Header: 0,5**: Points to the header area of the page.
- ... Méta tags**: Points to the meta tags area of the page.
- Normal: 0.1**: Points to the main body text of the page.





# Optimisation par le géographie

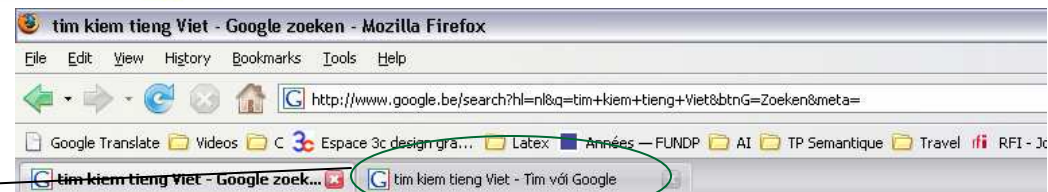
- Dépend aux IPs
- Dépend au langage de requête



www.google.com.vn

Même requête,  
les résultats  
différents

Goooooooooogle  
1 2 3 4 5 6 7 8 9 10 [Tiếp](#)



**KIEM TIEN TRUC TUYEN**, [Mua ban, rao vat \(Raovat\), tim viec, viec ...](#) - [ [Vertaal deze pagina](#) ]  
**KIEM TIEN TRUC TUYEN**, Thong tin Mua ban, rao vat, Rao Vat, **tim** viec, viec lam, ... mà đăng  
 vì trên đó có rất nhiều trang web nổi **tiếng** của **Việt** Nam. ...  
[www.azraovat.com/?mode=ads&siteid=631319](#) - 39k - [In cache](#) - [Gelijkwaardige pagina's](#)

**NGUOI VIET KIEM TIEN**, [Mua ban, rao vat \(Raovat\), tim viec, viec ...](#) - [ [Vertaal deze pagina](#) ]  
**NGUOI VIET KIEM TIEN**, Thong tin Mua ban, rao vat, Rao Vat, **tim** viec, viec lam, ... Phần  
 đăng kí bằng **tiếng viet** nên rất dễ dàng, các bạn chỉ cần điền đầy đủ ...  
[www.azraovat.com/?mode=ads&siteid=636406](#) - 30k - [In cache](#) - [Gelijkwaardige pagina's](#)  
[Meer resultaten van www.azraovat.com »](#)

**buom.vn084.com - Tim kiem** - Thong Tin - Quang Cao - Giai Tri ... - [ [Vertaal deze pagina](#) ]  
 Quý vị đang tìm **kiếm** thông tin web, xin nhập từ **kiếm** bên trên. ... Tra bươm trong từ điển  
 mở **tiếng Việt** Wiktionary ... Ngoài nghĩa đen chỉ loài côn trùng, ...  
[www.vn084.com/tim/web.asp?q=buom](#) - 17k - [In cache](#) - [Gelijkwaardige pagina's](#)

www.google.be

Goooooooooogle  
1 2 3 4 5 6 7 8 9 10 [Volgende](#)



## [Recherche]: L'interface des résultats

- Les résultats sont classés de façon décroissante de « points »
- Les textes des résultats sont résumés qui contiennent les mots de clés de la requête (comment peut-on résumer un texte automatiquement?)

[Des changements dans l'\*\*algorithme\*\* de Google ? \(22 février 2007 ... 22 fév 2007 ... J'ai vu des changements remarquables sur certains de mes sites, ... référencement ou l'effet de la modification de l'\*\*algorithme\*\* de Google. ... \[www.webrankinfo.com/actualites/200702-changements-google-fevrier-2007.htm\]\(http://www.webrankinfo.com/actualites/200702-changements-google-fevrier-2007.htm\) - 42k - En cache - Pages similaires - À noter](#)

### [Doctorant SOPENA Julien](#)

Dans une première partie j'ai proposé un nouvel **algorithme** d'exclusion mutuelle tolérant les fautes qui est une extension de l'**algorithme** de Naimi-Tréhel ... [www.lip6.fr/fr/actualite/personnes-thesard-fiche.php?RECORD\\_KEY\(thesard\)=id&id\(thesard\)=552](http://www.lip6.fr/fr/actualite/personnes-thesard-fiche.php?RECORD_KEY(thesard)=id&id(thesard)=552) - 9k - En cache - Pages similaires - À noter

### [L'\*\*algorithme\*\* du PageRank expliqué](#)

J'ai donc créé mes premiers sites dynamiques, en partant de la feuille blanche. ... L'**algorithme** du PageRank est un des sujets qui a suscité le plus de ... [www.webmaster-hub.com/publication/L-algorithme-du-PageRank-explique.html](http://www.webmaster-hub.com/publication/L-algorithme-du-PageRank-explique.html) - 43k - En cache - Pages similaires - À noter

### • What is summarization?

– **A summary is a concise restatement of the topic and main ideas of its source.**

- **Concise** – giving a lot of information clearly and in a few words. (Oxford American Dictionary)
- **Restatement** – in your own words.
- **Topic** – what is the source about?
- **Main ideas** – important facts or arguments about the topic.

Le résumé automatique se propose de faire une extraction de l'information jugée importante d'un texte d'entrée pour construire, à partir de cette information, un nouveau texte de sortie, condensé. Ce nouveau texte permet d'éviter la lecture en entier du document source.

Jean-Luc Minel, ingénieur de recherche du CNRS au laboratoire MoDyCo (université Paris X-Nanterre)

# [Recherche – Interface des résultats]: Résumé automatique

## Pourquoi?

- ❑ On n'a pas le temps de tout lire
- ❑ Limitation de l'interface d'affiche des résultats
- ❑ Les résumés multilingues => recherches multilingues

## Il y a deux approches:

- ❑ Psychologie (tam ly hoc): simuler l'habitude humain
- ❑ Ingénierie linguistique: basé sur la structure d'un texte, les phrase, les lexiques et analyseurs morphosyntaxiques



## Autres:

- ❑ Utile aussi pour les journaux ou les textes scientifiques.

Mais il y a plusieurs questions non résolues.

## [Recherche – Interface des résultats]: Résumé automatique

- Le source est peut-être un seul document mais aussi des documents (dans ce cas, on dit *multi-document summarization*).
- « According to Hovy and Lin (1997) there are two ways to view text summarization either as text extraction or as text abstraction. Text extraction means to extract pieces of an original text on a statistical basis or with heuristic methods and put together it to a new shorter text with the same information content. Text abstraction is to parse the original text in a linguistic way, interpret the text and find new concepts to describe the text and then generate a new shorter text with the same information content. »
- Il a un rôle important dans un moteur de recherche, donc je vais en parler plus longtemps que les autres



# Comment peut-on résumer un texte?

## ■ Identification du sujet

- Trouver les informations les plus importantes – les thèmes
- Trouver les idées principales qui soutiennent les thèmes et montrer comment ils se lisent

## ■ Interprétation du sujet

- Combiner certaines idées à une seule phrase
- Substituer une liste des articles, événements à une forme générale
- Enlever les informations insignifiantes et redondantes



# Algorithme de résumé automatique: Approches

- *Je sais ce que je veux! — ne pas confondez-moi les bêtises*



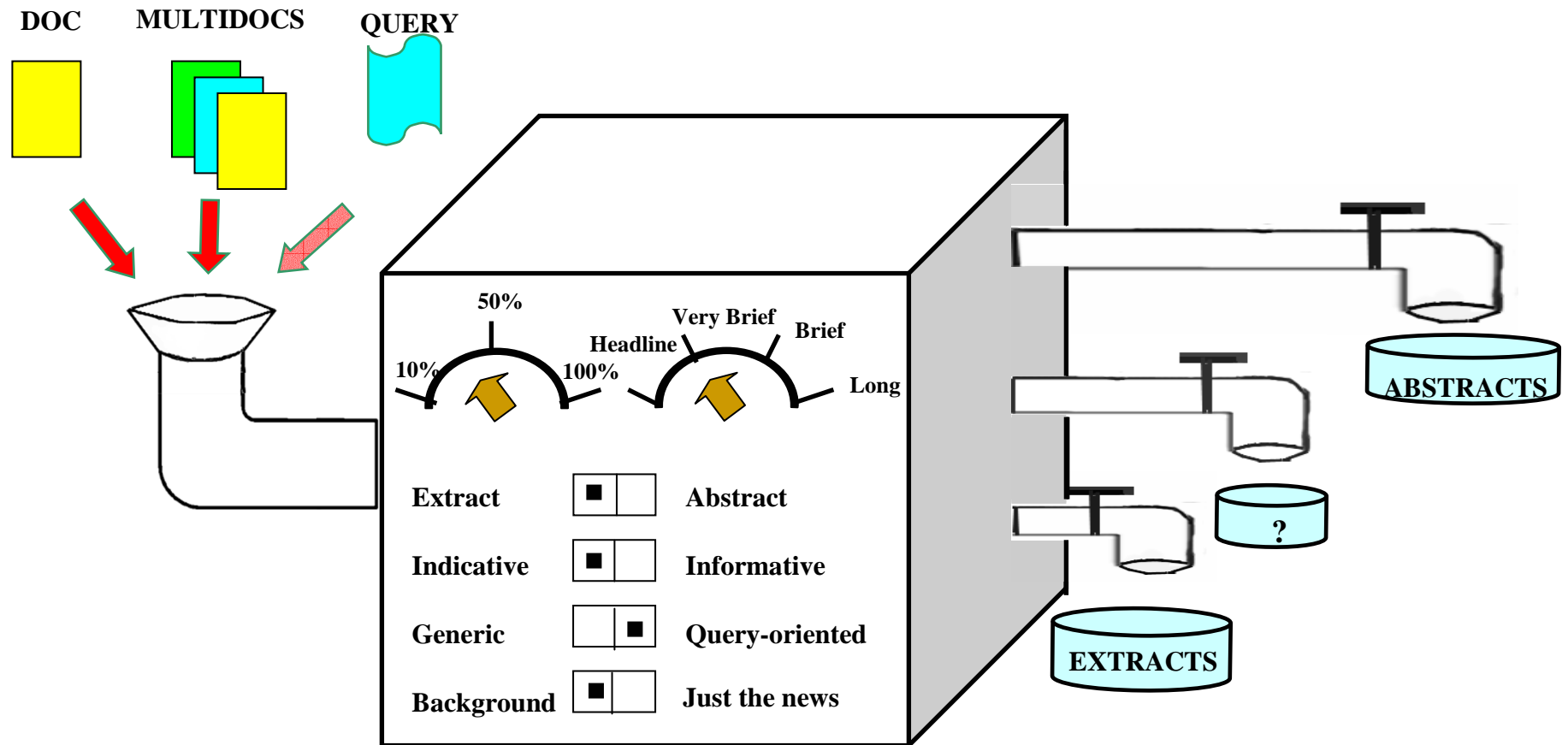
- Utilisateur souhaite seulement les types certains d'informations.
- Système a besoin de résumer le texte de façon de concentrer les intérêts.
  - Essayer de « comprendre » un texte et le transformer à une notation “plus profond”
  - Appliqué les analyses des textes à tout niveau des mots, phrases,..

- *Je suis curieux: Qu'est-ce que c'est le suivant?*

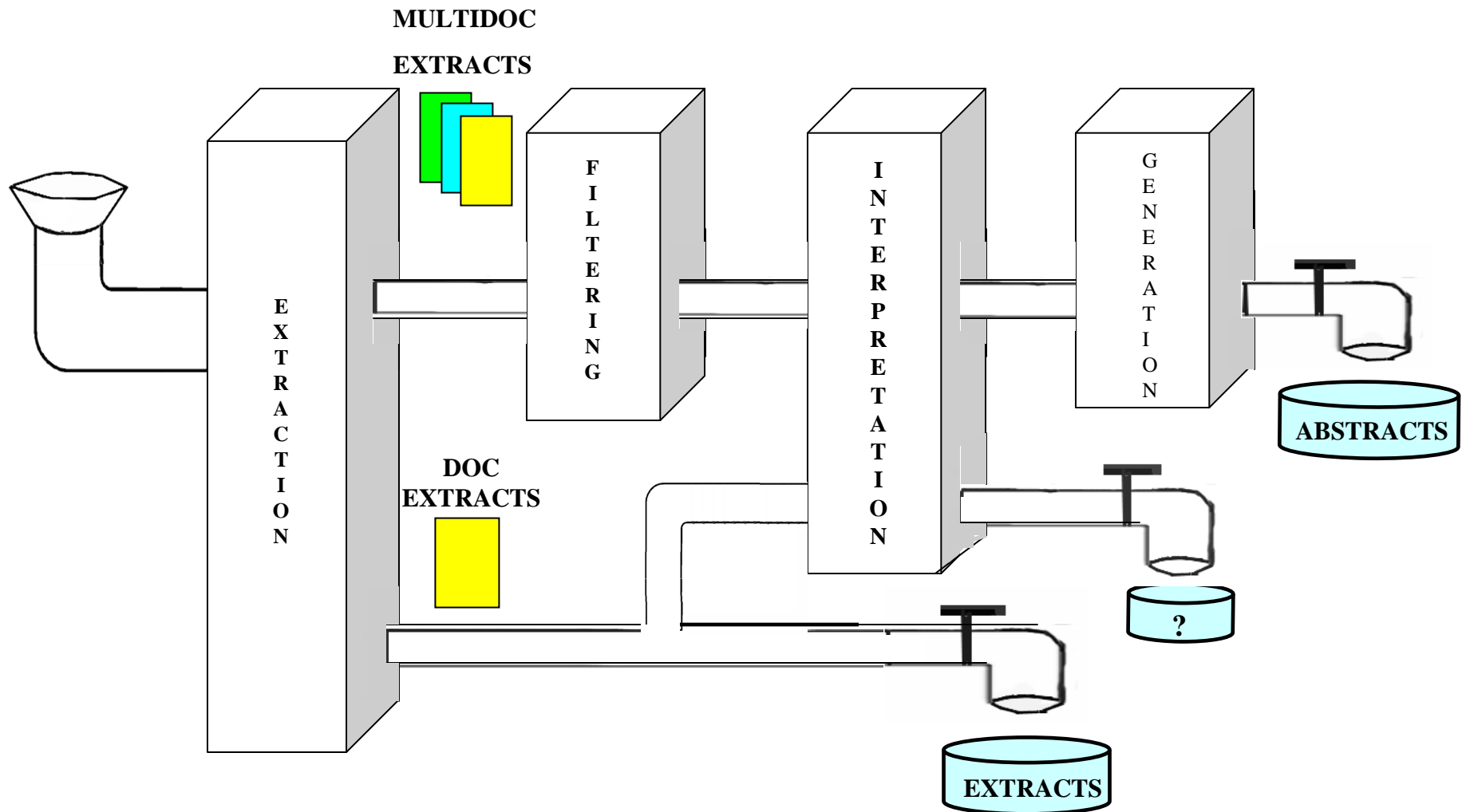


- Utilisateur souhaite tout ce qui est important.
- Système a besoin de résumer le texte de façon de concentrer les informations importants générales .
  - Opérer les textes au niveau des mots tels que la fréquence des mots, les collocations,...

# A Summarization Machine



# The Modules of the Summarization Machine





# Indentification du sujet

- **Méthode générale:** marquer un point pour chaque phrase, combiner les points et après choisir les phrases qui ont les points les plus hauts
- **Techniques:**
  - Sa position dans le texte: méthode d'avance (*plus important au début du texte*); position politique optimal; méthode des titres ou en tête (*c'est bien ajouter les mots dans les titres et les têtes au résumé*)
  - Signal des phrases dans le texte (contiennent les 'bons signaux' tels que '*En conclusion*', '*Significativement*',...)
  - Fréquence des mots
  - Cohérence: liens entre les mots (relation sémantique); co-occurrence des mots; co-référence; chaîne lexicale (synonyme, polysémique, domaine...), requête -> titres -> document
  - Structure du texte (construire un arbre des textes,...)

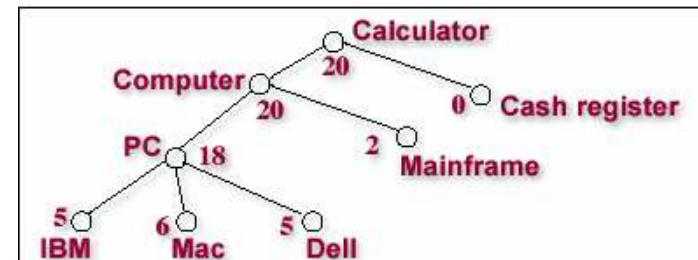
$$Score = a_1C + a_2K + a_3T + a_4P$$



# Interprétation du sujet

- Besoin de construire un grand nombre de dictionnaires et un bon texte sémantique
  - Concept général: *Sue ate apples, pears, and bananas => sue ate fruit*
  - Remplacement : *Both wheels, the pedals, saddle, chain,... => the bike*
  - Identification de script: *He sat down, read the menu, ordered, ate, paid, and left => He ate at the restaurant*
  - Métonymie (phep hoan du): *A spokesperson for the US Government announced that ... => Washington announced that ...*
- Détection des relations en utilisant les règles tels que *if ... then, arbres hiérarchiques,...*

*signature = {head (c1,f1) (c2,f2) ...}*  
*restaurant  $\leftarrow$  waiter + menu + food + eat...*



# Un simple algorithme basé sur l'extraction

## Automated Text Summarization in SUMMARIST

Eduard Hovy and ChinYew Lin  
Information Sciences Institute  
of the University of Southern California  
4676 Admiralty Way  
Marina del Rey, CA 90292-6695, U.S.A.  
tel: +1-310-822-1511  
fax: +1-310-823-6714  
email: {hovy,cyl}@isi.edu

- Lin (1999) describes a set of summarization methods and algorithms based on extraction:
- **Baseline:** Sentence order in text gives the importance of the sentences. First sentence highest ranking last sentence lowest ranking.
- **Title:** Words in title and in following sentences gives high score.
- **Term frequency (tf):** Open class terms which are frequent in the text are more important than the less frequent. Open class terms are words that change over time.
- **Position score:** The assumption is that certain genres put important sentences in fixed positions. For example. Newspaper articles has most important terms in the 4 first paragraphs.
- **Query signature:** The query of the user affect the summary in the way that the extract will contain these words.
- **Sentence length:** The sentence length implies which sentence is the most important.
- **Average lexical connectivity:** Number terms shared with other sentences. The assumption is that a sentence that share more terms with other sentences is more important.
- **Numerical data:** Sentences containing numerical data obtain boolean value 1 (is scored higher ) than the ones without numerical values.
- **Propername:** Dito for propernames in sentences.
- **Pronoun and Adjective:** Dito for pronouns and adjectives in sentences. Pronouns reflecting coreference connectivity.
- **Weekdays and Months:** Dito for Weekdays and Months:
- **Quotation:** Sentences containing quotations might be important for certain questions from user.
- **First sentence:** First sentence of each paragraphs are the most important sentences.
- **Decision tree combination function:** All the above parameters were put into decision tree and trained on set of texts and manual summarized texts.
- **Simple combination function:** All the above parameter were normalized and put in a combination function with no special weighting.



## Difficultés pour le résumé des sites web

- Un page web se compose les données différentes tels que les textes, les images, les sons, les vidéos, etc. Ce média est très important mais très difficile de résumer automatiquement.
- Les pages web contiennent peut-être les informations non directes. Par exemple, les sites [www.yahoo.com](http://www.yahoo.com) or [www.msn.com](http://www.msn.com) contiennent principalement les descriptions de façon de référencer aux autres sites web.
- Une partie des informations dans un page web est peut-être ignorée.
- Un document logique est peut-être présenté sur plusieurs pages.  
[source: Mikhail Kondratyev – Saint-Petersburg State University]



# Algorithmes du résumé pour les sites web

1. Intégrer les sites relatifs (les statiques et les dynamiques) à un seul texte
2. Supprimer les parties non informations.
3. Résumer le texte comme l'habitude mais à l'aide aussi des caractéristiques d'un document HTML

## Automated Query-biased and Structure-preserving Text Summarization on Web Documents

F. Canan Pembe  
*Dept. of Computer Engineering*  
*Boğaziçi University, İstanbul, Turkey*  
*canan.pembe@boun.edu.tr*

Tunga Güngör  
*Dept. of Computer Engineering,*  
*Boğaziçi University, İstanbul, Turkey*  
*gungort@boun.edu.tr*

```
<document>
<heading> Automated Query-biased and Structure-
preserving Text Summarization on Web Documents
</heading>
...
<section level = 1>
<heading>Proposed System</heading>
...
<section level = 2>
<heading>Structural Processing</heading>
...
<sentence>The structure of a document
may be considered as a hierarchy, where each
document has sections; each section has
subsections, and so on.</sentence>
...
</section>
...
</section>
</document>
```

### Algorithm:

- 1: Rank all the sentences according to their score.
- 2: Add the main title of the document to the summary.
- 3: Add the first level-1 heading to the summary.
- 4: **While** (summary size limit not exceeded)
- 5:   Add the next highest scored sentence.
- 6:   Add the structural context of the sentence:  
     (if any and not already included in the summary)
- 7:   Add the highest level heading above the  
     extracted text (call this heading *h*).
- 8:   Add the heading before *h* in the same level.
- 9:   Add the heading after *h* in the same level.
- 10:   Repeat steps 7, 8 and 9 for the next highest level  
       headings.
- 11: **End while**

### Automated Query-biased and Structure-preserving Text Summarization on Web Documents

#### 1. Abstract

...  
 Different from the previous work, both the structural information and the content to be displayed in the summary are selected in a query-biased way.

#### 2. Related Work

#### 3. Proposed System

##### 3.1. Structural Processing

...  
 The structure of a document may be considered as a hierarchy, where each document has sections; each section has subsections, and so on.

##### 3.2 Linguistic Processing

...



# Implémentation des moteurs de recherche

- Systèmes distribués
- Serveurs de secours
- Plusieurs mémoires temporaires possibles (en fait, le lexique est stocké dans un RAM – la mémoire vive d'un ordinateur)
- les caches pour stocker les sites entiers récupérés, les caches pour stocker les résultats des requêtes



# Implémentation des moteurs de recherche

- En utilisant les grands fichiers virtuels pour éviter le temps de recherche sur un disque dur (les fichiers sont découpés en blocs de 64MB nommés *chunks*. *Chaque bloc possède un identifiant unique de 64 bits, ainsi qu'un numéro de version, afin de détecter les inconsistances de données. Un fichier est répliqué en plusieurs exemplaires dans le système, dans les nœuds différents, ainsi, si une copie disparaît (à cause d'une panne du serveur le stocke), le fichier reste généralement accessible – source de wiki* )
- Compression en utilisant zlib (voir RFC1950) avec le taux 3:1

Repository: 53.5 GB = 147.8 GB uncompressed

sync	length	compressed packet
sync	length	compressed packet

...

**Packet** (stored compressed in repository)

docid	ecode	urlen	pagelen	url	page
-------	-------	-------	---------	-----	------

Figure 2. Repository Data Structure

# Futur

Suivant le Navigators « The best navigation service should make easy to find almost anything on the Web (once all the data is entered). »

- Donc, deux buts: on peut trouver n'importe quel information existant sur Internet.
- Les résultats sont plus proches le souhait des requêtes.





# Difficultés

- Il n'y a pas beaucoup de documents officiels qui parlent des techniques, des algorithmes du moteur de recherche parce que presque tous les moteurs de recherche sont commerciales et ils ne veulent pas publier leurs secrets.
- Les techniques de clés sont cachées.



# Conclusion

- On souhaite que dans le futur proche, il y a un SE idéal qui pourra trouver tous les informations dans tous les types de formats (vidéos, images, documents – imprimés, manuscrits, audios, ...)
- Un moteur de recherche est un riche environnement pour faire les recherches, les Als en particulier.



# Vous avez des questions ?

