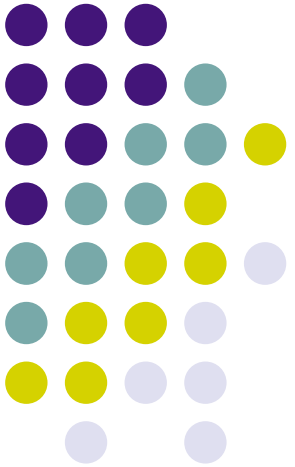
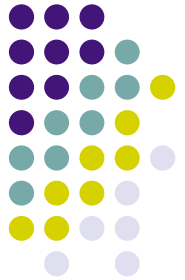
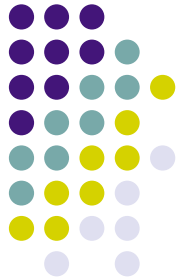
Recherche d’information structurée



Plan



- Quelques rappels: XML
- Problématique
- Indexation
  - Termes : propagation des termes ou unités disjointes
  - Structure
- Interrogation
- Appariement
  - Approches basées sur la propagation des termes (pondérés ou non)
  - Approches basées sur la propagation des scores
- Evaluation
- Visualisation

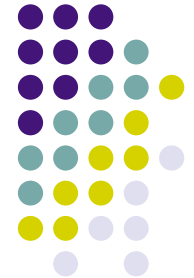


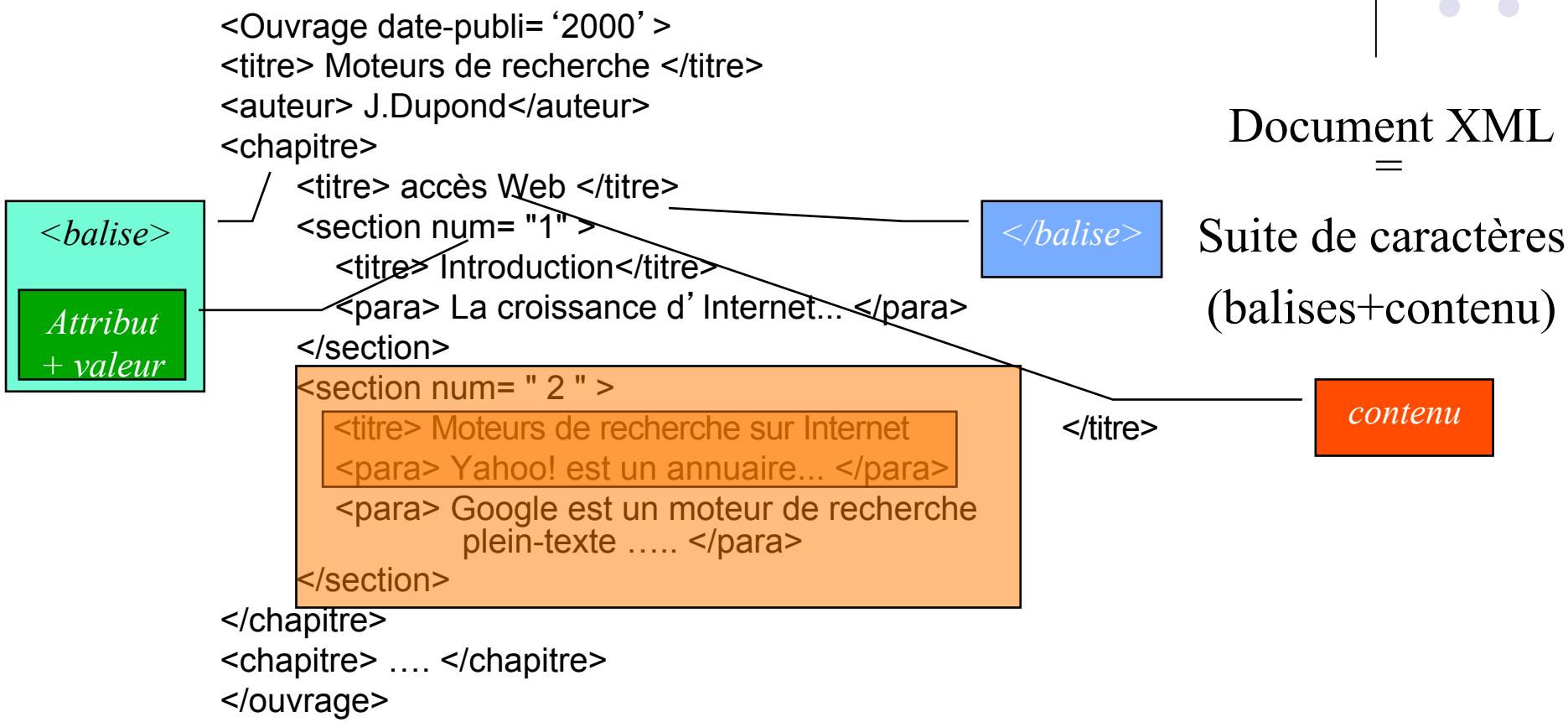
Documents XML

- XML (eXtensible Markup Language) est un méta-langage, utilisé pour représenter du texte et de la structure

- Applications XML : échange de données, « digital libraries », gestion de contenu, documentation complexe, etc.

Document XML: rappels





Document XML

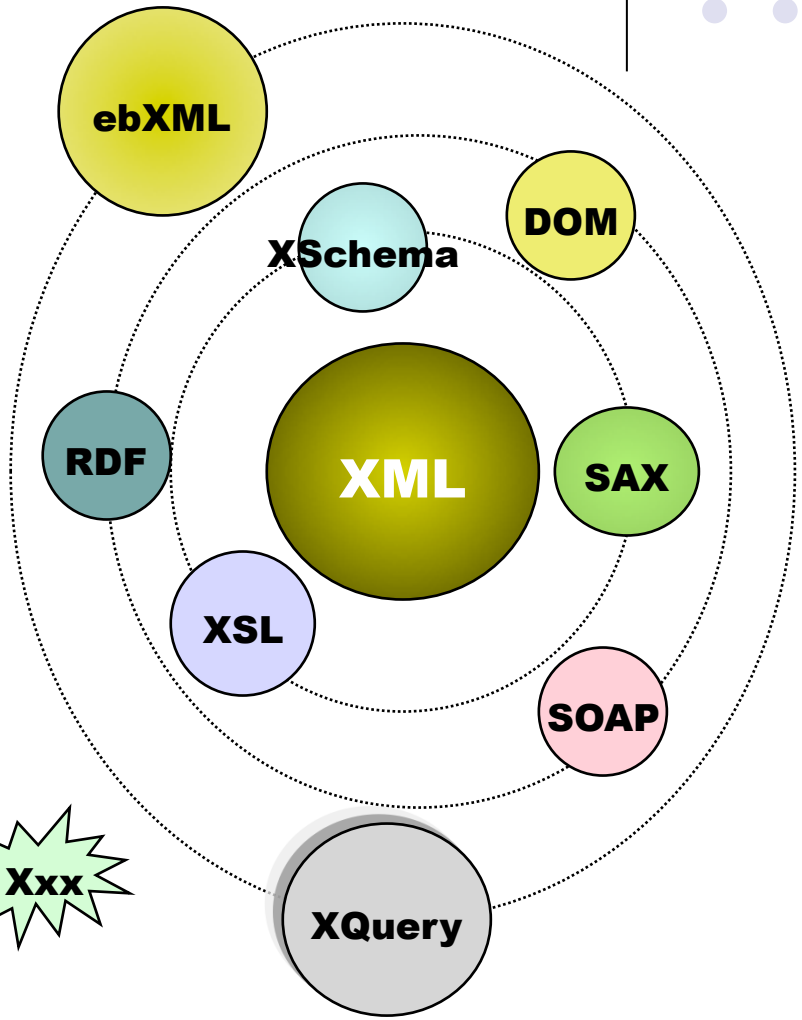
=

Suite de caractères (balises+contenu)

Un élément = <balise> contenu </balise>

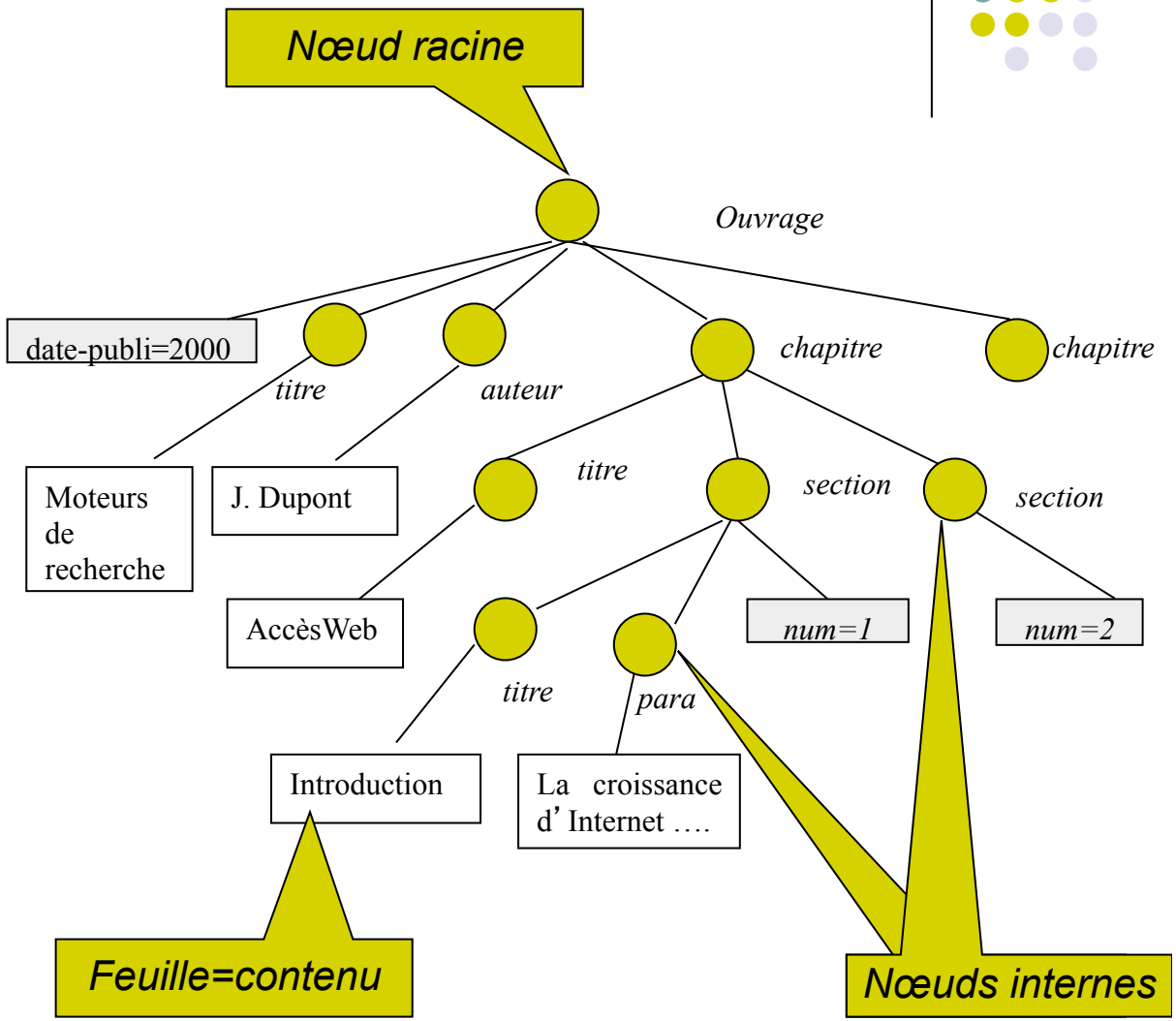
La galaxie de standards: rappels

- XSchema
  - Schémas de documents
- XSL
  - Feuilles de styles
- SAX
  - API de programmation événementielle
- DOM
  - API de programmation objet
- SOAP
  - Protocole Web Services
- RDF
  - Description de ressources Web
- Xxx
  - Standards par métiers …

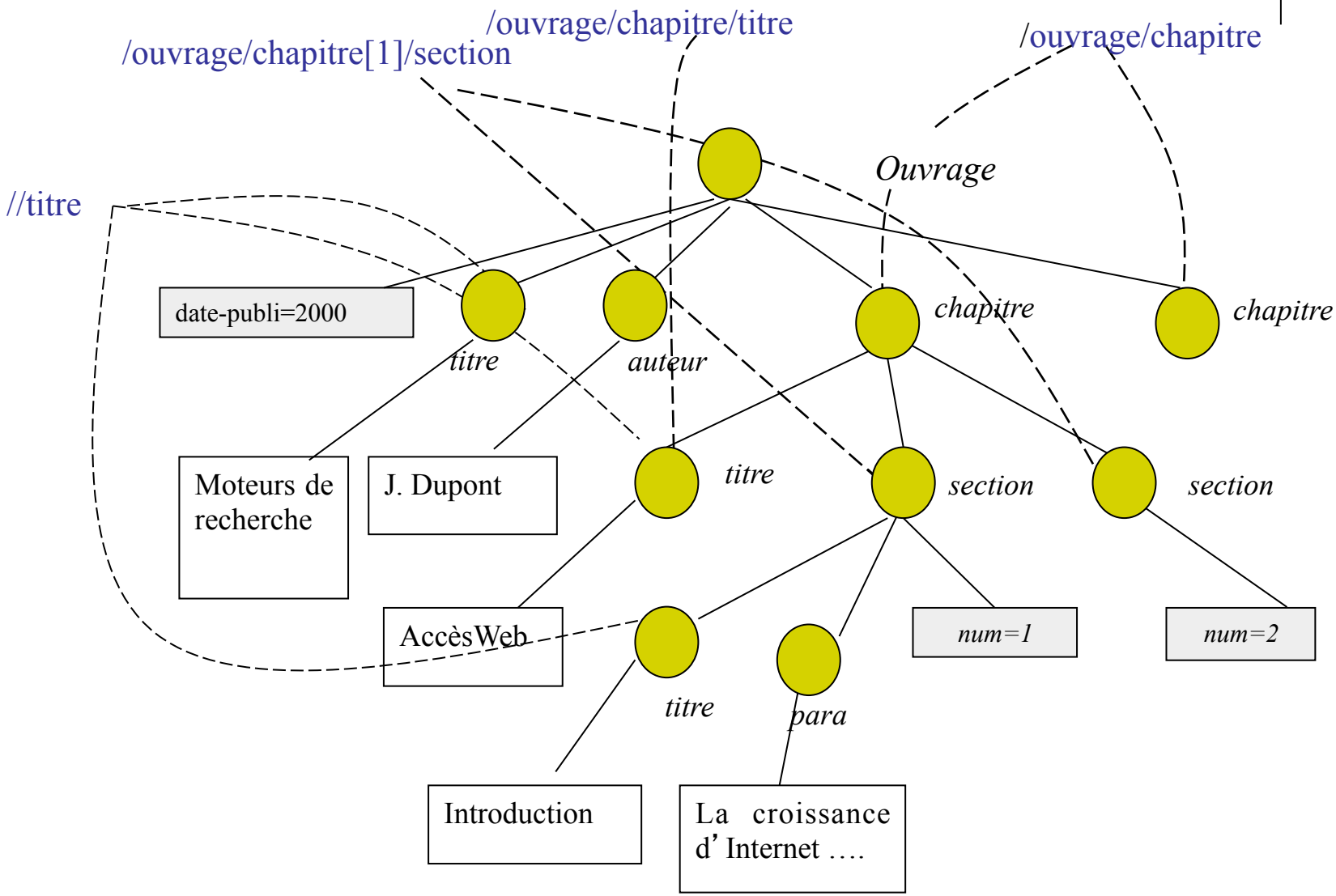


Arbre DOM: rappel

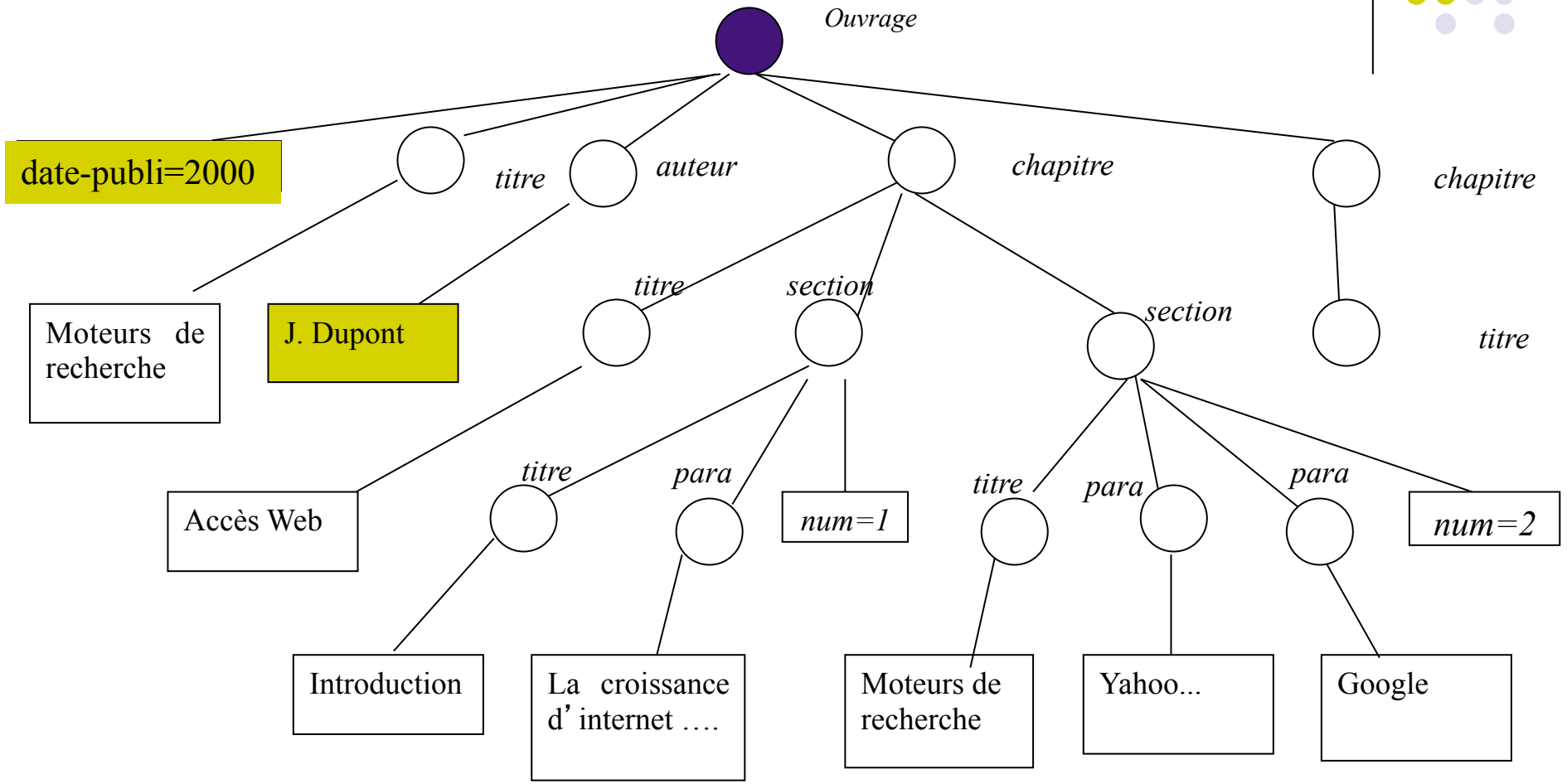
```
….
!-- element racine -->
<ouvrage date-publi= '2000' >
<!-- enfants -->
<titre> Moteurs de recherche </titre>
<auteur> J.Dupond</auteur>
<chapitre>
      <titre> accès Web </titre>
      <section num= "1" >
         <titre> Introduction </titre>
         <para> La croissance
      d' Internet... </para>
      </section>
<section num= "2" >…
</section>
</chapitre>

<chapitre> …. </chapitre>
</ouvrage>
```



XPath - Exemples

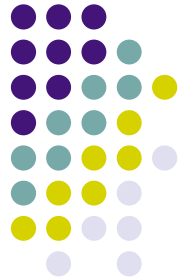


XPath - Exemples



Filtrer par le Contenu :

/ouvrage[@date-publi="2000" AND auteur="J. Dupont"]



XPath: rappel

- Langage d'expressions de chemins permettant de sélectionner des parties d'un document XML

- **Chemin = étape de localisation1/…/étape de localisationN**
- **Etape de localisation = Axe:: testDeNoeud [predicat(s)]**

  - Axe: sens de sélection des éléments

  - TestDeNoeud: type ou nom des éléments sélectionnés selon l'axe directionnel

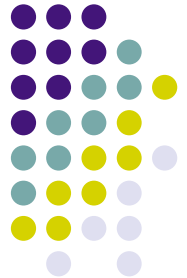
  - Predicat: 0 ou plusieurs prédicats

//	descendants
/	fils
.	Nœud courant
..	père
@	attribut

*	tous les noeuds

- Ce qui est renvoyé: **un ensemble de noeuds**

Deux grands types de documents XML



- Documents centrés données (« BD »)

```
<CLASS name="DCS317" num_of_std="100">
<LECTURER lecid="111">Thomas</LECTURER>
<STUDENT marks="70" origin="Oversea">
        <NAME>Mounia</NAME>
</STUDENT>
<STUDENT marks="30" origin="EU">
        <NAME>Tony</NAME>
</STUDENT>
</CLASS>"
```

  - Du domaine de la BD

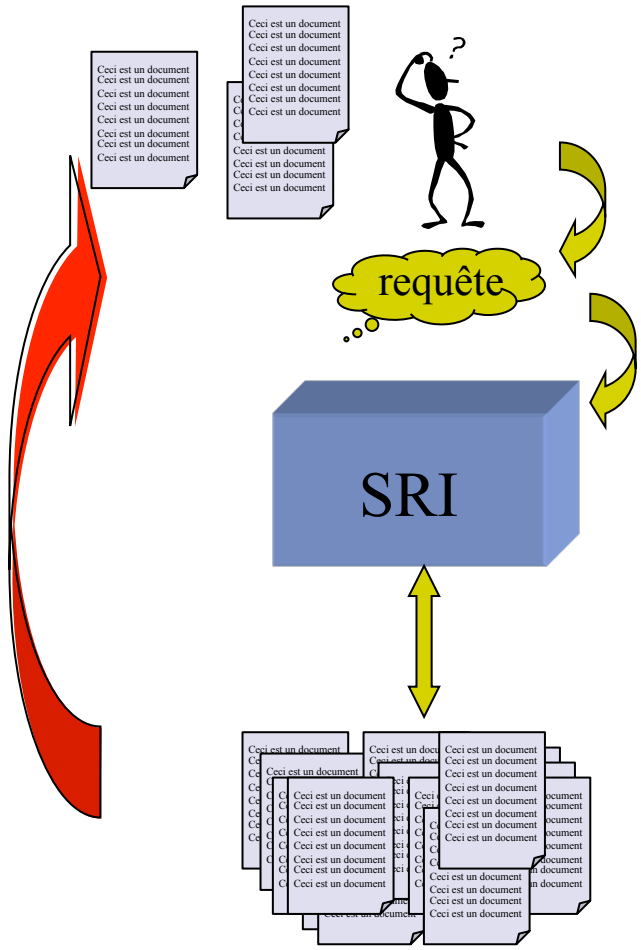
Deux grands types de documents XML

- Documents centrés documents

```
<CLASS name="DCS317" num_of_std="100">
 <LECTURER lecid="111">Mounia</LECTURER>
 <STUDENT studid="007" >
        <NAME>James Bond</NAME> is the best student in the class.
        He scored <INTERM>95</INTERM> points out of <MAX>100</MAX>.
        His presentation of <ARTICLE>Using Materialized Views in
        Data Warehouse</ARTICLE> was brilliant.
</STUDENT>
 <STUDENT stuid="131">
        <NAME>Donald Duck</NAME> is not a very good student. He
          scored <INTERM>20</INTERM> points...
</STUDENT>
</CLASS>
```

  - Du domaine de la recherche d'information

Contexte



- Systèmes de Recherche d'Information classiques
  - Requêtes = mots clés
  - Granule documentaire = document entier
- Avec les documents structurés et semi-structurés
  - Séparation du contenu de la structure et de la présentation

Contexte

- Focalisation sur le besoin de l'utilisateur
  - Granule documentaire = Partie de document

Hypertexte

  - Comment définir la pertinence?

Chercher = interroger + naviguer ?

```
<?xml version="1.0" ?>
<!--Exemple de fichier XML d'ecrivant un article scientifique -->
<article annee="2003">
<en-tête>
<titre>Recherche d'information sur le web : la grande révolution</
titre>
<auteur> André Dupont </auteur>
</en-tête>
<corps>
    <section>
        <sous-titre> Histoire de l'hypertexte : des pères fondateurs au
World Wide Web</sous-titre>
        <par> Afin de maîtriser les enjeux de ces systèmes, il convient,
même si c'est une tâche ardue,
        d'essayer de les définir… </par>
    </section>
    <section></section>
   <section>
        <sous-titre> L'analyse des liens  </sous-titre>
        <par> … </par>
    </section>
</corps>
</article>
```

Recherche « focalisée »

Focused retrieval: Scientific Collection

- Query

  model checking
  aviation systems

- Answer

  one section in a
  workshop report

Recherche « focalisée »

Focused Retrieval: Encyclopedia

- Information need

  volcanic eruption prediction

- Answer

  relatively small portion of the volcano topic

Recherche « focalisée »

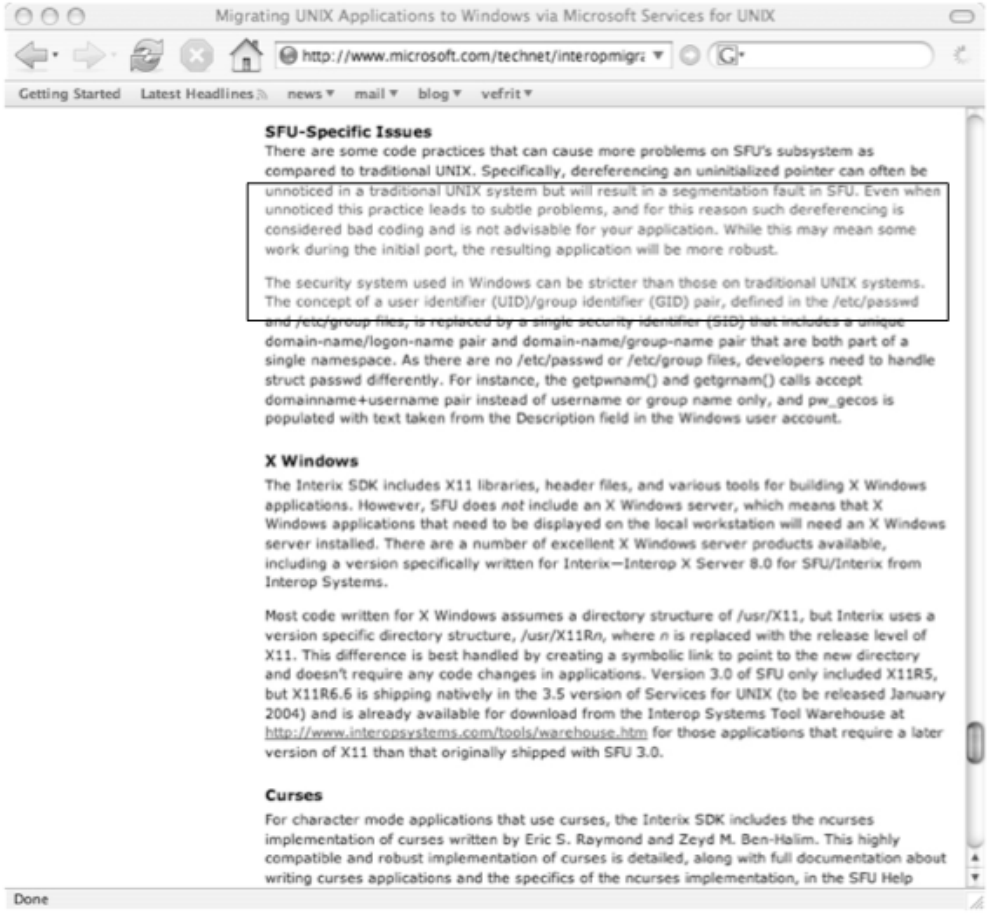
Focused retrieval: Technical Manual

- Query

  segmentation fault
  windows services
  for unix

- Answer

  only a single paragraph in a long manual



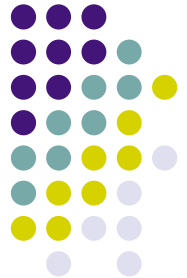
Recherche « focalisée »

Focused retrieval: Right level of granularity

Query: wordnet information retrieval

OntoSeek: Content-Based Access to the Web
Nicola Guarino, Claudio Masolo, Guido Vetere

article[1]
fm[1]
abs[1]
p[1]
... and precision of content-based **retrieval**. Our OntoSeek system adopts ... large ontology based on **WordNet** for content matching.
bdy[1]
sec[3] (THE ROLE OF LINGUISTIC ONTOLOGIES)
ss1[1] (Some advantages)
p[4]
The **retrieval** quality improves considerably if ... linguistic ontology such as **WordNet**. For example, let's add **WordNet** to a simple matching ...
p[7]
... linguistic ontologies such as **WordNet** and structured representation formalisms can help an **information-retrieval** system to
sec[4] (ONTOSEEK)
... of a project on **retrieval** and reuse of object-oriented ... system designed for content-based **information retrieval** from online yellow pages ... mostly resulting from merging **WordNet**'s thesaurus into the Penman ... broad ontology endowed with **WordNet**'s powerful lexical interface, which ...



Problématique

- Deux dimensions de pertinence
  - **Exhaustivité**
    - toutes les informations requises dans la requête sont présentes
  - **Spécificité**
    - tout le contenu de l'unité d'information concerne la requête
- But:
  - Proposer des modèles permettant de traiter 2 types de requêtes
    - contenu
    - contenu et structure
  - Gérer les 2 dimensions de pertinence

Problématique

Collection de documents XML

Indexation

Représentation

Requêtes
- Contenu
- Contenu et structure

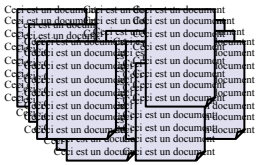
~~Requête (mots-clés)~~

Indexation

Représentation

contenu + structure

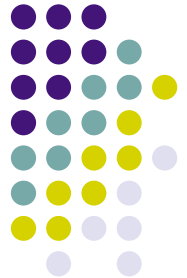
Appariement



~~Documents sélectionnés~~

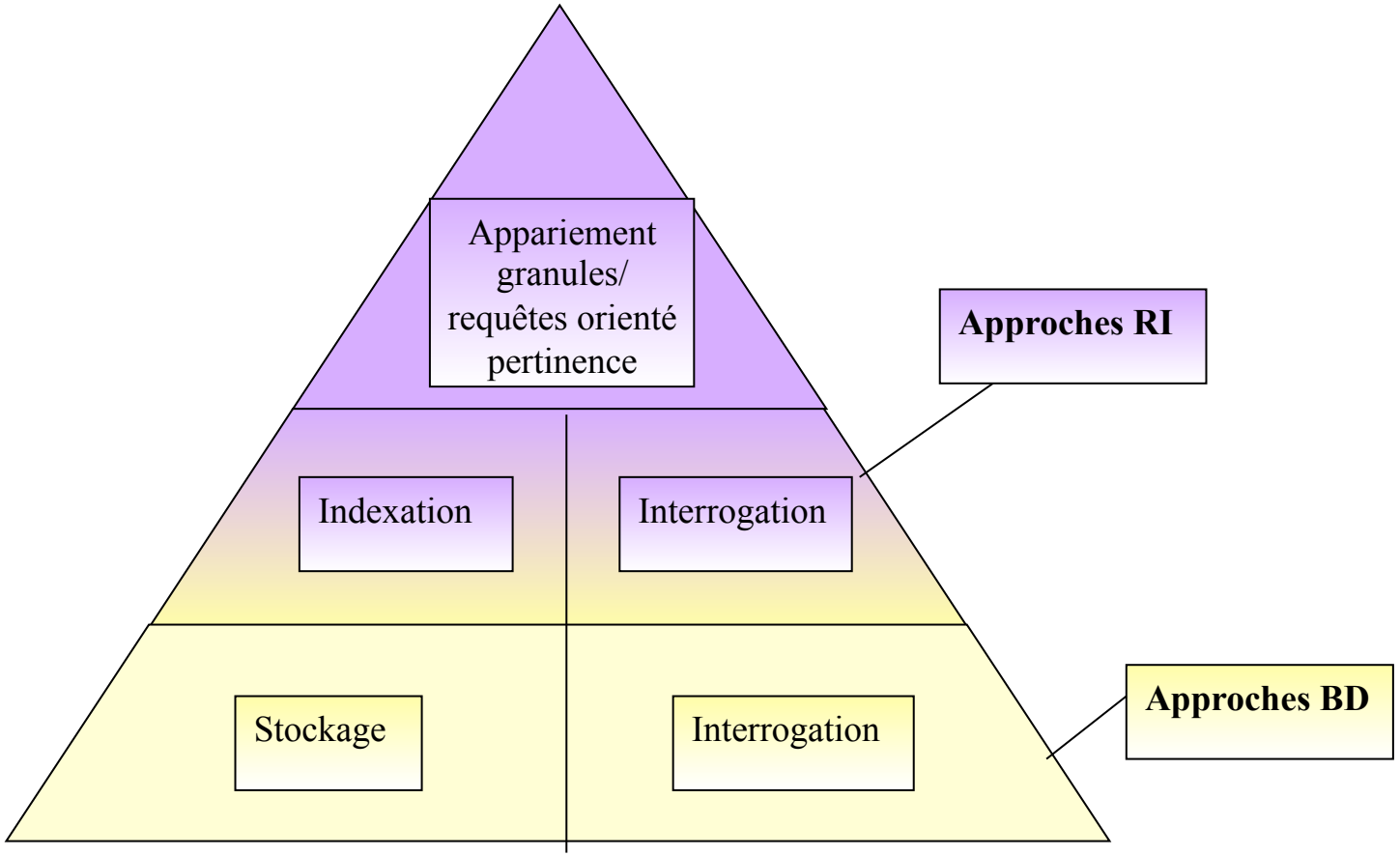
Parties de documents sélectionnées

Deux approches … BD et RI



- Accès aux documents XML est abordé selon deux approches principales :
  - approche centrée données : communauté des bases de données (BD)
  - approche centrée documents : communauté de la recherche d'information (RI)

BD et RI


Appariement granules/ requêtes orienté pertinence
Approches RI
Indexation
Interrogation
Stockage
Interrogation
Approches BD

Les approches orientées BD

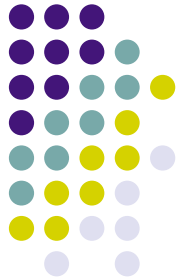
- Premières approches proposées pour l'accès aux données semi structurées
- Transformation des documents XML en tables et vice-versa
- Développement de nombreux langages d'interrogation
  - UnQL, Lorel, XQL, Quilt et XQuery (Recommandation du W3C)
  - (Balises, Données)=(attributs, valeurs)
- ☹ : traitement du contenu textuel
  - Traitent des expressions **attribut = valeur**
  - ⇒ **correspondance exacte**

Les approches orientées RI

- Surcouche pour l'évaluation de la pertinence
  - Adaptation des modèles traditionnels de RI
    - vectoriel
    - probabiliste
    - ...

  ⇒ **correspondance partielle**

- ☹ : traitement de la structure



Indexation

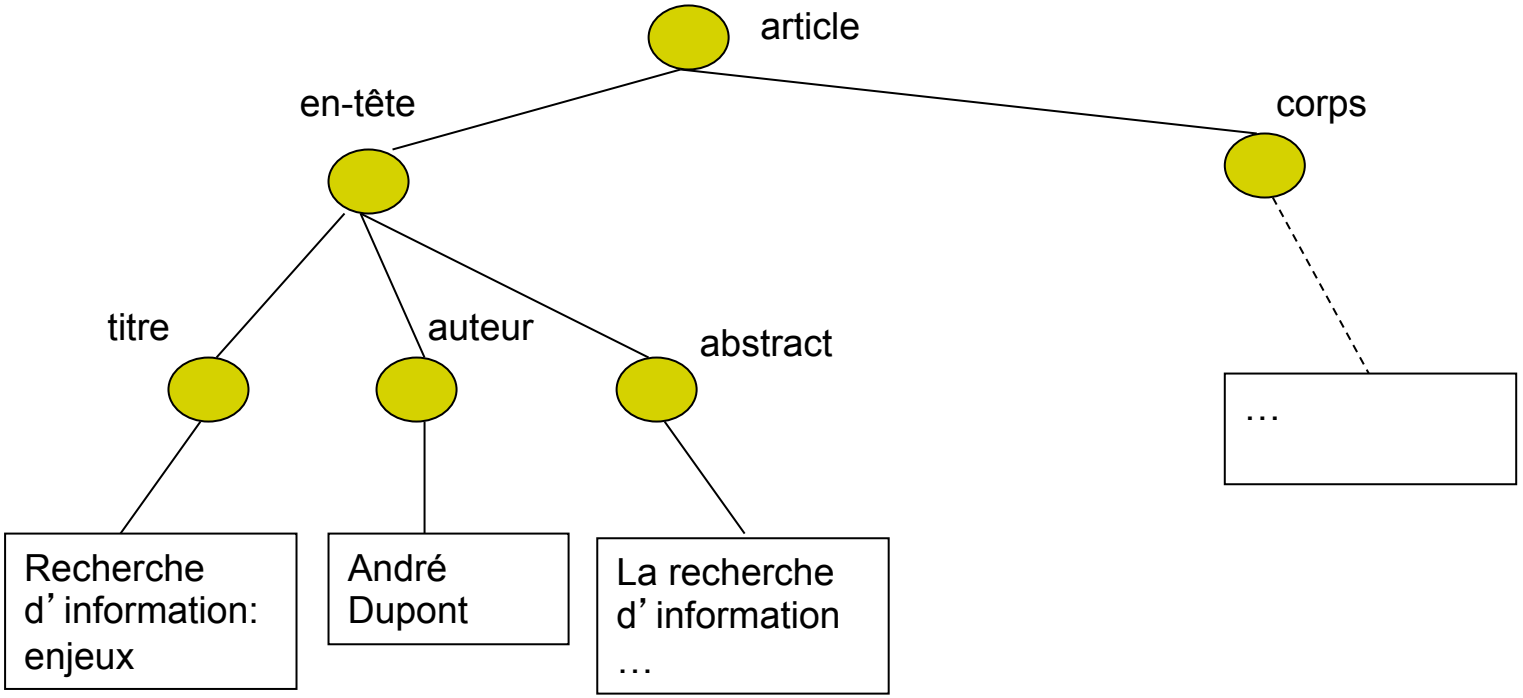
- Deux problèmes
  - Indexation de la structure
  - Indexation des termes
    - Portée des termes d'indexation

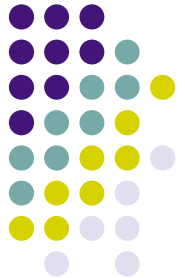
-> comment relier les informations entre elles ?

-> comment représenter les termes dans l'arborescence des documents ?

Indexation du contenu

- Unités disjointes
  - Le texte de chaque nœud de l'index est l'union d'une ou plus de ses parties disjointes



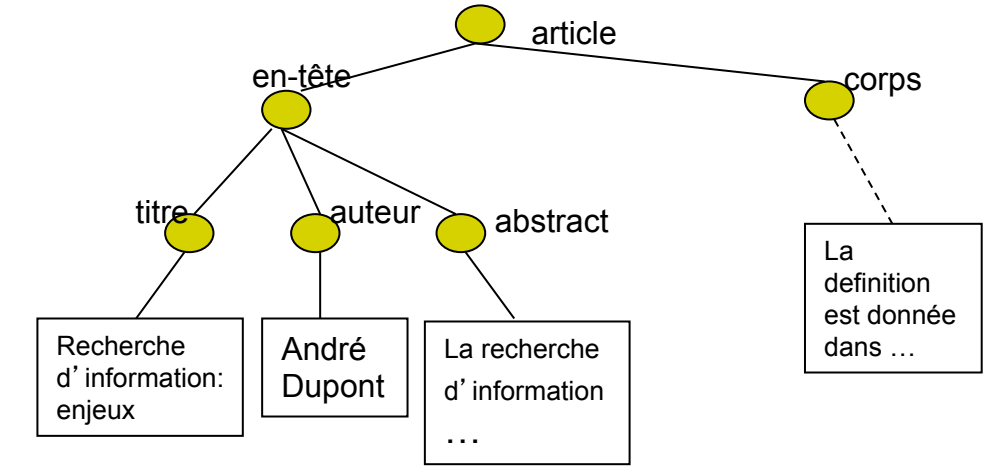


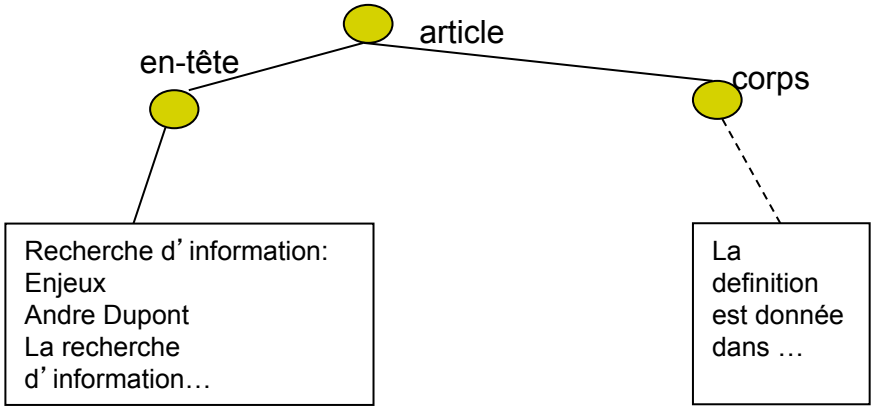
Indexation du contenu

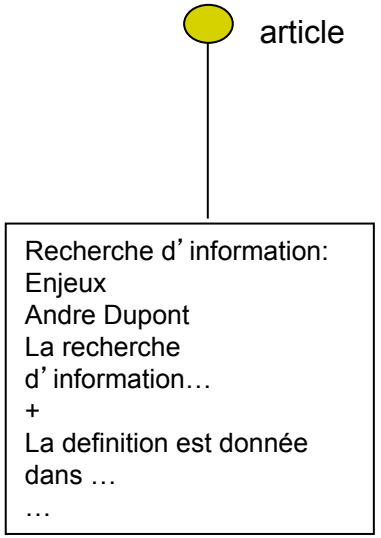
- Sous-arbres imbriqués
  - Le texte complet de chaque nœud de l'index est un document atomique
  - Indexation de tous les sous-arbres des documents
    - Nœuds de l'index imbriqués les uns dans les autres
    - Nombreuses informations redondantes dans l'index

Indexation du contenu

- Sous-arbres imbriqués





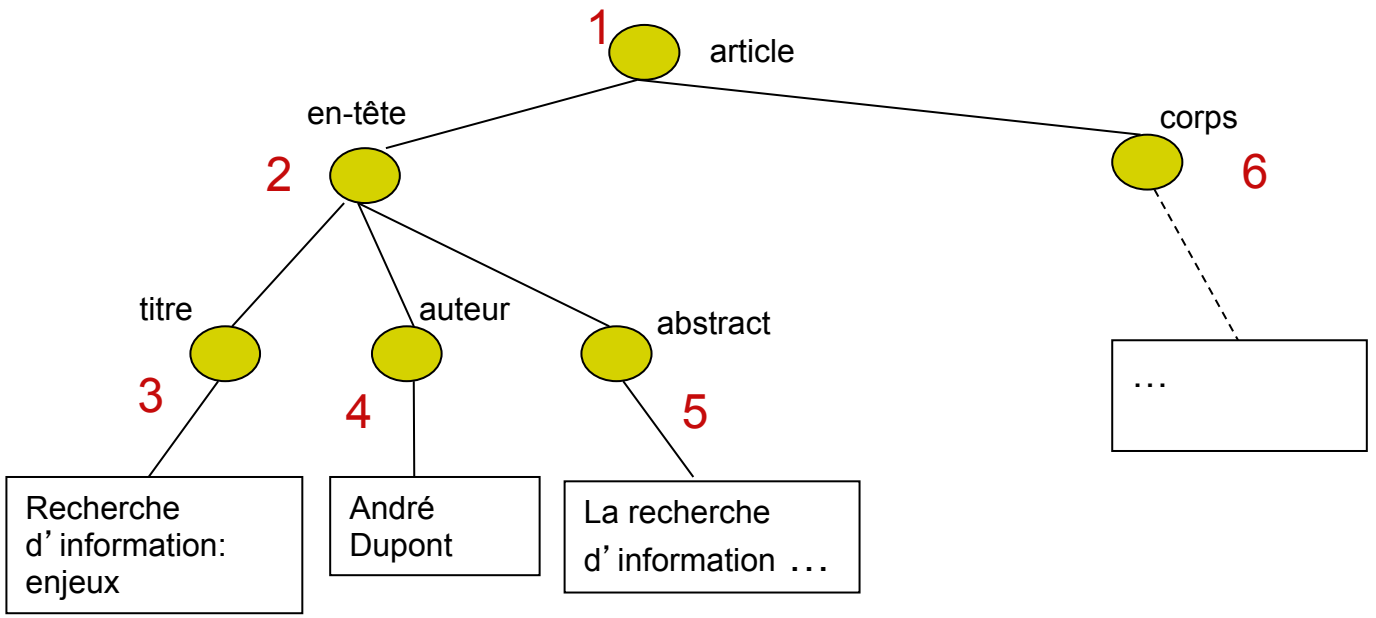


Remarques sur la pondération des termes

- *IDF ?*
  - Il faut s'adapter à la nouvelle granularité des documents
- *IEF* (*Inverse Element Frequency*)
  - $IEF_i = Log\ (E/e_i)$
    - E le nombre d'éléments de la collection
    - e_i le nombre d'éléments contenant le terme i
- Et encore…:
  - *Itdf (Inverse Tag and Document Frequency)*
    - Force discriminatoire d'un terme t par rapport à une balise b pour un document d

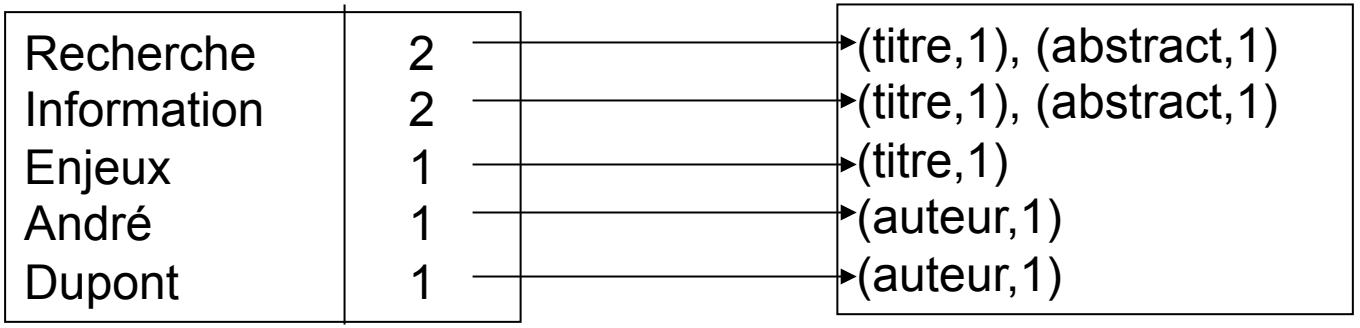
Indexation de la structure

- Doit-on indexer toute la structure ?



Indexation de la structure

- Indexation basée sur des champs
  - Recherche restreinte à certains champs



☹ On perd totalement la structure arborescente des documents

Indexation de la structure

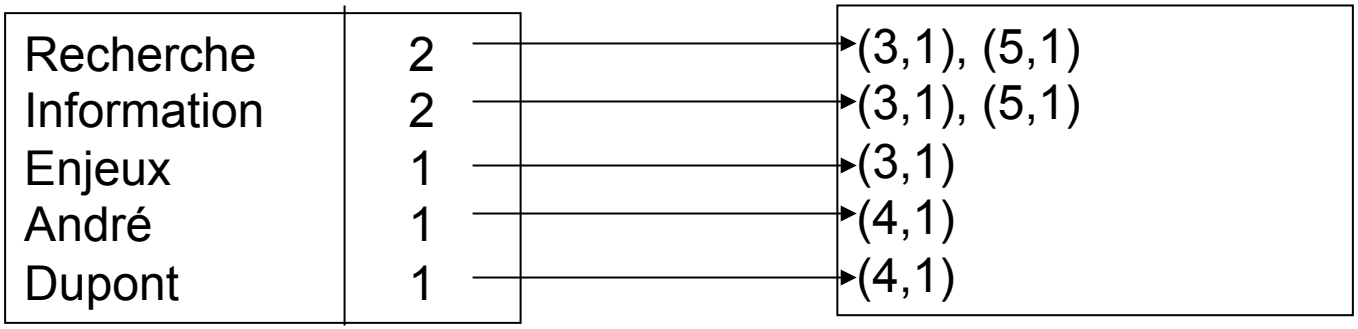
- Indexation basée sur des chemins
  - Recherche pour des valeurs connues de certains éléments ou attributs
    - Pour chaque valeur répertoriée d'un chemin de balises, liste des documents répondant et contenant un élément atteignable par ce chemin et ayant cette valeur

Recherche	2	(/article/en-tête/titre,1), (/article/en-tête/abstract,1)
Information	2	(/article/en-tête/titre,1), (/article/en-tête/abstract,1)
Enjeux	1	(/article/en-tête/titre,1),
André	1	(/article/en-tête/auteur,1)
Dupont	1	(/article/en-tête/auteur,1)

/article	doc1,…
/article/en-tête	doc1,…
/article/en-tête/titre	doc1,…
/article/en-tête/auteur	doc1,…
/article/en-tête/abstract	doc1,…
/article/corps	doc1,…

Indexation de la structure

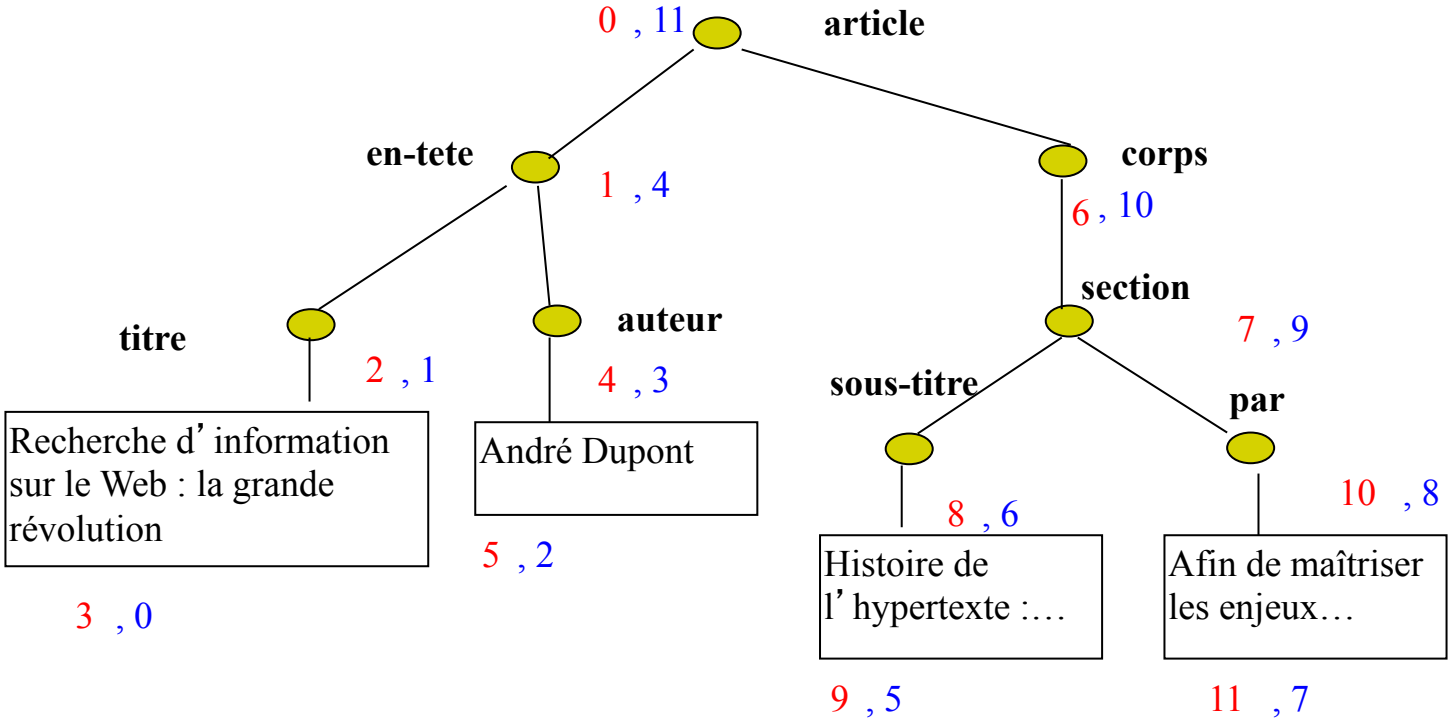
- Indexation basée sur des arbres
  - Permettent de retrouver les relations ancêtres-descendants entre les nœuds des documents



  - Problème: comment numéroter des nœuds ?

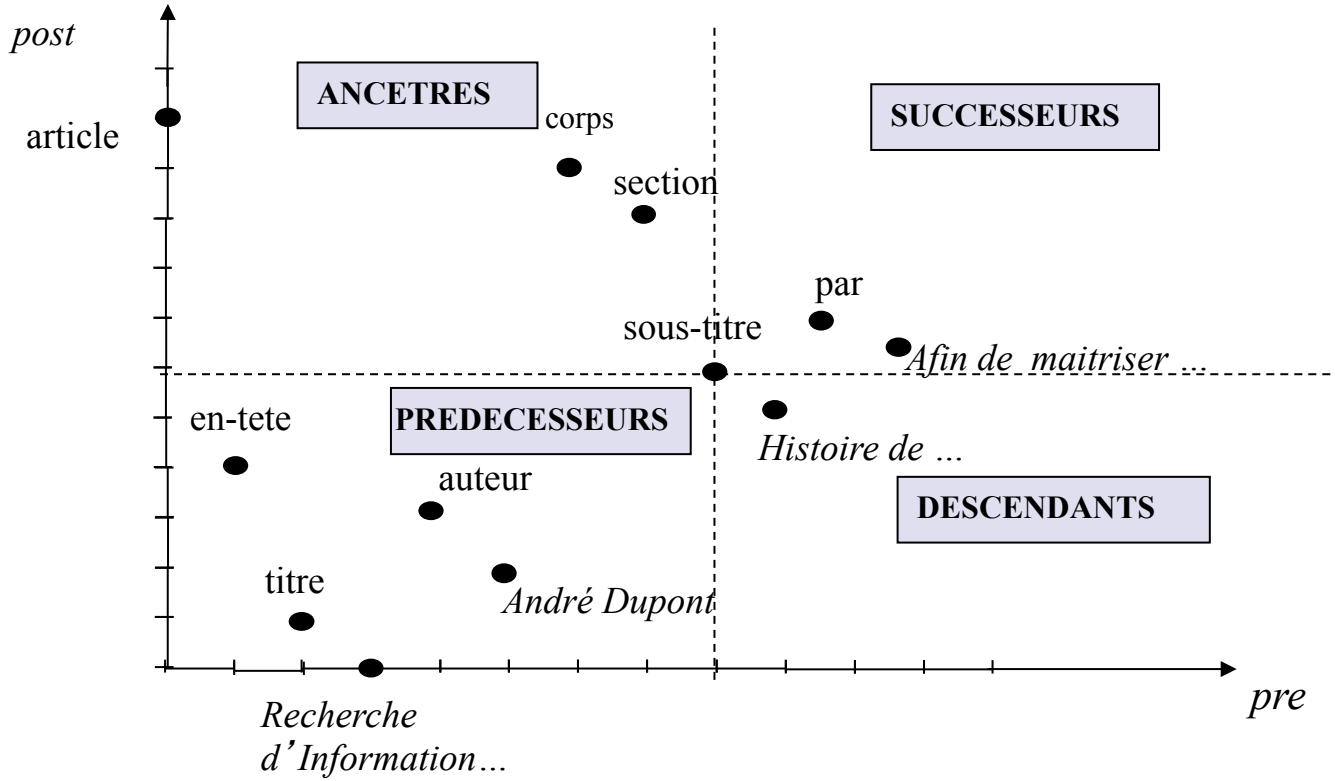
Exemple d'indexation basée sur les arbres

- XPath Accelerator (1)
  - Valeurs de pré-ordre et post-ordre assignées aux noeuds



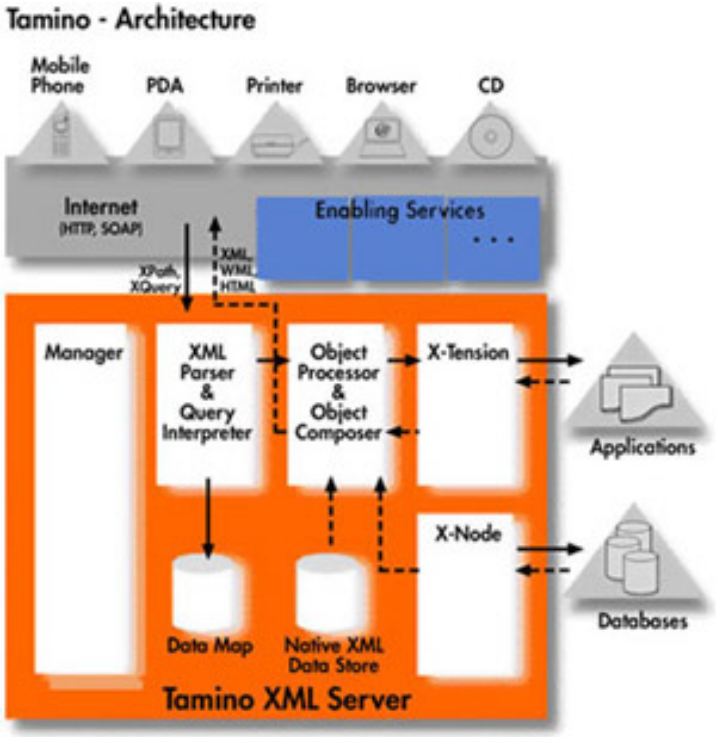
Exemple d'indexation basée sur les arbres

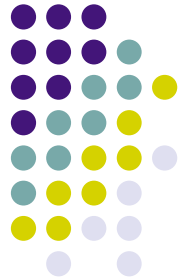
- XPath Accelerator (1)
  - Traitement efficace des relations ancêtres-descendants



Techniques de stockage XML natives

- Systèmes commerciaux
  - TextML Server de IxiaSoft
  - Xylème Zone Server
  - Tamino XML de Software A.G.
  - eXist
  - … et bien d'autres encore

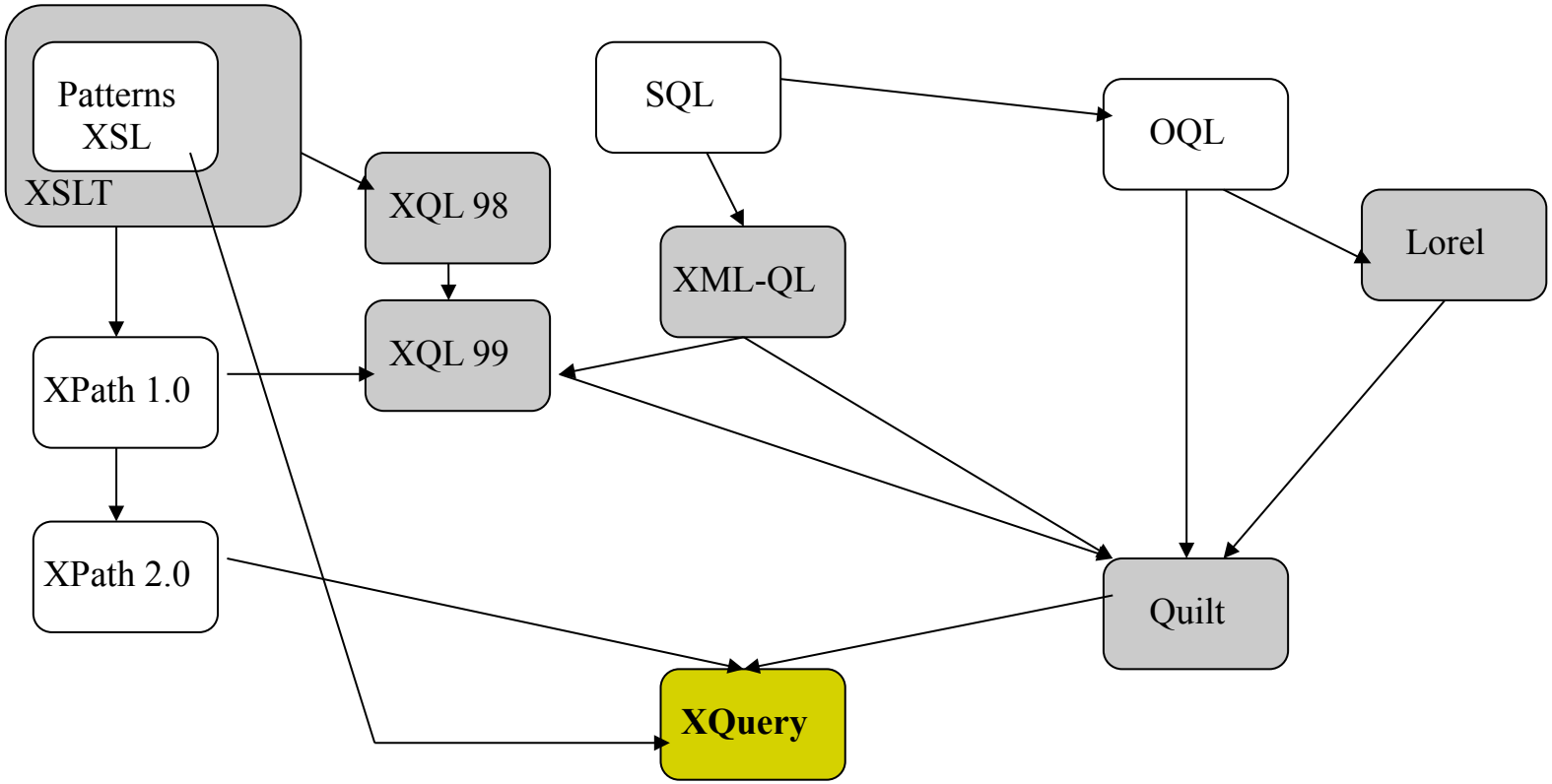




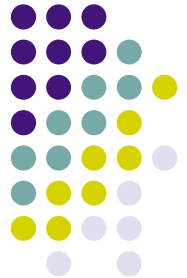
Langages de requêtes

- Deux façons principales d'interroger les collections de documents XML
  1. L'utilisateur n'a pas d'idée précise de ce qu'il recherche
     - Requêtes orientées contenu
  2. Besoin plus précis et connaissance au moins partielle de la collection interrogée
     - Requêtes orientées contenu et structure

Historique des langages d'interrogation

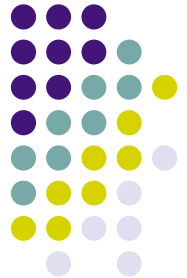


XQuery



- « le SQL de XML »
- Recommandation du W3C
- Les requêtes XQuery
  - Peuvent sélectionner des documents entiers ou des sous-arbres qui répondent à la requête
  - Peuvent construire des documents nouveaux fondés sur ce qui est sélectionné

XQuery



Forme des requêtes (FLWR)

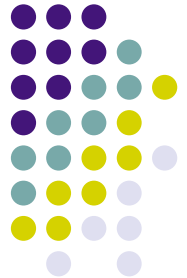
FOR $<var1> in <forêt1> //expression path (éléments du documents

LET $<varn>:=<subtree> //assignation

WHERE <condition> //élagage

RETURN <result> //construction

- Le résultat est une forêt



Exemple de XQuery

fichier : bib.xml

```
<book>
<title> XML: An Introduction</title>
<author>Smith </author> <author>Miller</author>
<publisher>Morgan Kaufmann</publisher>
<year>1998</year>
<price>50</price>
</book>
<book>
<title>XSLT Course</title>
<author> Jones</author>
<publisher>Addison Wesley</publisher>
<year>2000</year>
<price>40</price>
</book>
```

Exemple de XQuery

- Exemple
  - Titre des ouvrages publié par « Morgan Kaufmann » en 1998

```
FOR $b in document("bib.xml")//book
WHERE $b/publisher=« Morgan Kaufmann »
AND $b/year=« 1998 »
RETURN $b/title
```

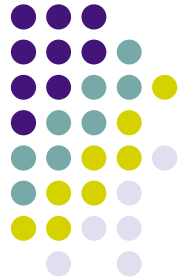
  - $b : parcourt la séquence des éléments book
  - WHERE filtre la liste des tuples ($b/publisher, $b/year)
  - RETURN construit pour chaque tuple le résultat

Autres langages de requêtes

- Xquery n'est pas vraiment adapté à la recherche d'information!
  - Même si Xquery full-text est en cours de normalisation

- Langages orientés RI
  - Souvent des extensions de XPath
  - Exemple, langage NEXI
    - //article[about(.,'recherche information') // section [about (p,'Google')]

Conclusion sur les langages de requêtes



- Historiquement, langages orientés BD
- Langages orientés RI
  - Permettent une recherche non exacte sur le contenu

- Mais…
  - Nombreuses sont les spécifications et rares sont les implémentations

Modèles de recherche

- Approches basées sur la propagation des termes
  - Les termes (pondérés ou non) sont propagés dans l'arborescence des documents
  - La pertinence de chaque élément est évaluée principalement en fonction des termes qu'il contient

- Approches basées sur la propagation des scores de pertinence
  - On calcule des scores de pertinences pour les feuilles de l'arbre
  - Ces scores de pertinences sont propagés et agrégés dans l'arbre des documents

Exemple d'une approche basée sur la propagation des termes

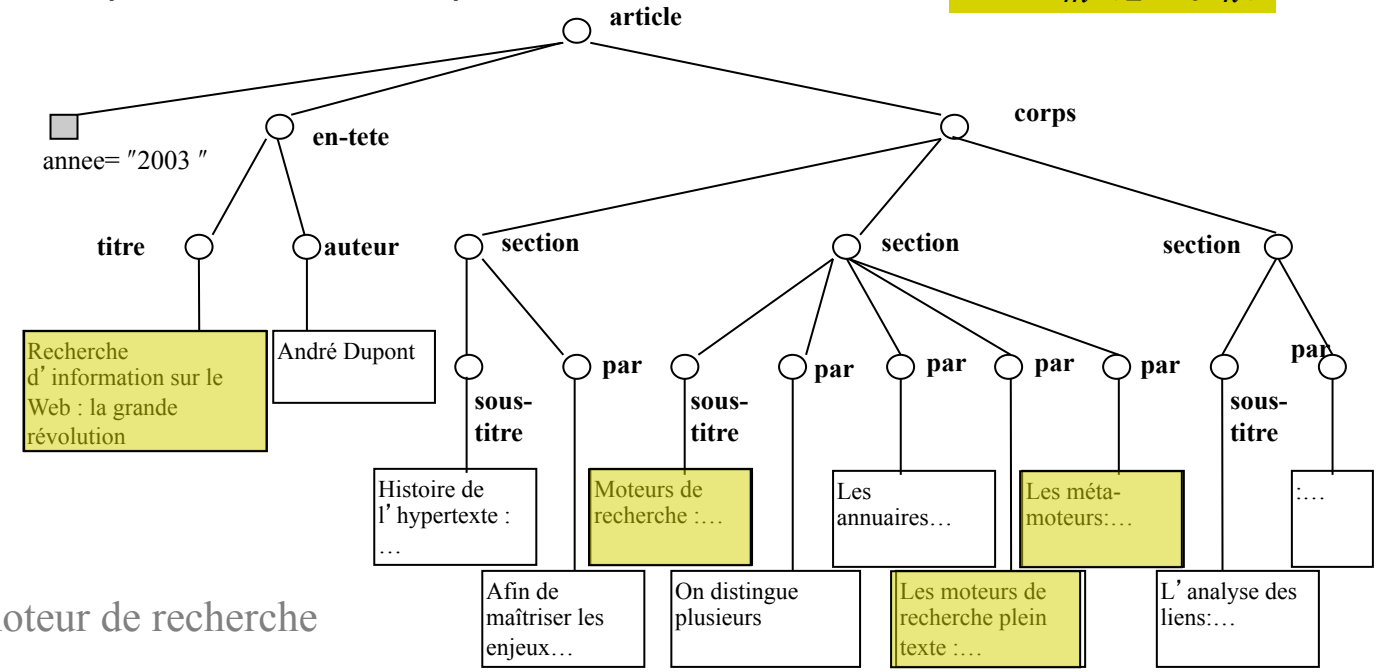
- Adaptation du modèle vectoriel (1)
  - On identifie dans la collection de documents les types d'éléments qui peuvent être des unités d'information potentiellement intéressantes pour l'utilisateur
    - Par exemple, *article, section, paragraphe, …*
  - On crée autant de collections que de types d'éléments
    - Collection d'*article*, collection de *section*, collection de *paragraphe*,…

Exemple d'une approche basée sur la propagation des termes

- Adaptation du modèle vectoriel (2)
  - Pour chaque sous-collection, on crée un index
  - La recherche est effectuée sur chaque sous-index
    - Possibilité d'utiliser la représentation du modèle vectoriel
      - Pondération utilisée souvent basée sur *tf-ief* et la longueur des éléments
      - Poids d'un terme t_j : $w_j^k = f(tf_j^k, tf_j^d idf_j, ief_j, l_k, l_d, \Delta l)$
  - Les résultats des sous-index sont normalisés et une seule liste de résultats est créée

Exemple d'une approche basée sur la propagation de la pertinence

- Pour chaque nœud feuille,
  - un poids de pertinence à la requête est calculé: $RSV_m(q, nf_k)$



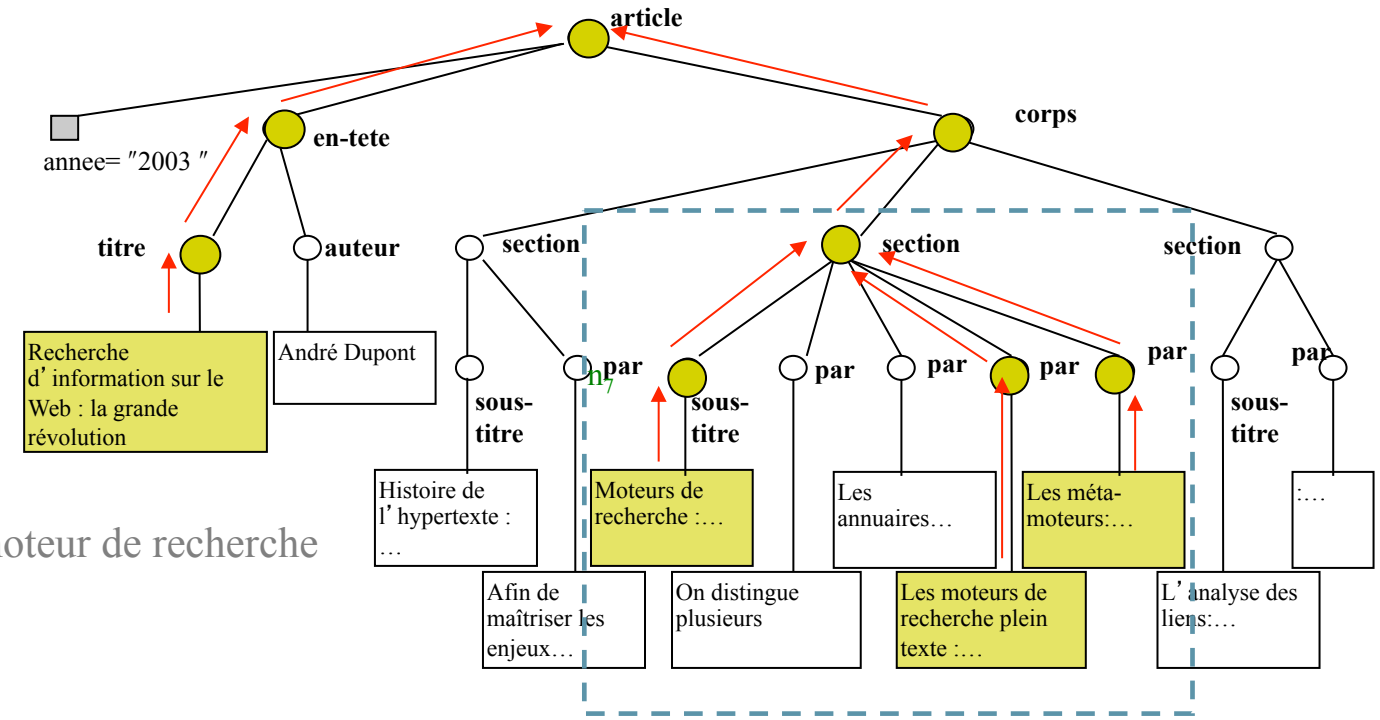
Requête : moteur de recherche

  - Importance de la pondération des termes d'indexation

Exemple d'une approche basée sur la propagation de la pertinence

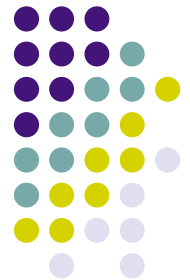
- La pertinence p_n d'un nœud n est ensuite calculée grâce à la **propagation** et à l'**agrégation** des scores des nœuds feuilles

$$p_n = f_k(RSV(q, nf_k), dist(n, nf_k))$$



Requête : moteur de recherche

Traitement des requêtes structurées ?



- Le plus souvent, filtre sur les résultats pour répondre aux conditions de structure

- Certains modèles ont été adapté pour évaluer la pertinence de la structure
  - Exemple du modèle vectoriel dans le cas de la propagation des termes
  - Exemple d'un modèle de propagation des scores

Traitement des requêtes structurés ?

- Exemple d'adaptation du modèle vectoriel

$$RSV(q,e) = \frac{\sum_{(t,c_i)\in q} \sum_{(t,c_k)\in e} w_q(t,c_i) * w_d(t,c_k) * cr(c_i,c_k)}{|q| * |e|}$$

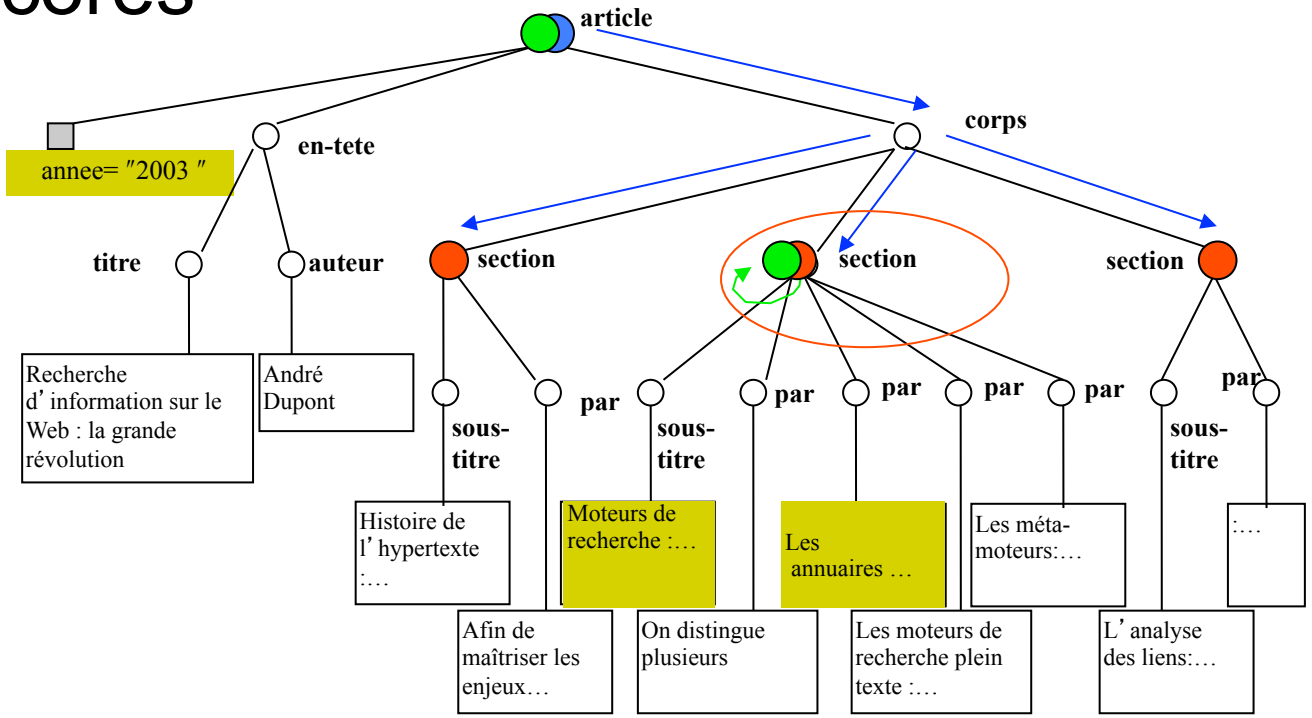
Matching exact: $cr(c_i,c_k) = 1$ si $c_i = c_k$; 0 sinon.

Matching partiel: $cr(c_i,c_k) = \frac{1+|c_i|}{1+|c_k|}$ si c_i est une sous-sequence de c_k; 0, sinon

Exemple: Cr(/article/bibl, /article/bm/bib/bibl/bb)=3/6

Traitement des requêtes structurés ?

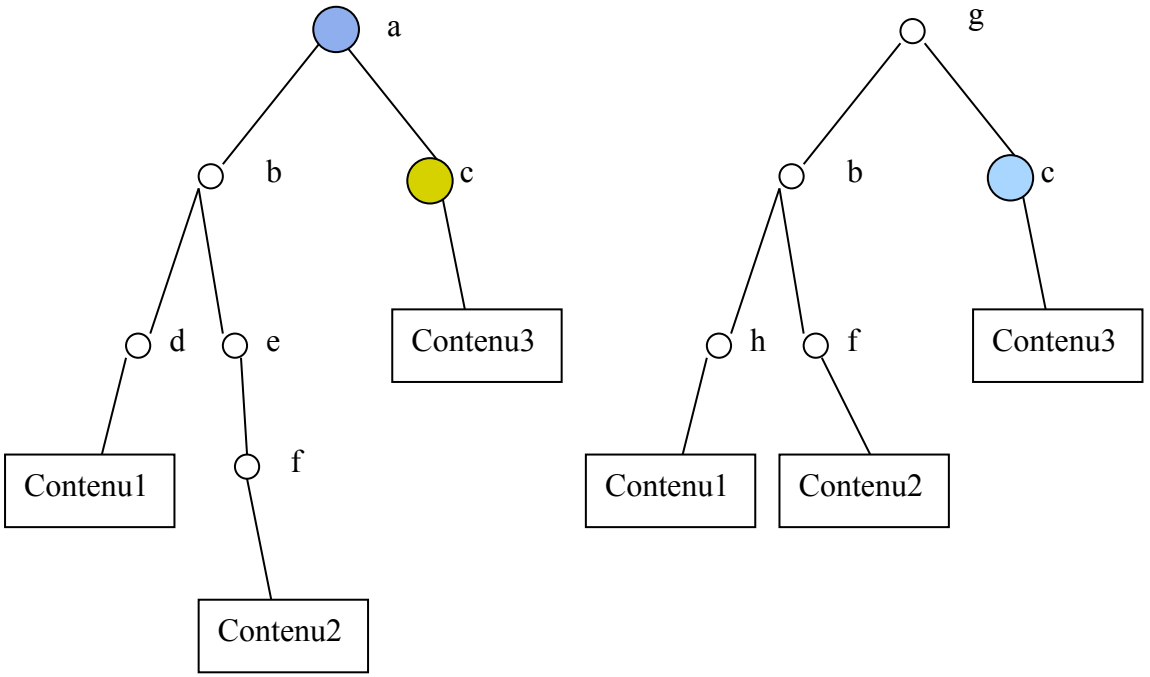
- Exemple d'adaptation pour la propagation des scores



Article[@annee=2003] // ec:section[] // par[annuaire] ET titre[moteur de recherche]

Correspondance partielle de la structure

- //a[contenu1]//i[contenu2]//ec: c[contenu3]
- //a[]//d[]//b[contenu2]

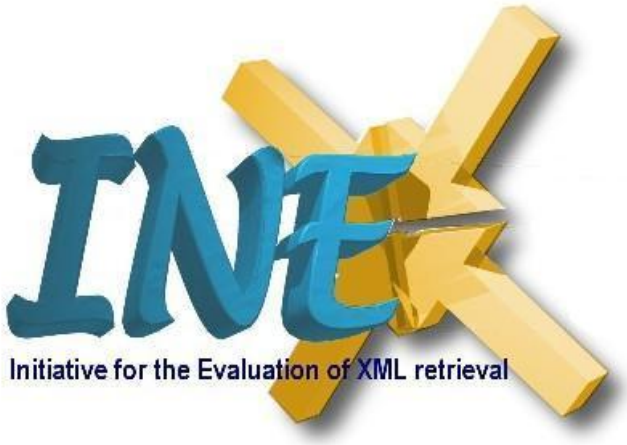


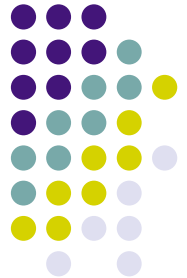
Document 1 ***Document 2***

Evaluation

- Campagne d'évaluation pour la recherche d'information structurée:
  - https://inex.mmci.uni-saarland.de/

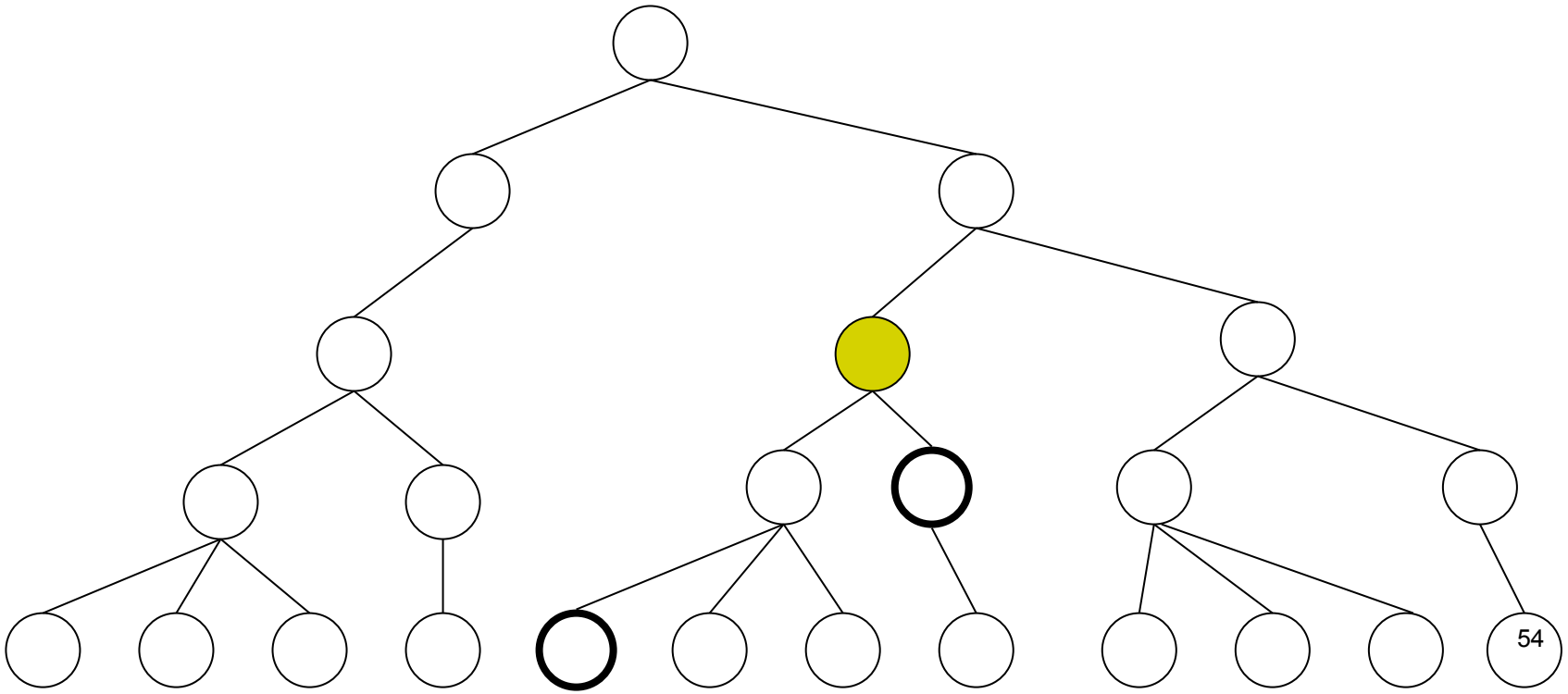
  - Tâches de recherche:
    - Adhoc
    - Relevance Feedback
    - Jeopardy
    - …

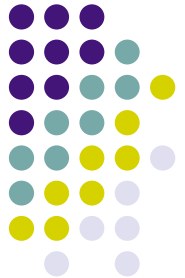




Evaluation

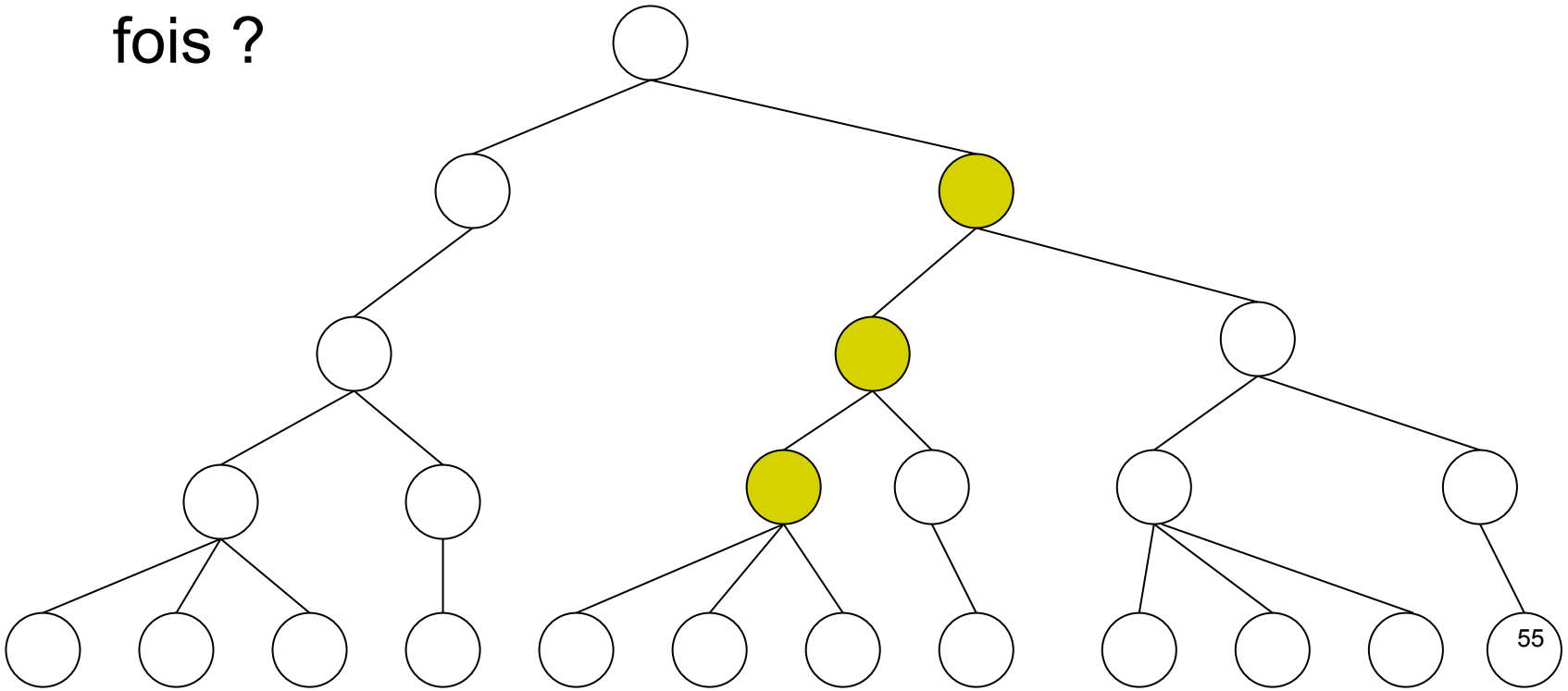
- Difficulté 1: Near misses
  - On retourne un élément voisin d'un élément recherché

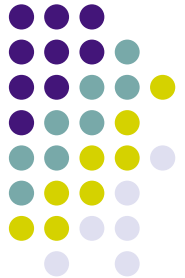




Evaluation

- Difficulté 2: Imbrication des résultats
  - Comment prendre en compte le fait qu'une même information puisse être retournée plusieurs fois ?





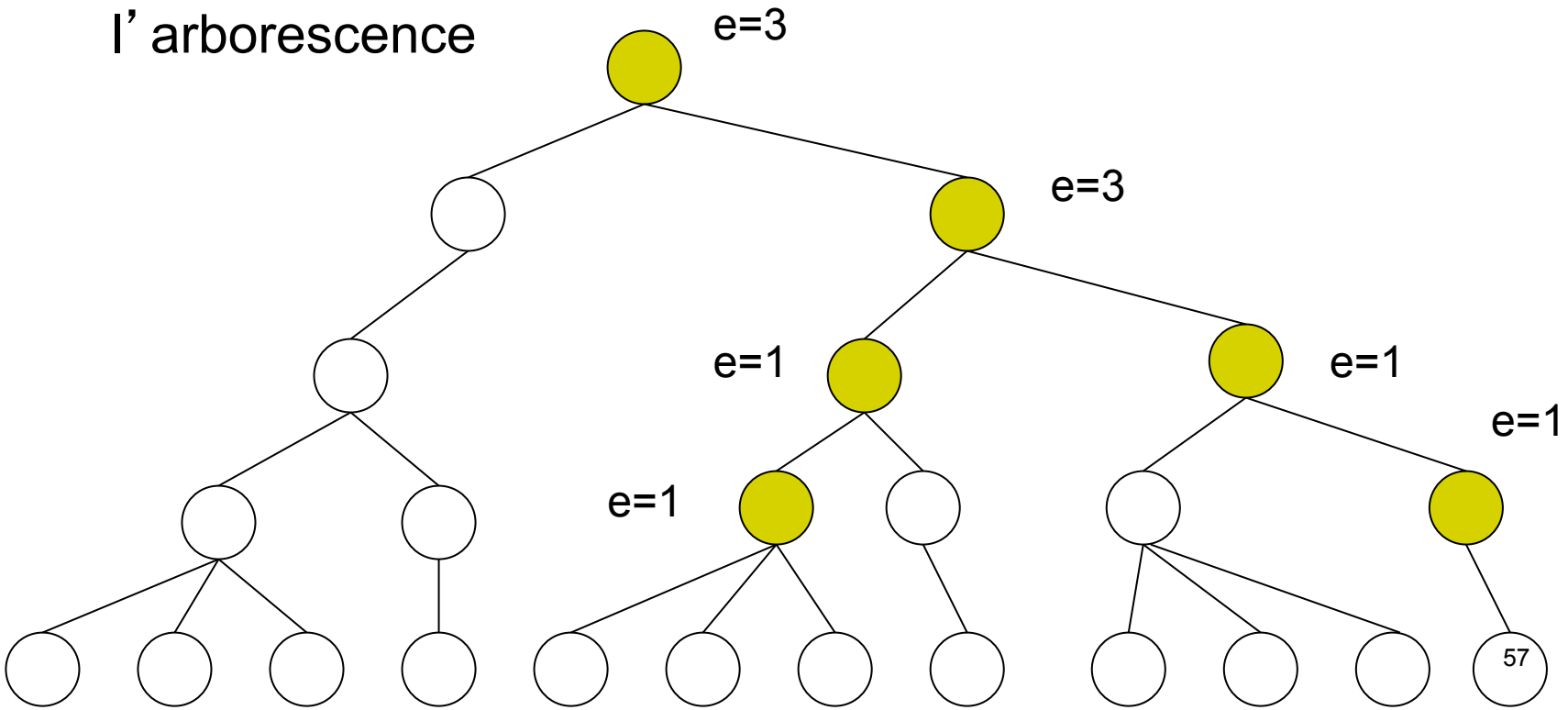
Evaluation

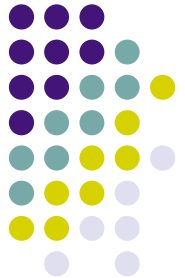
- Difficulté 3: prise en compte des deux dimensions de pertinence :
  - Exhaustivité
  - Spécificité
- Echelle graduelle:
  - Pas, un peu, beaucoup, très exhaustif {0,1,2,3}
  - Pas, un peu, beaucoup, très spécifique {0,1,2,3}

- Si un élément est pertinent, ses ascendants le sont aussi…

Evaluation

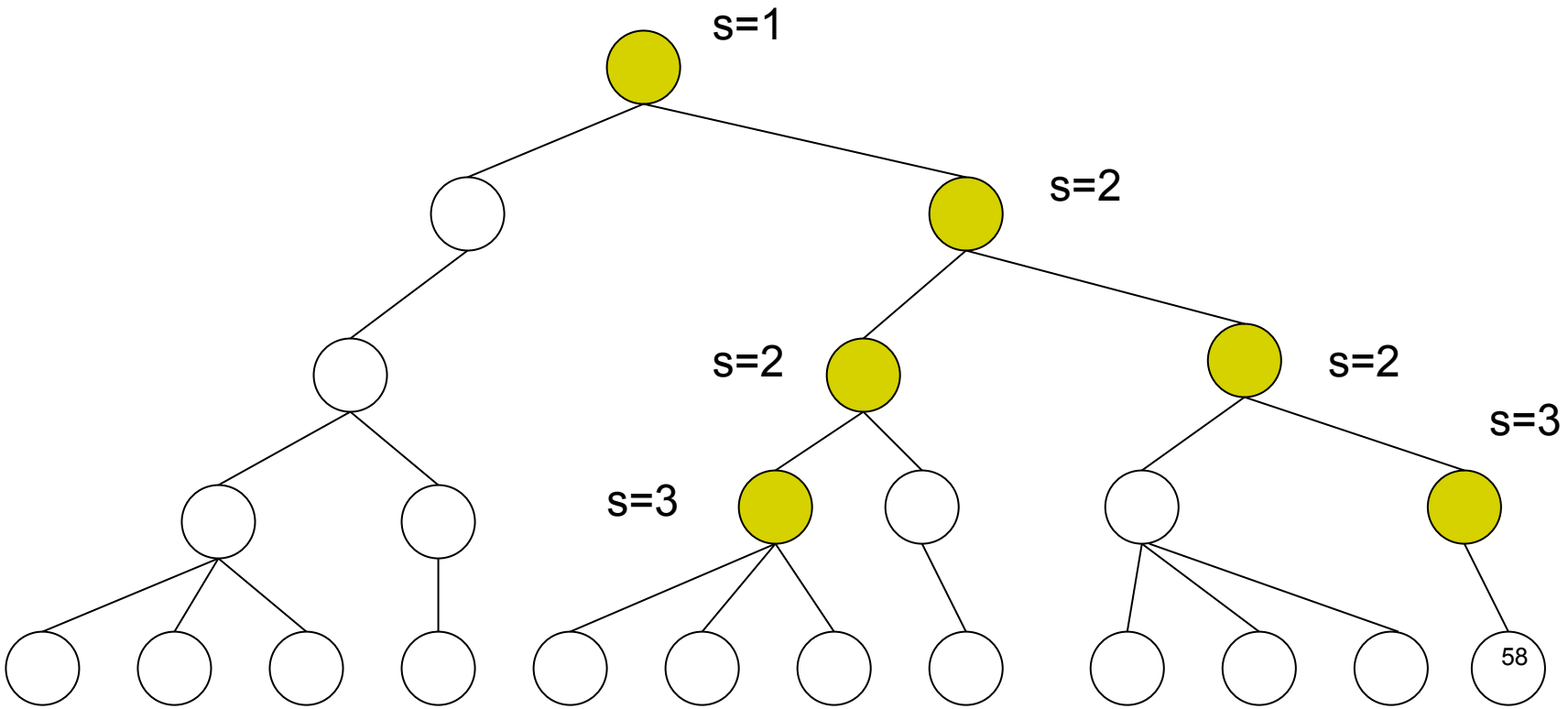
- L'exhaustivité d'un parent est supérieure ou égale à l'exhaustivité de ses fils
  - Le degré d'exhaustivité des éléments est propagé dans l'arborescence





Evaluation

- La spécificité d'un parent est inférieure ou égale à la spécificité de ses fils



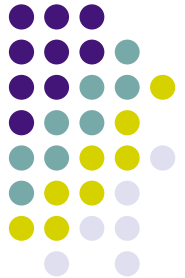
Evaluation

- On doit ensuite transformer les deux degrés de pertinence en une seule valeur de pertinence pour pouvoir utiliser les mesures d'évaluations
  - Fonctions d'agrégation
    - Agrégation stricte
      - Recherche d'éléments très exhaustifs et très spécifiques

$$f_{strict}(e,s) = \begin{cases} 1 \text{ si } e = 3 \text{ et } s = 3 \\ 0 \text{ sinon} \end{cases}$$

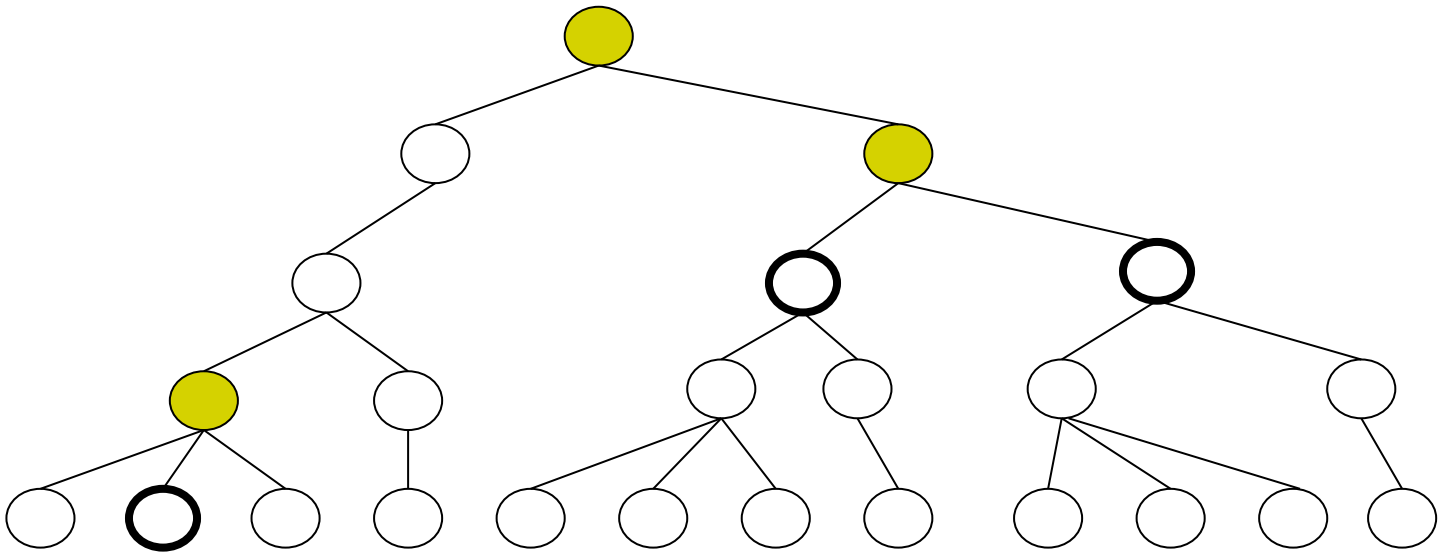
    - Agrégation généralisée
      - Evaluation des éléments selon leur degré de pertinence

$$f_{generalisee}(e,s) = \begin{cases} 1 \text{ si } (e,s) = (3,3) \\ 0.75 \text{ si } (e,s) = (2,3) \text{ ou } (3,\{2,1\}) \\ 0.5 \text{ si } (e,s) = (1,3) \text{ ou } (2,\{2,1\}) \\ 0.25 \text{ si } (e,s) = (1,2) \text{ ou } (1,1) \\ 0 \text{ si } (e,s) = (0,0) \end{cases}$$

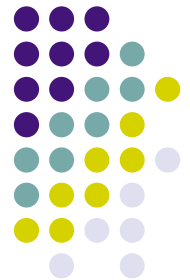


Evaluation

- Metriques
  - Rappel/précision?
    - Pas adapté : par exemple, système retournant systématiquement un élément plus petit



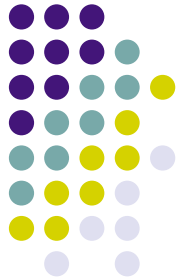
Exemple de mesure: Gain cumulé



- La mesure xCG cumule les scores de pertinence des éléments de la liste des résultats
- Etant donnée une liste triée d'éléments xCG dans laquelle les identifiants d'éléments sont remplacés par leur score de pertinence, le gain cumulé au rang i, noté xCG[i] est la somme des pertinences jusqu'à ce rang
  - Depend des jugements et de la liste elle-même

$$xCG[i] = \sum_{j=1}^{i} xG[i]$$

  - Exemple:
    - Soit xGi=<2,1,0,1,0,0> un vecteur de gain jusqu'au rang i
    - Le vecteur de gain cumulé sera <2,3,3,4,4,4>

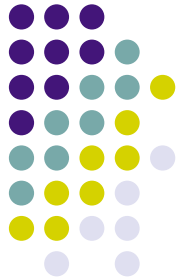


Gain cumulé

- Vecteur de gain idéal xCI à partir de la base de rappel, en cumulant les scores de pertinences des éléments triés par ordre décroissant
  - Base de rappel = base des éléments pertinents dans le corpus

- Le xCG est ensuite comparé au gain idéal:

$$nxCG[i] = \frac{xCG[i]}{xCI[i]}$$



Gain cumulé

- Pour un rang donné i, le gain cumulé nxCG[i] reflète le gain relatif de l'utilisateur accumulé jusqu'à ce rang, comparé à ce qu'il aurait pu atteindre si le système avait produit une liste triée optimale

Evaluation

Aucune mesure ne fait cependant consensus…

Présentation des résultats de recherche

- Renvoyer à l'utilisateur une simple liste d'éléments n'est pas suffisant
  - Il faut supprimer ou réduire les éléments imbriqués
  - Les éléments d'un même document peuvent être groupés
  - Autre possibilité : renvoyer un seul élément par document (Best Entry Point)

dbdk_training in **Baseline System**

Search

<XML> HyREX </XML>

query was: text classification naive bayes
Results **1** - **10** of **100**.
Result pages: **1 2 3 4 5 6 7 8 9 10 next**

Search Result

1: (0.247) **Scalable Feature Mining for Sequential Data**
Neal Lesh Mitsubishi Electric Research Lab Mohammed J. Zaki Rensselaer Polytechnic Institute Mitsunori Ogihara University of Rochester
Result path: `/article[1]/bdy[4]/sec[5]`

2: (0.204) **Probability and Agents**
Marco G. Valtorta University of South Carolina mgv@cse.sc.edu Michael N. Huhns University of South Carolina huhns@sc.edu
Result path: `/article[1]/bdy[4]/sec[3]`

3: (0.176) **Combining Image Compression and Classification Using Vector Quantization**
Karen L. Oehler Member IEEE Robert M. Gray Fellow IEEE
Result path: `/article[1]/bdy[4]/sec[4]/ss1[2]/ss2[4]`

4: (0.175) **Text-Learning and Related Intelligent Agents: A Survey**
Dunja Mladenic J. Stefan Institute
Result path: `/article[1]/bm[5]/app[4]/sec[5]`

5: (0.175) **Detecting Faces in Images: A Survey**
Ming-Hsuan Yang Member IEEE David J. Kriegman Senior Member IEEE Narendra Ahuja Fellow IEEE
Result path: `/article[1]/bdy[4]/sec[2]/ss1[9]/ss2[10]`

Baseline system

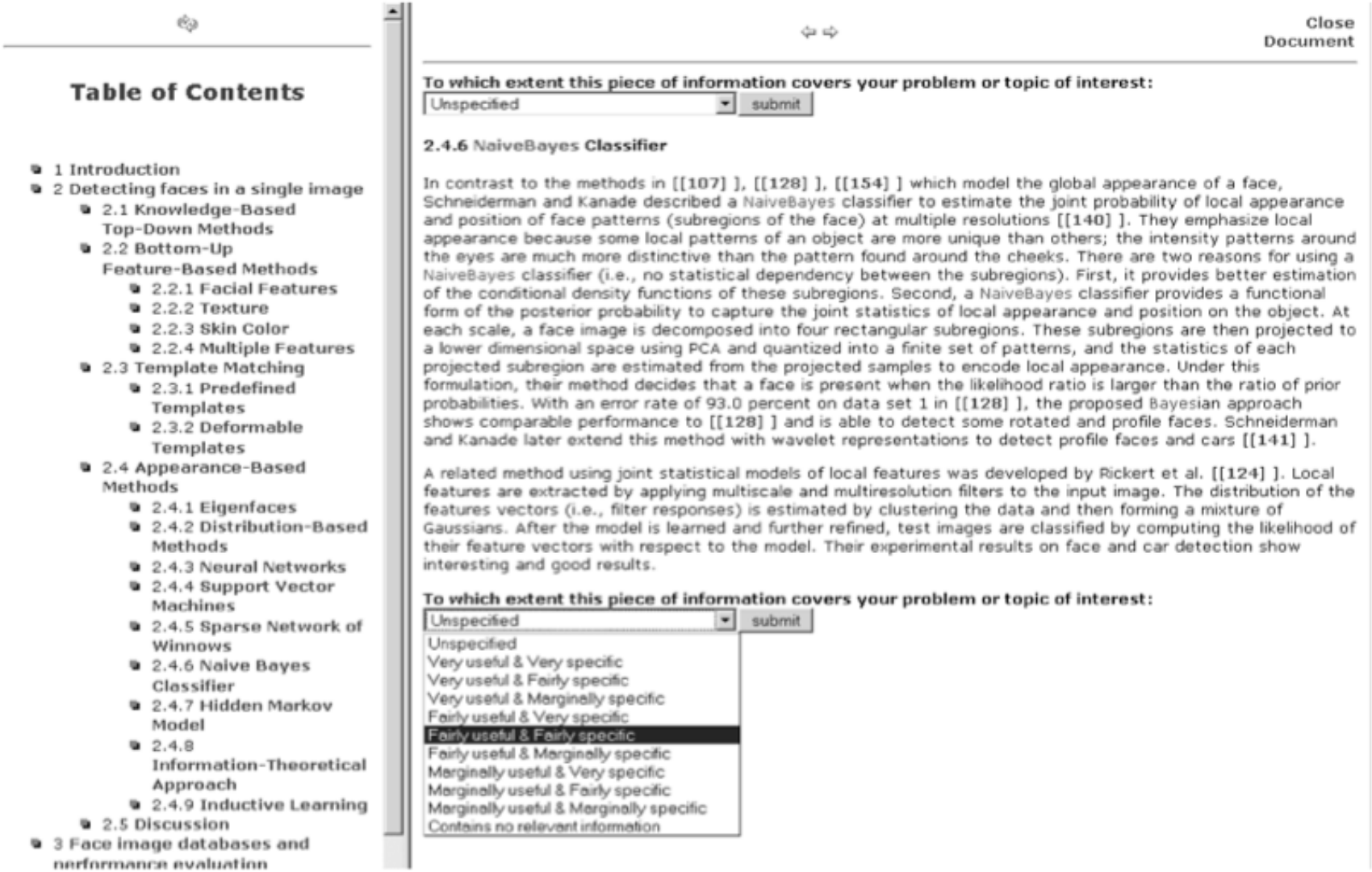


Table of Contents

- 1 Introduction
- 2 Detecting faces in a single image
  - 2.1 Knowledge-Based Top-Down Methods
  - 2.2 Bottom-Up Feature-Based Methods
    - 2.2.1 Facial Features
    - 2.2.2 Texture
    - 2.2.3 Skin Color
    - 2.2.4 Multiple Features
  - 2.3 Template Matching
    - 2.3.1 Predefined Templates
    - 2.3.2 Deformable Templates
  - 2.4 Appearance-Based Methods
    - 2.4.1 Eigenfaces
    - 2.4.2 Distribution-Based Methods
    - 2.4.3 Neural Networks
    - 2.4.4 Support Vector Machines
    - 2.4.5 Sparse Network of Winnows
    - 2.4.6 Naive Bayes Classifier
    - 2.4.7 Hidden Markov Model
    - 2.4.8 Information-Theoretical Approach
    - 2.4.9 Inductive Learning
  - 2.5 Discussion
- 3 Face image databases and performance evaluation

Close Document

To which extent this piece of information covers your problem or topic of interest:
Unspecified submit

2.4.6 NaiveBayes Classifier

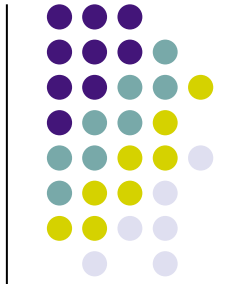
In contrast to the methods in [[107]], [[128]], [[154]] which model the global appearance of a face, Schneiderman and Kanade described a NaiveBayes classifier to estimate the joint probability of local appearance and position of face patterns (subregions of the face) at multiple resolutions [[140]]. They emphasize local appearance because some local patterns of an object are more unique than others; the intensity patterns around the eyes are much more distinctive than the pattern found around the cheeks. There are two reasons for using a NaiveBayes classifier (i.e., no statistical dependency between the subregions). First, it provides better estimation of the conditional density functions of these subregions. Second, a NaiveBayes classifier provides a functional form of the posterior probability to capture the joint statistics of local appearance and position on the object. At each scale, a face image is decomposed into four rectangular subregions. These subregions are then projected to a lower dimensional space using PCA and quantized into a finite set of patterns, and the statistics of each projected subregion are estimated from the projected samples to encode local appearance. Under this formulation, their method decides that a face is present when the likelihood ratio is larger than the ratio of prior probabilities. With an error rate of 93.0 percent on data set 1 in [[128]], the proposed Bayesian approach shows comparable performance to [[128]] and is able to detect some rotated and profile faces. Schneiderman and Kanade later extend this method with wavelet representations to detect profile faces and cars [[141]].

A related method using joint statistical models of local features was developed by Rickert et al. [[124]]. Local features are extracted by applying multiscale and multiresolution filters to the input image. The distribution of the features vectors (i.e., filter responses) is estimated by clustering the data and then forming a mixture of Gaussians. After the model is learned and further refined, test images are classified by computing the likelihood of their feature vectors with respect to the model. Their experimental results on face and car detection show interesting and good results.

To which extent this piece of information covers your problem or topic of interest:
Unspecified submit

Unspecified
Very useful & Very specific
Very useful & Fairly specific
Very useful & Marginally specific
Fairly useful & Very specific
Fairly useful & Fairly specific
Fairly useful & Marginally specific
Marginally useful & Very specific
Marginally useful & Fairly specific
Marginally useful & Marginally specific
Contains no relevant information

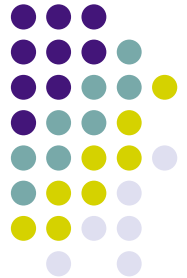
En conclusion



- Pas vraiment de « gagnant » en termes de modèles de recherche

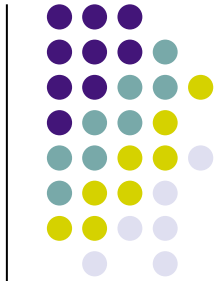
- Ce qui semble bien fonctionner
  - Utilisation de la taille des éléments
  - Contextualisation par le document

Perspectives (1)



- Gestion de structures hétérogènes
  - Pour l'interrogation
  - Pour la recherche
- Recherche de contenus multimedia
  - Notamment dans le domaine médical
- Aggregated search… (encore)
- Mais encore…
  - Découverte de services Web
  - Link the Wiki
  - Entity ranking
  - Linked Data

Perspectives (2)



Hôtel Crillon Paris
Le Crillon à Paris est un hôtel luxueux, idéalement situé sur l'élégante avenue de Suffren. Sa situation géographique incomparable vous permet de pratiquer vos loisirs ou bien de vous consacrer à votre travail.

image-Crillon-Paris.jpg

A quelques minutes de l'hôtel, vous trouverez de très bons restaurants, ainsi que des rues commerçantes et toutes sortes de

divertissements.

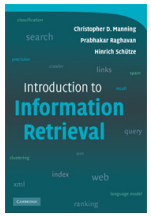
Un document multimédia

<titre> Hôtel CrillonParis**</titre>**
<paragraphe>

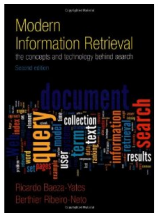
Le Crillon à Paris est un hôtel luxueux, idéalement situé sur l'élégante avenue de Suffren. Sa situation géographique incomparable vous permet de pratiquer vos loisirs ou bien de vous consacrer à votre travail.
</paragraphe>
<image>
<nomImage> Photo-Crillon-Paris.jpg**</nomImage>**
<caption> hotel Crillon paris interieur</caption>
</image>
<paragraphe>
A quelques minutes de l'hôtel, vous trouverez de très bons restaurants, ainsi que des rues commerçantes et toutes sortes de divertissements.
</paragraphe>

Fragment XML associé

Pour conclure, si vous voulez en savoir plus sur la recherche d'information…



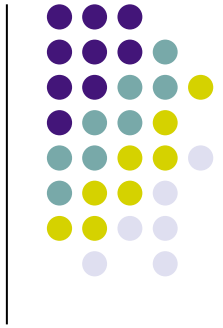
- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, **Introduction to Information Retrieval**, Cambridge University Press. 2008. *www.informationretrieval.org*



- Ricardo Baeza-Yates, Berthier Ribeiro-Netto, **Modern Information retrieval**, ACM Press Book, 2010

- Ayse Goker, John Davies**, Information Retrieval: Searching in the 21st Century,** Wiley, 2010

Sur le Web:



- http://www.sigir.org/
  - Avec ressources, (collections, moteurs open source) papiers importants,…
- http://www.ir-facility.org/

- http://singhal.info/
  - Du côté de Google