www.mas-ict.ch

Hes·so
Haute Ecole Spécialisée de Suisse occidentale

Module: XML et les bases de données

XQuery *Full-Text*

W3C Recommendation 17 March 2011

Houda Chabbi Drissi

houda.chabbi@hefr.ch

Sources:
http://www.w3.org/TR/xpath-full-text-10/
http://www.kc.tsukuba.ac.jp/colloqium/050510.pdf

Plan

1. Introduction: Xquery – IR (Information Retrieval)
2. Syntaxe XFT
3. Annexe: Modèle sémantique: ALLMatches

Plan

1. Introduction: Xquery – IR(Information Retrieval)
2. Syntaxe XFT
3. Annexe: Modèle sémantique: ALLMatches

Xquery

- Permet de requêter la structure XML.
- Permet de requêter sur le contenu textuel = chaîne de caractères:
  - Fonction: boolean fn:contains(string, substring)
  - Exemple:

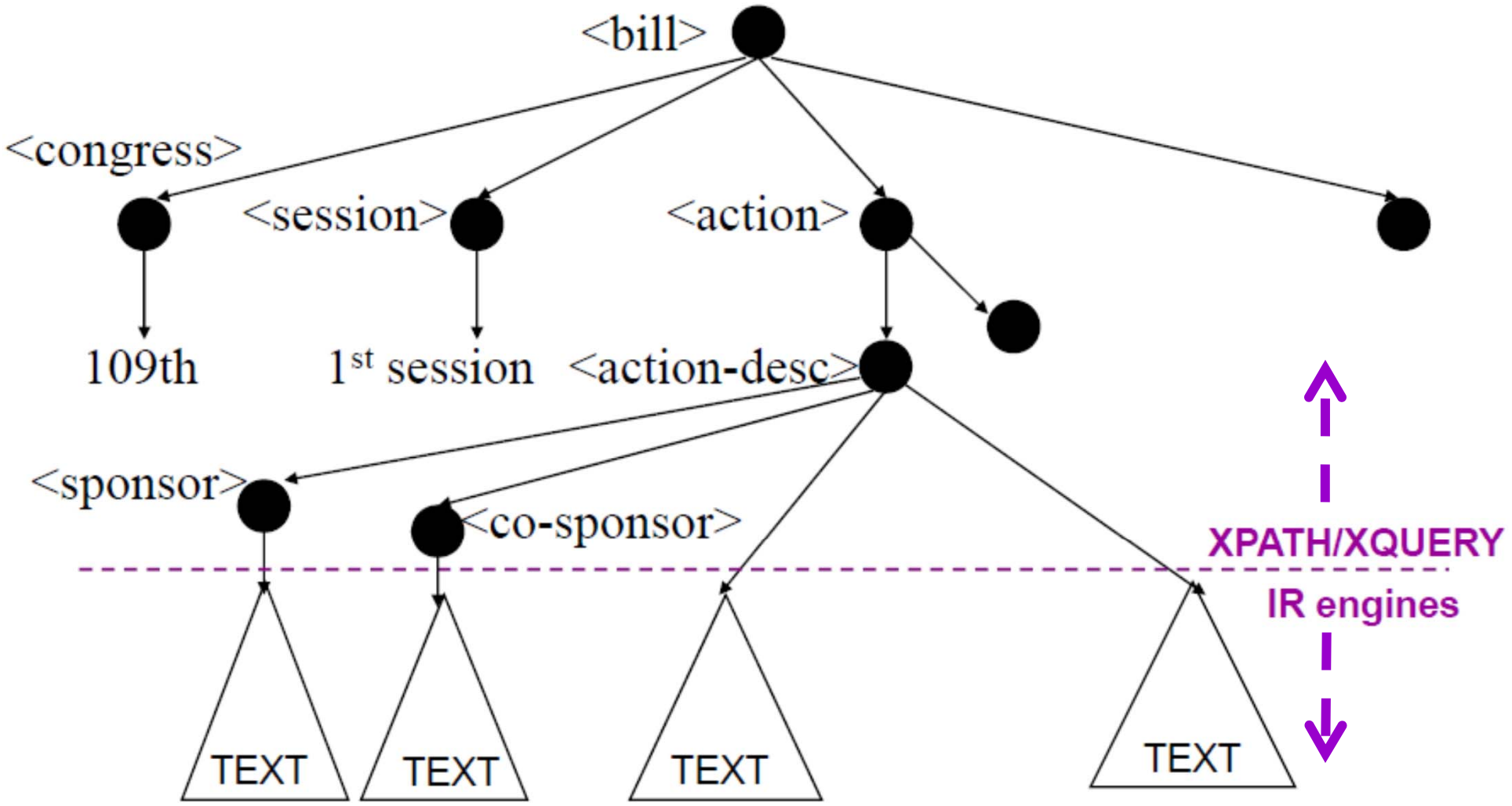
```
for $b in /books/book
let $text := $b/fn:string()
where    fn:contains($text, "web site") and
                fn:contains($text,  "usability")
return <result> <title> {$b/title} </title> </result>
```

Besoins

- Aller au-delà de la chaîne de caractères vers du texte:
  - Construction grammaticale
  - Sémantique (thésaurus)
  - Mesure de similarité
  - Expression regulière
  - Proximité
  - …

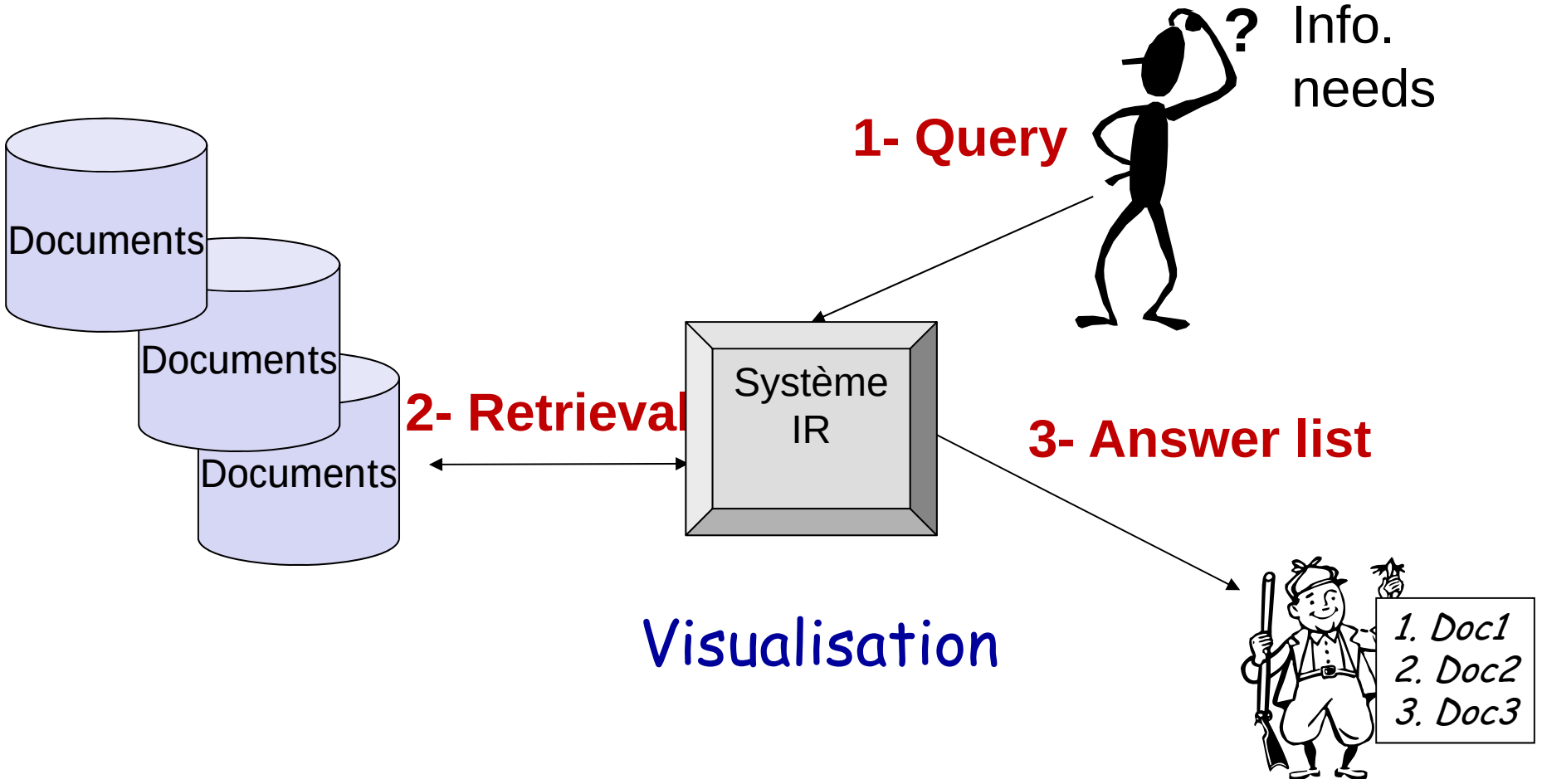
→ Monde le l'IR (Information Retrieval)

But: DB et IR



http://thomas.loc.gov/home/gpoxmlc109/h2739_ih.xml

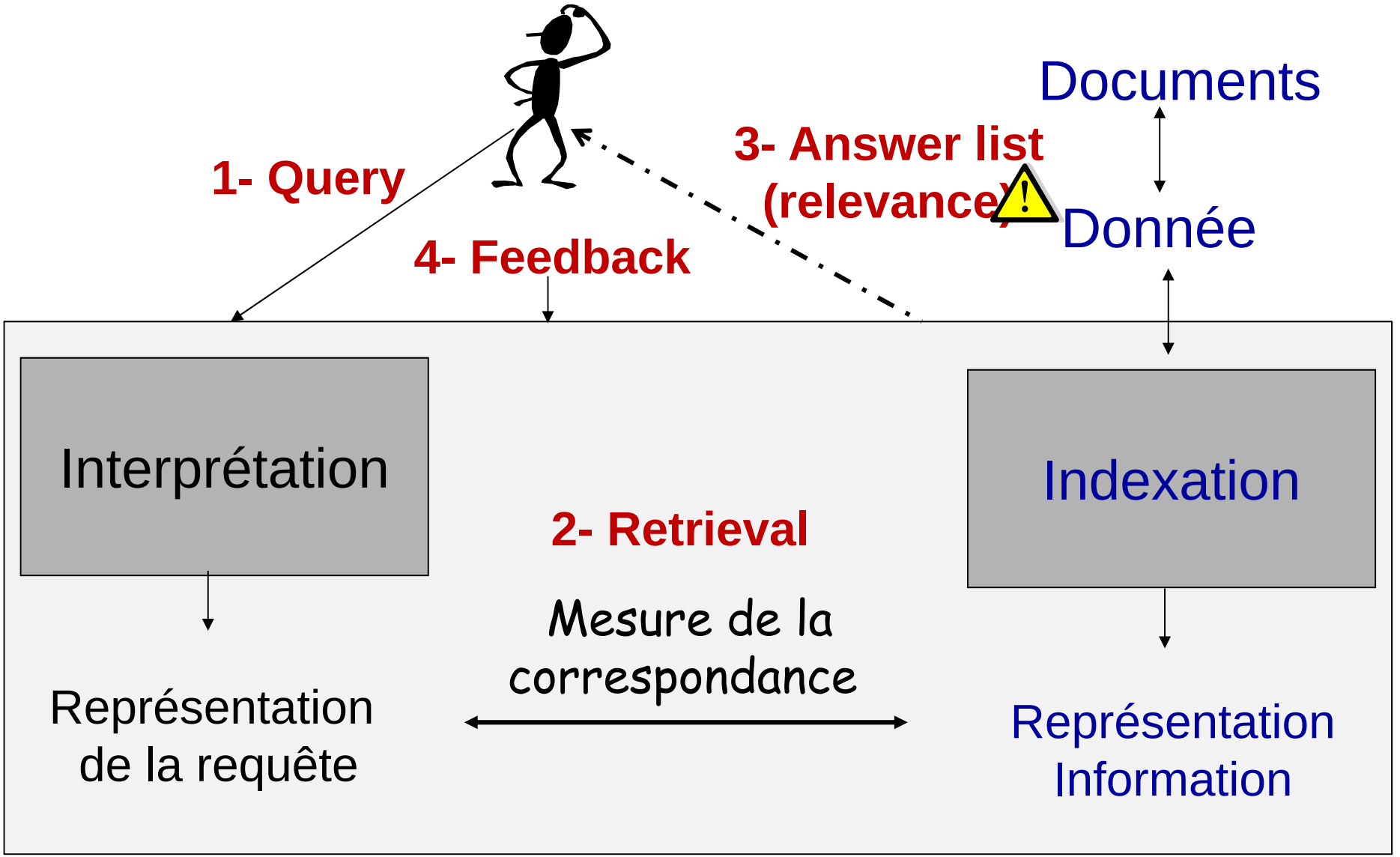
Modèle conceptuel de l'IR



Source:
Document collections

Résultat:
Ranked documents

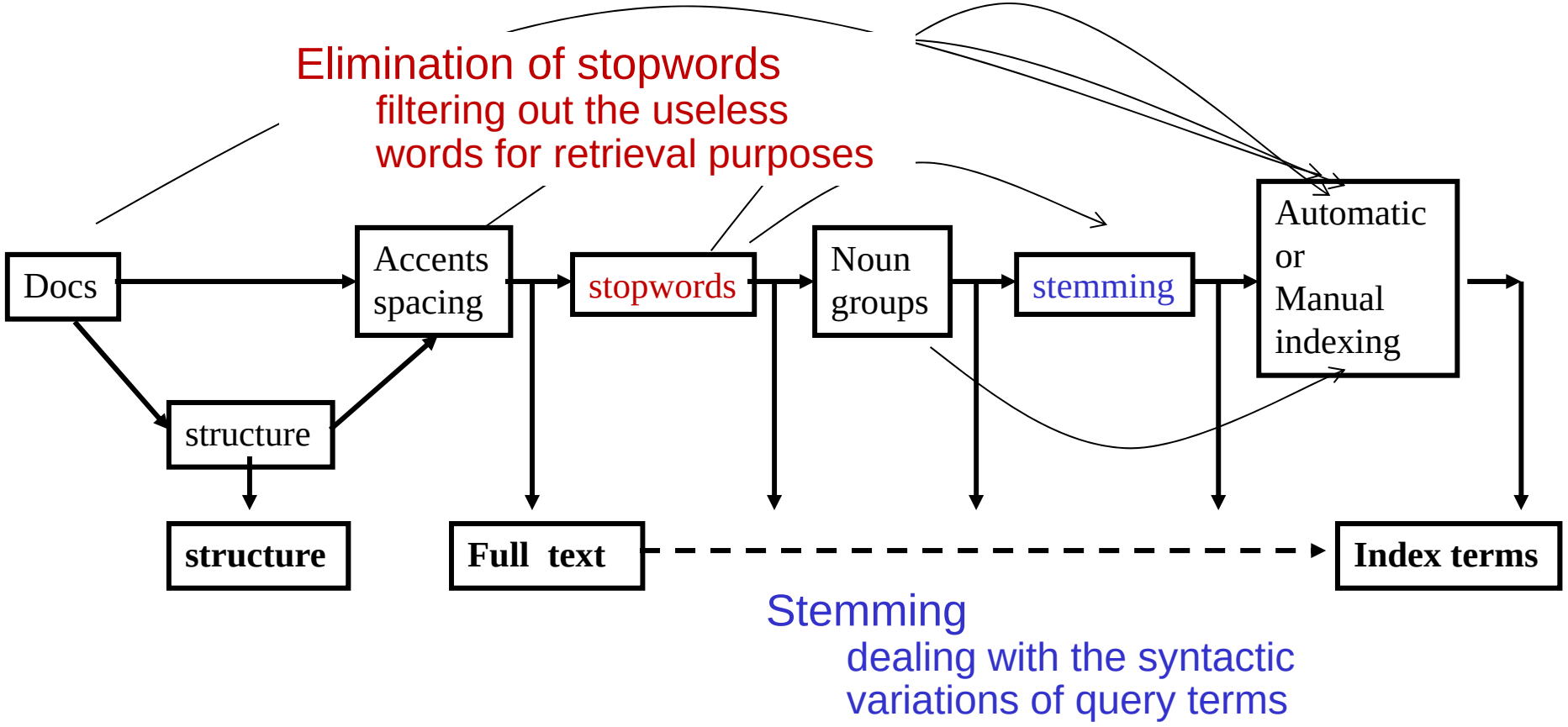
Modèle global de l'IR



La pertinence (relevance) – IR → score

- La pertinence mesure l'adéquation du résultat obtenu par rapport à la réponse attendue.
  - Pour son calcul, peuvent être utilisés: le nombre de termes de la requête trouvés dans les informations, prise en compte de leur localisation dans la structure des documents : titre, corps, ...; rareté relative des termes de la requête ce qui permet de privilégier les informations contenant les termes rares ; etc.
- C'est une information normalement liée au jugement des utilisateurs mais est évaluée par les technologies!
- Elle est importante car elle permet d'ordonner la présentation des résultats: « ranking » (google).

The Process of Preprocessing in IR



+ Thesauri

the expansion of the original query with related term

Quelques définitions (wikipedia) (1)

Stemming: La **racinisation** est le nom donné au procédé qui vise à transformer les flexions en leur radical.

Exemples: cheval, chevaux, chevalier, chevalerie, chevaucher⇒«cheva» (mais pas «cavalier»)

Diacritics: Un signe diacritique, c'est un signe que l'on ajoute à une lettre ou à un groupe de lettres pour le distinguer.

Exemples: mais et maïs, tue et tué

Quelques définitions (wikipedia) (2)

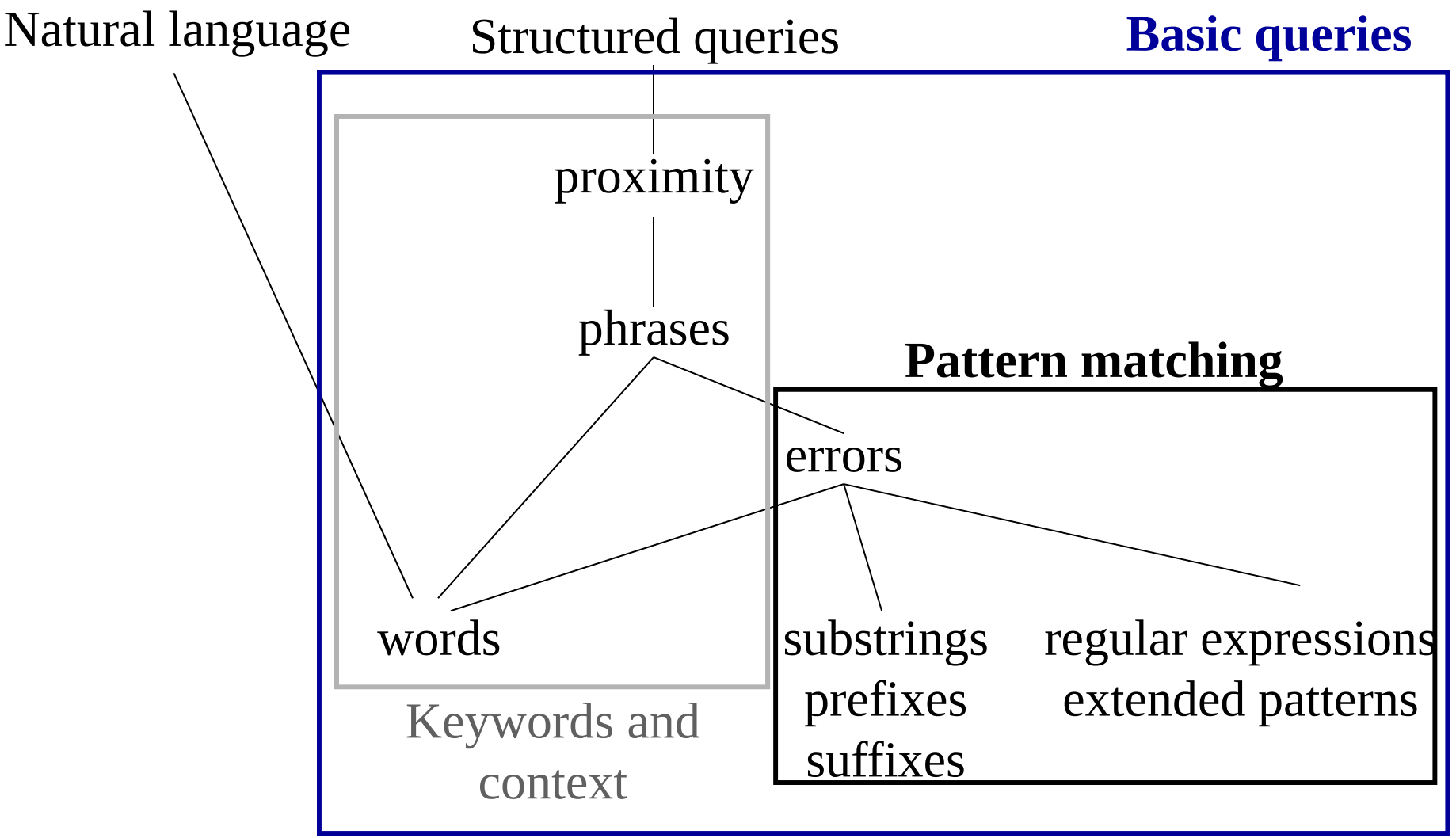
Thesaurus: est une liste organisée de termes représentant les concepts d'un domaine de la connaissance. Permet de passer à des synonymes ou à une généralisation ou a une spécialisation etc..

Exemples: chat et félin ou chat et siamois

Stop words: sont des mots qui sont tellement communs qu'il est inutile de les indexer ou de les utiliser dans une recherche.

Exemples: le, la , et, du, de …

Type de requêtes de l'IR



Plan

1. Introduction: Xquery – IR
2. Syntaxe XFT
3. Annexe: Modèle sémantique: ALLMatches

XFT

XQuery *Full-Text* extends Xquery and Xpath with:

- A new operator: contains text
- Extends FLWOR expression to take into account the pertinence of an answer to a textuel query.
- Extends the XDM model with the concept of ALLmatches

XFT: example

XQuery *Full-Text* is able to use XML elements to restrict or refine queries.

```
//section[ title contains text {"enhancement",
                                "xquery« } same sentence ]
```

Xquery/XPath

IR

Sentences are delimited by end of line markers (.,!,? Etc)

XML FT Search Definition

- *Context expression*: XML elements searched:
  - pre-defined XML elements.

  → Use XPath/XQuery queries.
- *Return expression*: XML fragments returned:
  - pre-defined meaningful XML fragments.

  →Use XPath/XQuery to build answers.
- *Search expression*: FT search conditions:
  - Boolean keyword search, proximity distance, scoping, thesaurus, stop words, stemming.

  → *Need for new language primitives*
- *Score expression*:
  - a scoring function for threshold or top-K queries.

  → *Need for scoring framework*

FTContainsExpr

The fundamental full-text operator

La séquence de nœuds à interroger

La condition que doit vérifier la valeur textuelle des nœuds à interroger

XqueryExpr **contains text** FTSelection

(**without content** UnionExpr)?

Les nœuds à ne pas interroger dont la valeur textuelle doit être ignorée

returns a boolean value:

- true if the full-text query is matched by at least one node in XqueryExpr (search domain),
- false if no match.

contains text: 1er exemple

XqueryExpr **contains text** FTSelection
(without content UnionExpr)?

```
//PLAY[ TITLE contains text "Henry" ]
```

Retourner toutes les pièces dont le titre contient le mot Henri.

without content UnionExpr: example

XqueryExpr contains text FTSelection
(**without content** UnionExpr)?

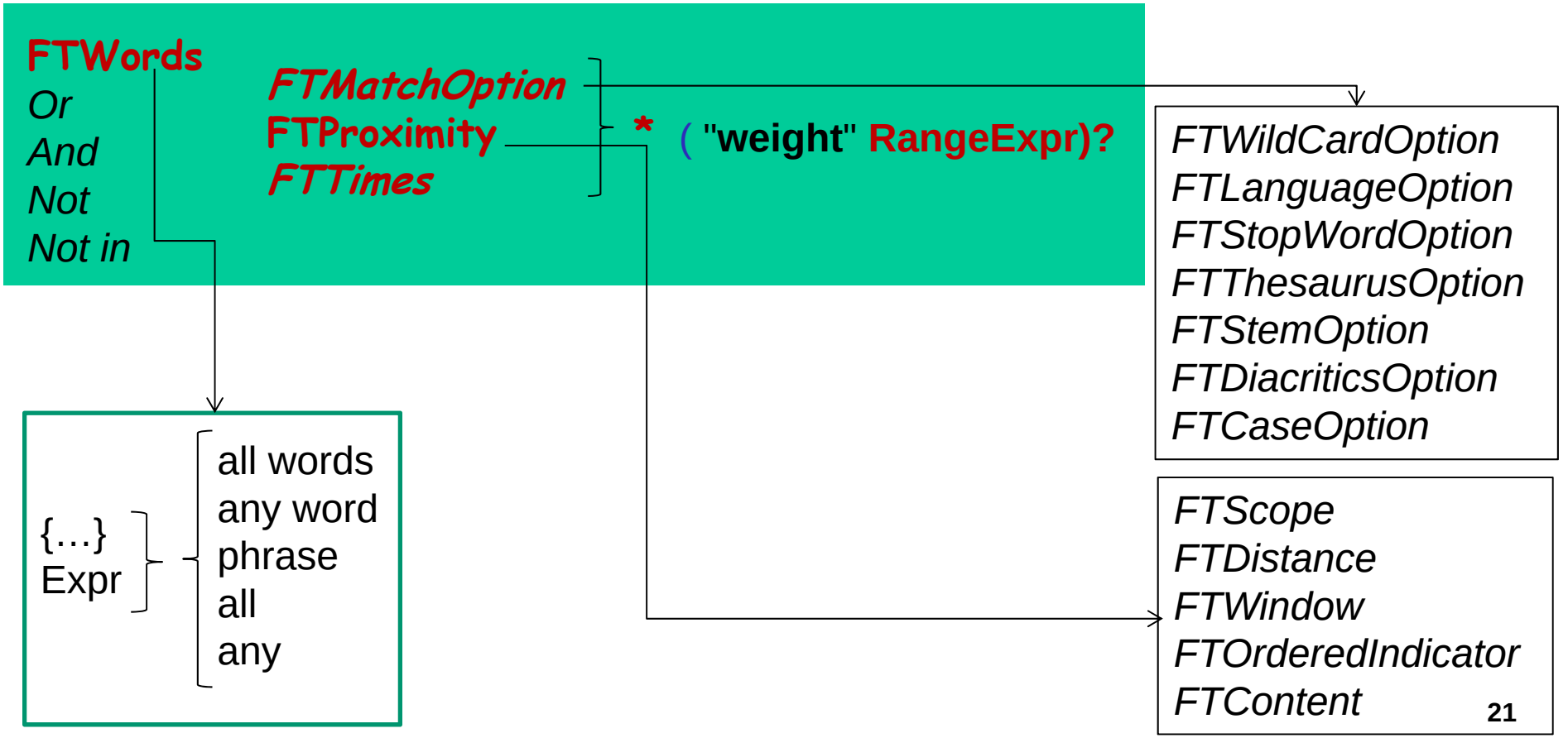
```
/books/book[. contains text "XFT" without
 content .//footnote]/title
```

Retourne tous les titres des livres qui contiennent XFT sans regarder le contenu des éléments footnote.

FTContains: *FTSelection*

XqueryExpr contains text **FTSelection**
(without content UnionExpr)?

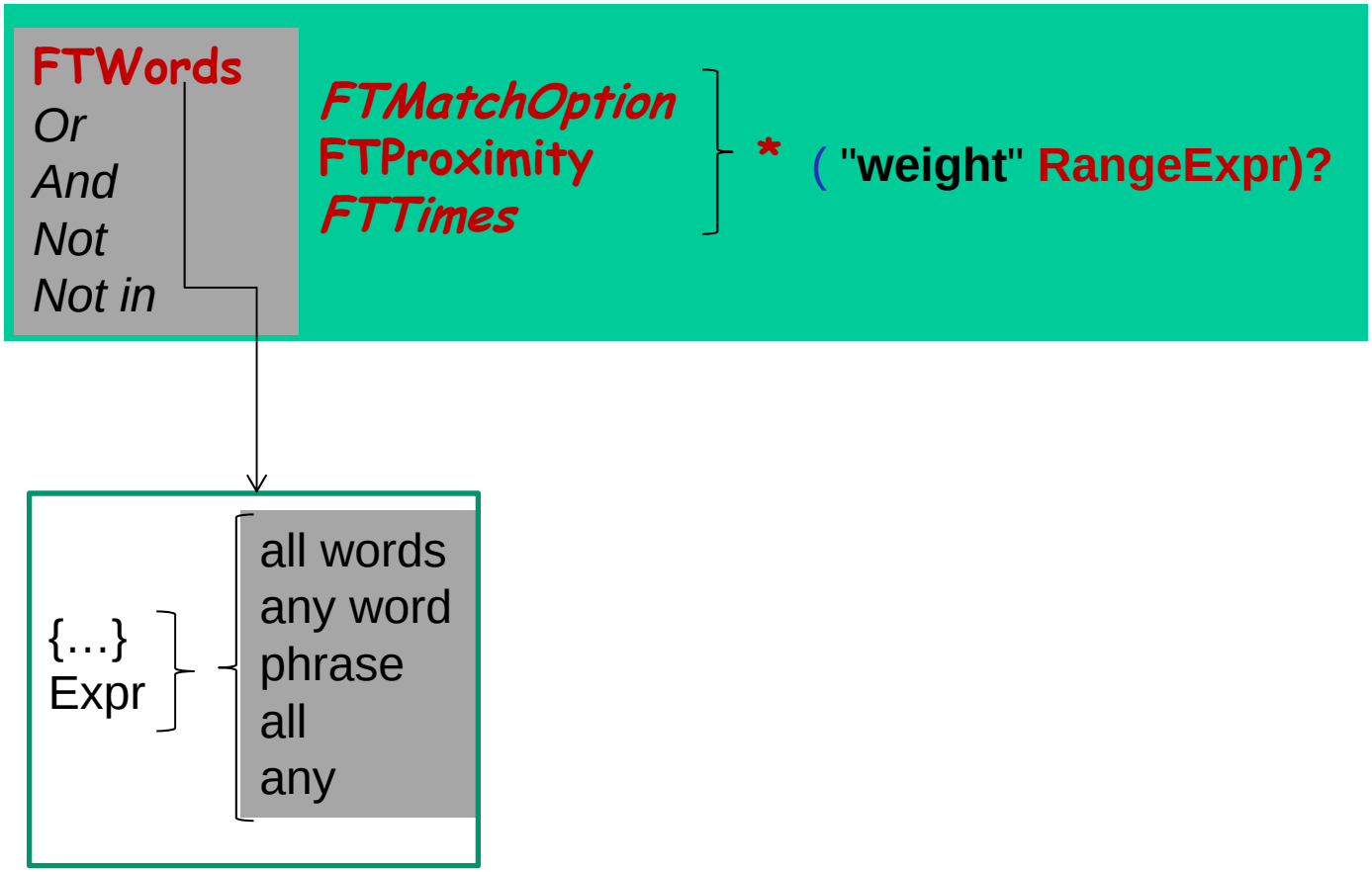
Expr contains text ***FTSelection***



FTContains: *FTSelection-FTWord*

XqueryExpr contains text **FTSelection**
(without content UnionExpr)?

Expr contains text ***FTSelection***



FTWords: examples (1)

Dans le fragment suivant: *"very very big"*

Les FTWords suivants donnent les résultats suivants:

- "very big"
  → *very very big"*
  → 1 match
  Considérée comme une phrase

- {"very", "big"} **all**
  → *very very big"*
  → *very very big"*
  → 2 matchs

- {"very", "big"} **any**
  → *very very big"*
  → *very very big"*
  → *very very big"*
  → 3 matchs

FTWords: examples (2)

*/books/book/title contains text "web" **ftand** "site" **ftand** "usability"*

Retourne vrai si le titre contient les 3 mots.

/books/book contains text ("web" ftand "site" ordered) ftand ("usability" ftor "testing")

Retourne vrai si le livre contient le mot *"web"* et *"site"* dans cet ordre et un des 2 mots *"usability"* ou *"testing"*.

FTContains: *FTSelection-FTTimes*

XqueryExpr contains text **FTSelection**
(without content UnionExpr)?

Expr contains text FTSelection

FTWords *FTMatchOption* / **FTProximity** / *FTTimes* } * ("**weight**" **RangeExpr)?**

occurs FTRange times

- exactly…
- at least…
- at most…
- from … to …

FTTimes : examples

*//book[. contains text "usability" **occurs at least 2 times**]*

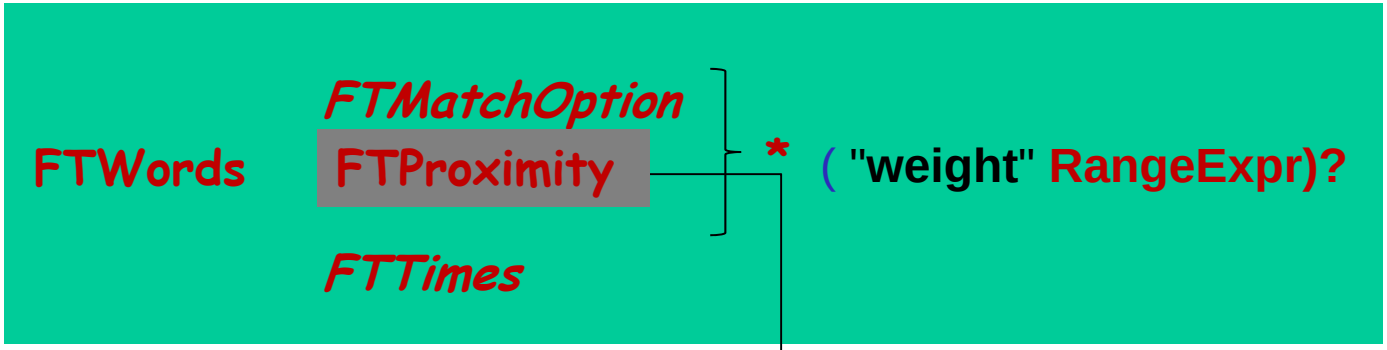
Retourne tous les livres qui contiennent usability avec au moins deux occurrences de ce mot.

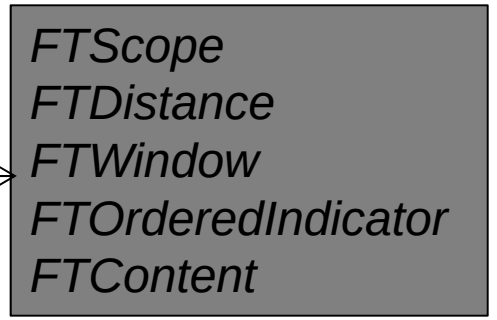
*//book[@number="1" and title contains text {"usability", "testing"} any **occurs at most 2 times**]*

Retourne tous les livres qui ont un attribut *@number="1"* et dont le titre contient au plus les tokens *usability"* ou *"testing"* au nombre de 2.

FTContains: *FTSelection-FTProximity*

Expr contains text ***FTSelection***





FTProximity: examples (1)

/books/book/title contains text "web" ftand "site" ftand "usability" ***window 5 words***

Retourne vrai si le titre contient les 3 mots à l'intérieur d'une fenêtre de 5 mots.

/books/book[@number="1" and . contains text "efficient" ftand ftnot "and" ***window 2 words****]*

Retourne les livres qui ont un attribut *@number="1"* et qui contiennent le mot *"efficient"* et pas à coté su mot *"and"* (fenêtre de 2 mots).

FTProximity: examples (2)

```
/books/book contains text ("completion" ftand "errors"
distance at least 11 words)
```

Retourne vrai si le livre contient le mot *completion"* et le mot *"errors"* avec une distance de 11 tokens entre eux.

```
/books/book[ . contains text
(  (("richard" ftand "nixon") distance at most 2 words)
     ftand
    (("george" ftand "bush") distance at most 2 words)
  distance at least 20 words) ]
```

Retourne les livres qui contiennent par exemple *"Richard M. Nixon"* et qui contiennent le mot *"George W. Bush"* distant d'au moins 20 mots

FTProximity: examples (3)

//book contains text "usability" ftand "testing" ***same sentence***

Retourne vrai si le livre contient le mot *"usability"* et le mot *"testing"* dans une même phrase.

<introduction>
... The usability of a Web site is how well the site supports the user in achieving specified goals. ... Expert reviews and usability testing are methods of identifying problems in layout, terminology, and navigation. ...
</introduction>

Ce texte satisfait les deux conditions suivantes *same sentence* ou *different sentence*

FTContains: *FTSelection-FTMatchOption*

XqueryExpr contains text **FTSelection**
(without content UnionExpr)?

Expr contains text FTSelection

FTWords *FTMatchOption* / FTProximity / *FTTimes* } * ("weight" RangeExpr)?

using FTMatchOptions

FTWildCardOption
FTLanguageOption
FTStopWordOption
FTThesaurusOption
FTStemOption
FTDiacriticsOption
FTCaseOption

FTMatchOptions

- Stemming/linguistic:
  - recherche du mot exact ou ses déclinaisons
- Character case variations:
  - Insensible ou pas à la casse des mots
- Diatrics (naïve vs naive)
  - Insensible ou pas
- Wildcards character
  - 1, optionnel, 0 ou plus, entre n et m caractères
- Thesaurus (chat vs félin)
  - On peut spécifier le thesaurus, à quels type de relationt et à quelle niveau on veut étendre la recherche (synonymes, termes associés, termes spécifiques, termes équivalents, termes génériques…)
- StopWords (le, la, et …)
- Language: langue utilisée dans le document ou la requête.

FTWildcardOption

Un "wildcard" est composé d'un indicateur le point . éventuellement suivi ou pas d'un "qualifier" de:

- Si point seul: joker pour un caractère.
- "?": joker pour 0 ou 1 caractère.
- "*": joker pour 0 ou plusieurs caractères.
- "+": joker pour 1 ou plusieurs caractères.
- Une expression régulière de type {[0-9]+,[0-9]+}: joker pour un nombre de caractères compris entre le premier et le deuxième nombre du couple donné.

FTWildcardOption: example

//book[@number="1"]/p contains text "w.ll" **using wildcards**

true, if the p element contains "well".

//book[@number="1"]/p contains text "w.ll" using **no wildcards**

Note that, without wildcards, the sample tokenization will treat the . in "w.ll" as punctuation, thus producing "w" and "ll" as separate tokens.

false, because the p element does not contain the phrase "w ll".

FTStemOption et FTStopWordOption: Examples

/books/book[@number="1"]/title contains text "improve" **using stemming**

true, because the title of the specified book contains "improving" which has the same stem as "improve"

Stop words sont des tokens dans la **query** qui remplace n'importe quel token dans le texte recherché.

/books/book[@number="1"]//p contains text "propagating of errors" **using stop words ("a", "the", "of")**

true, because the document contains the phrase "propagating few errors"

XFT: Score

Le score d'un résultat d'une requête exprime sa pertinence par rapport aux conditions de recherche.

XFT a étendu les requêtes XQuery et XPath en ajoutant des variables de score optionnels au for et let des expressions FLWOR

In any order {
for $v [score $s]? [at $i]? in Expr
let …
where …
order by …
return …

Les valeurs Score sont xs:double [0; 1]; Plus les valeurs sont grandes plus la pertinence est élevée.

Score: exemple (1)

```
for $b score $s in
  /pub/book[. contains text "Usability" ftand "testing"]
order by $s descending
return <result score={$s}> $b</result>
```

Retourne les éléments livre qui contiennent les mots "Usability" et "testing" avec leur score. Ils sont rangés dans l'ordre du score le plus haut au plus bas.

Score: exemple (2)

```
for $b score $s in
        /pub/book[.//chapter/title contains text "testing"]
let score $s := $b/content contains text "web site" ftand
                                                "usability"
order by $s descending
return <result score={$s}> $b</result>
```

Retourne les éléments livre qui contiennent des titres de chapitres qui contiennent le mot "testing", avec leur score. Ces scores reflètent cependant si le contenu du livre contient les mots "web site" et "usability".

FTContains: *FTSelection- Using Weights*

Expr contains text ***FTSelection***

FTWords ***FTMatchOption*** **FTProximity** ***FTTimes*** ***** **("weight" RangeExpr)?**

Les poids (RangeExpr) peuvent être utilisées à l'intérieur d'une FTContains pour influencer le calcul du score final. Un poids varie de 0.0 .. 1000.0.

Permet d'indiquer l'importance relative des tokens de recherche entre eux.

Exemple d'utilisation

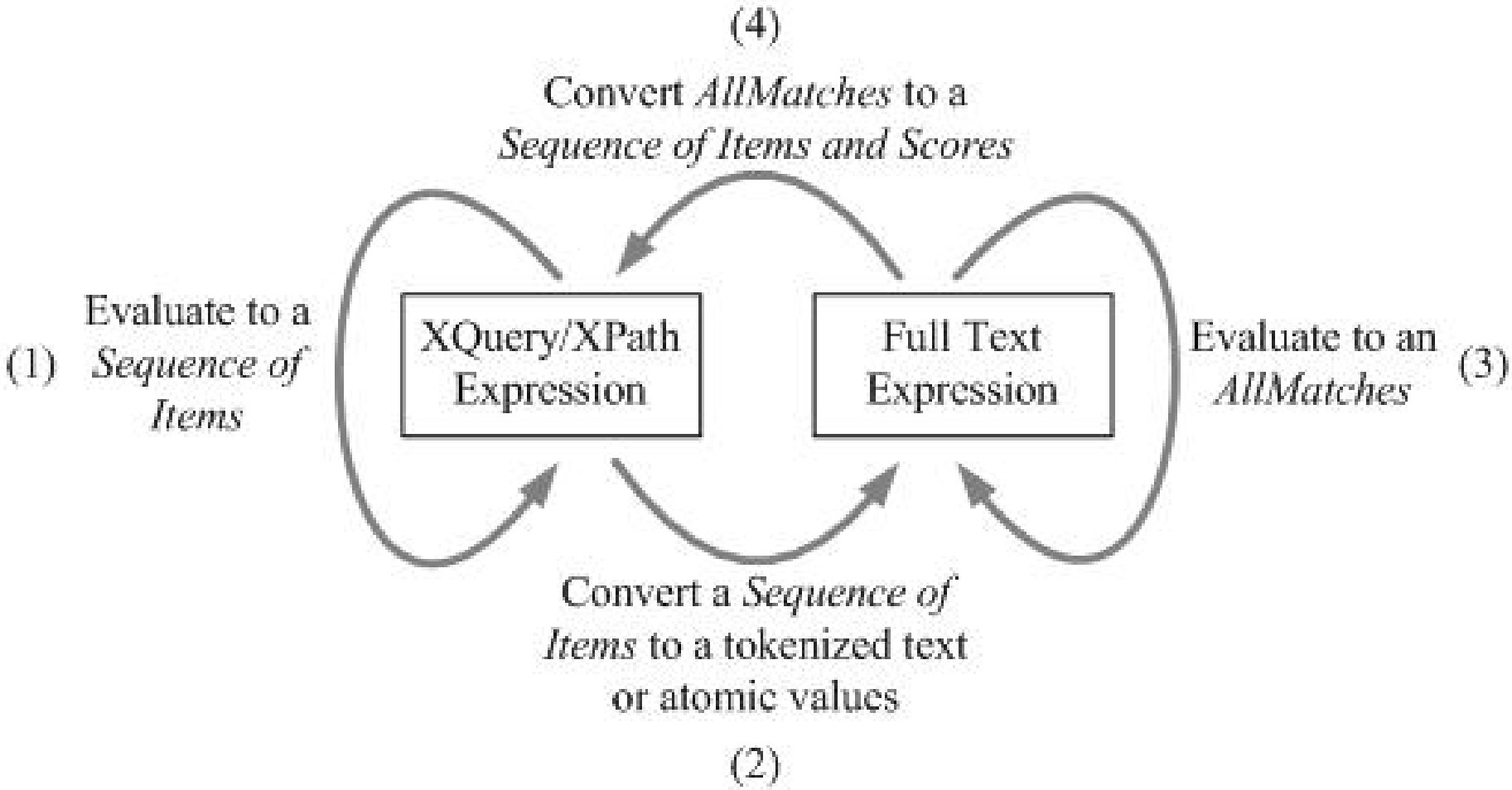
```
for $b in /books/book
let score $s := $b/content contains text
                    ("web site" weight {0.5}) ftand
                    ("usability" weight {2})
return <result score="{$s}">{$b}</result>
```

Scoring is completely implementation-defined: le comportement est défini par le compilateur, et non par la norme du langage. Donc comportement différents suivants les compilateurs. Voir avoir un comportement indéfini.

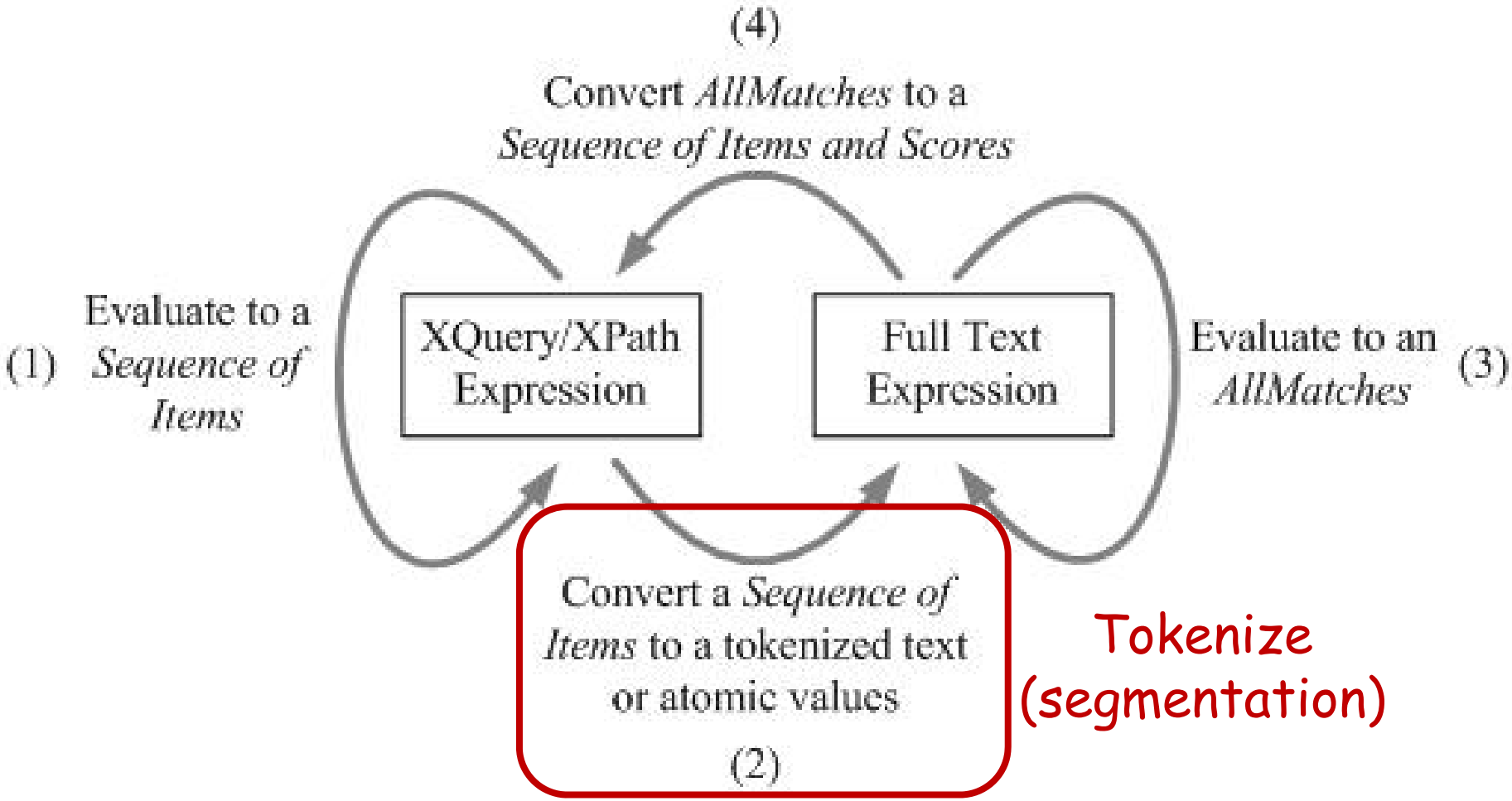
Plan

1. Introduction: Xquery – IR
2. Syntaxe XFT
3. Annexe: Modèle sémantique: ALLMatches

FT semantic: interaction Xpath/Xquery - XFT



FT semantic: interaction Xpath/Xquery - XFT



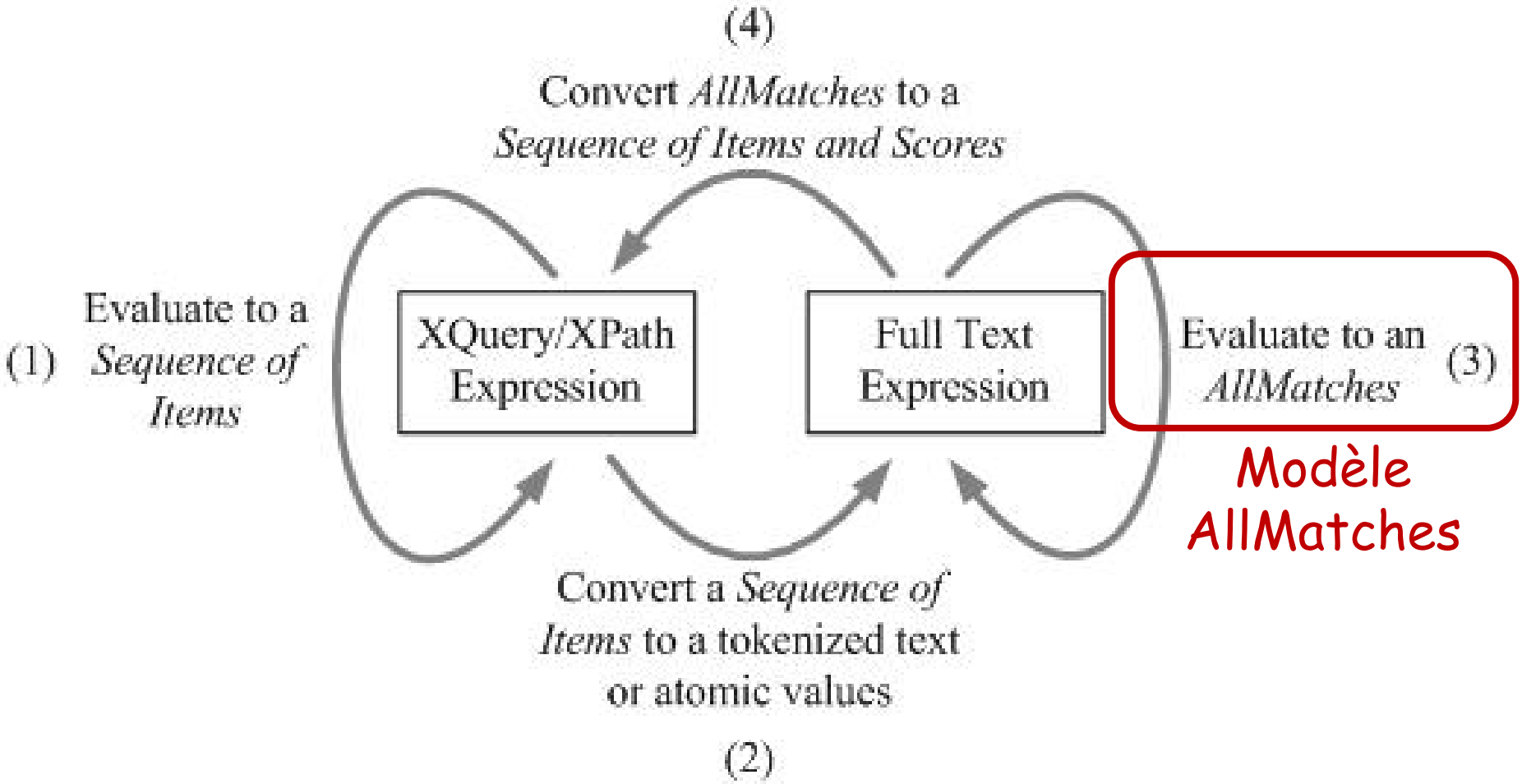
Exemple: segmentation

```
<book id="1000">
  <author> Gerald Bruce and Elina F. Rose</author>
  <content>
        <p> Here we present the usability of software which
              measures how well the software provides
              support for quickly achieving specified
              goals. </p>
        <p>But the users must not only
              be well-served, but must
              feel well-served.</p>
  </content>
</book>
```

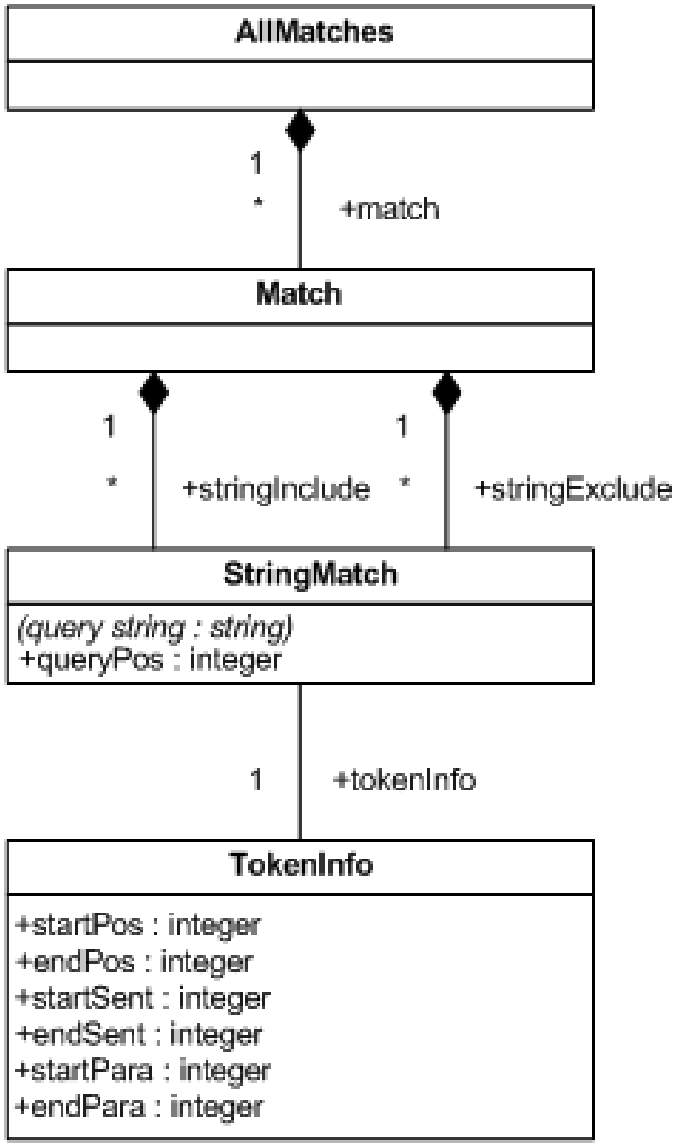
Exemple: segmentation

<book id="1000">
  <author> Gerald(1) Bruce(2) and(3) Elina(4) F.(5) Rose(6)</author>
  <content>
     Here(7) we(8) present(9) the(10) usability(11) of(12)
      software(13) which(14) measures(15) how(16)
      well(17) the(18) software(19) provides(20)
      support(21) for(22) quickly(23) achieving(24)
      specified(25) goals(26).
    But(27) the(28) users(29) ...
  </content>
</book>

FT semantic: interaction Xpath/Xquery - XFT



Modèle AllMatches



- An **AllMatches** describes the possible results of an FTSelection. Each **Match** describes one result to the FTSelection.

- The **AllMatches** is a disjunction of **Match**es. Each **Match** is a conjunction of **StringInclude**s, and **StringExclude**s.

Exemple

```
$doc contains text ('usability' using steming
                    ftand
                    'Rose')
                    window at most 10
```

Exemple: segmentation

<book id="1000">

<author> Gerald(1) Bruce(2) and(3) Elina(4) F.(5) **Rose(6)**</author>

<content>

 Here(7) we(8) present(9) the(10) **usability(11)** of(12) software(13) which(14) measures(15) how(16) well(17) the(18) software(19) provides(20) support(21) for(22) quickly(23) achieving(24) specified(25) goals(26).

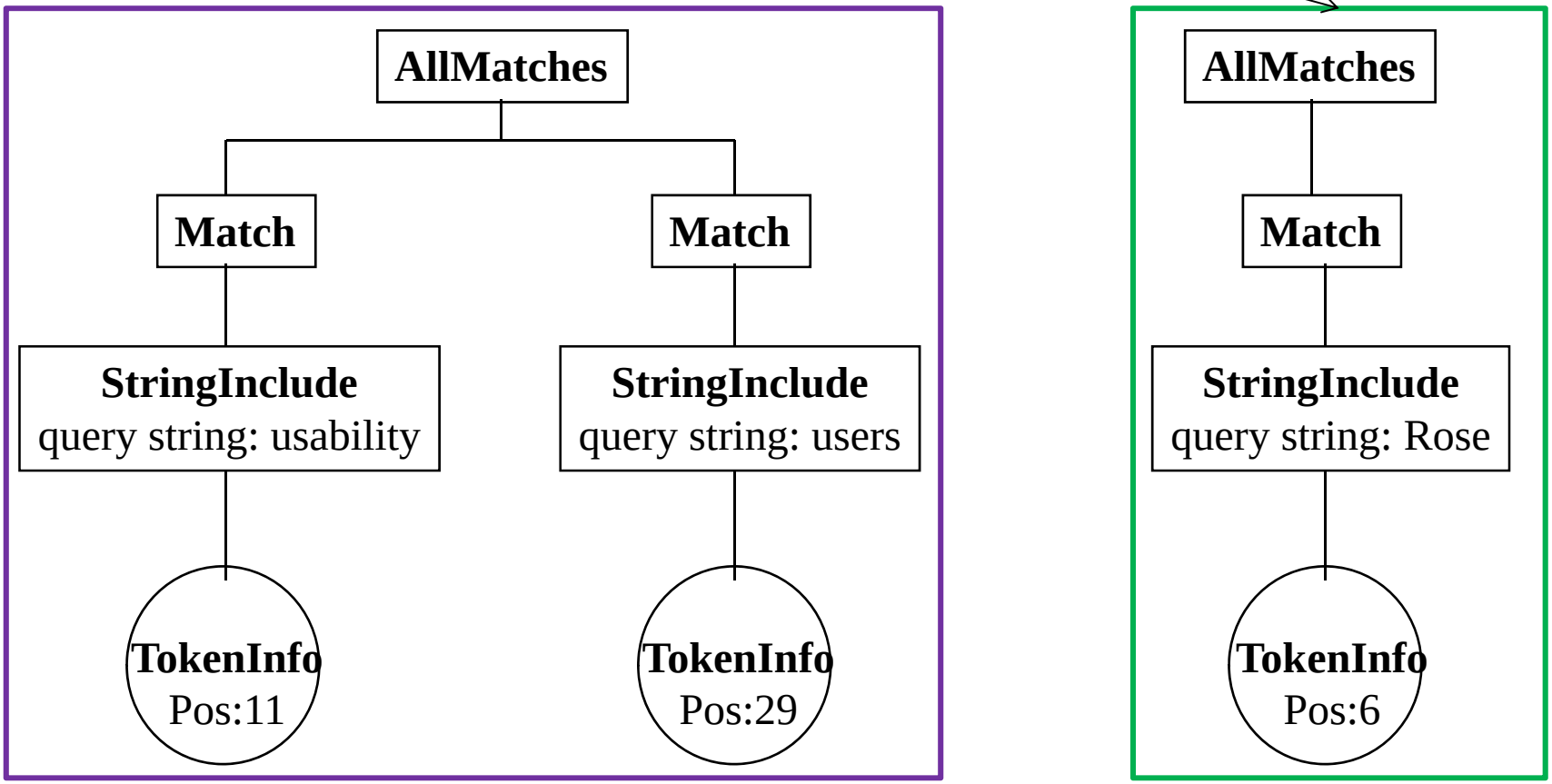
But(27) the(28) **users(29)** ...

</content>

</book>

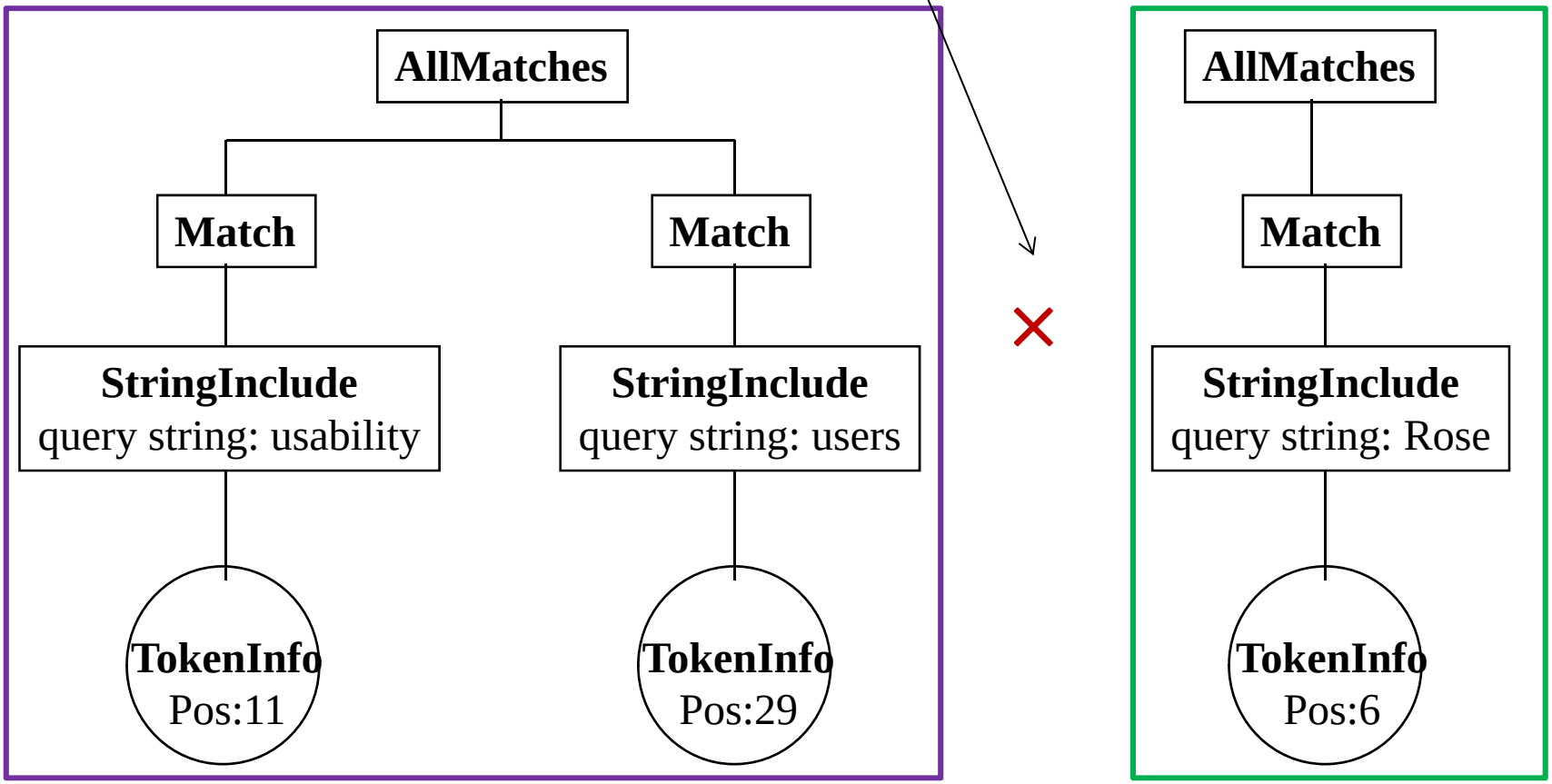
Exemple - suite

```
$doc contains text ('usability' using steming
                    ftand
                    'Rose')
                    window at most 10
```



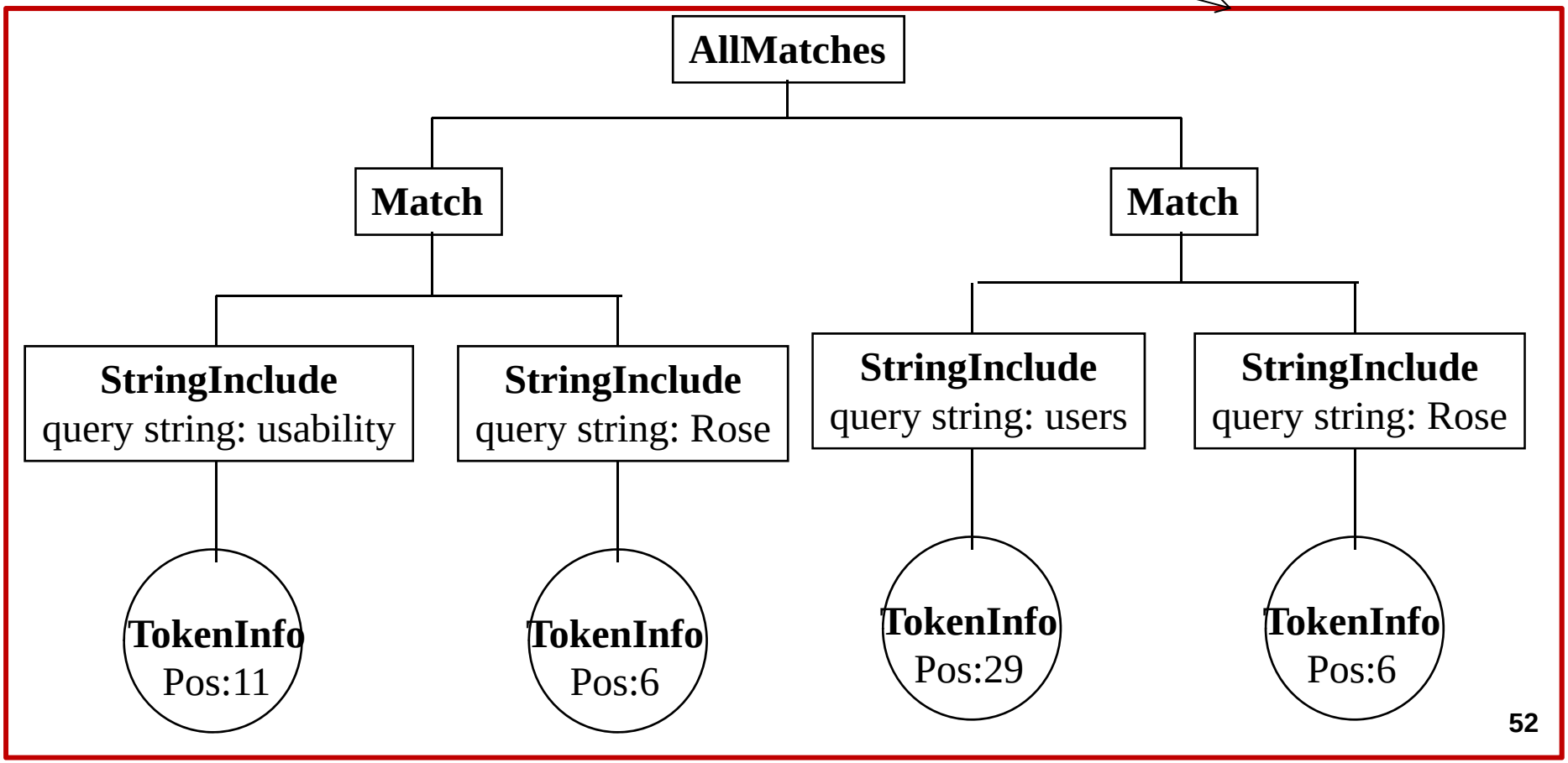
Exemple - suite

```
$doc contains text ('usability' using steming
                    ftand
                    'Rose')
                    window at most 10
```



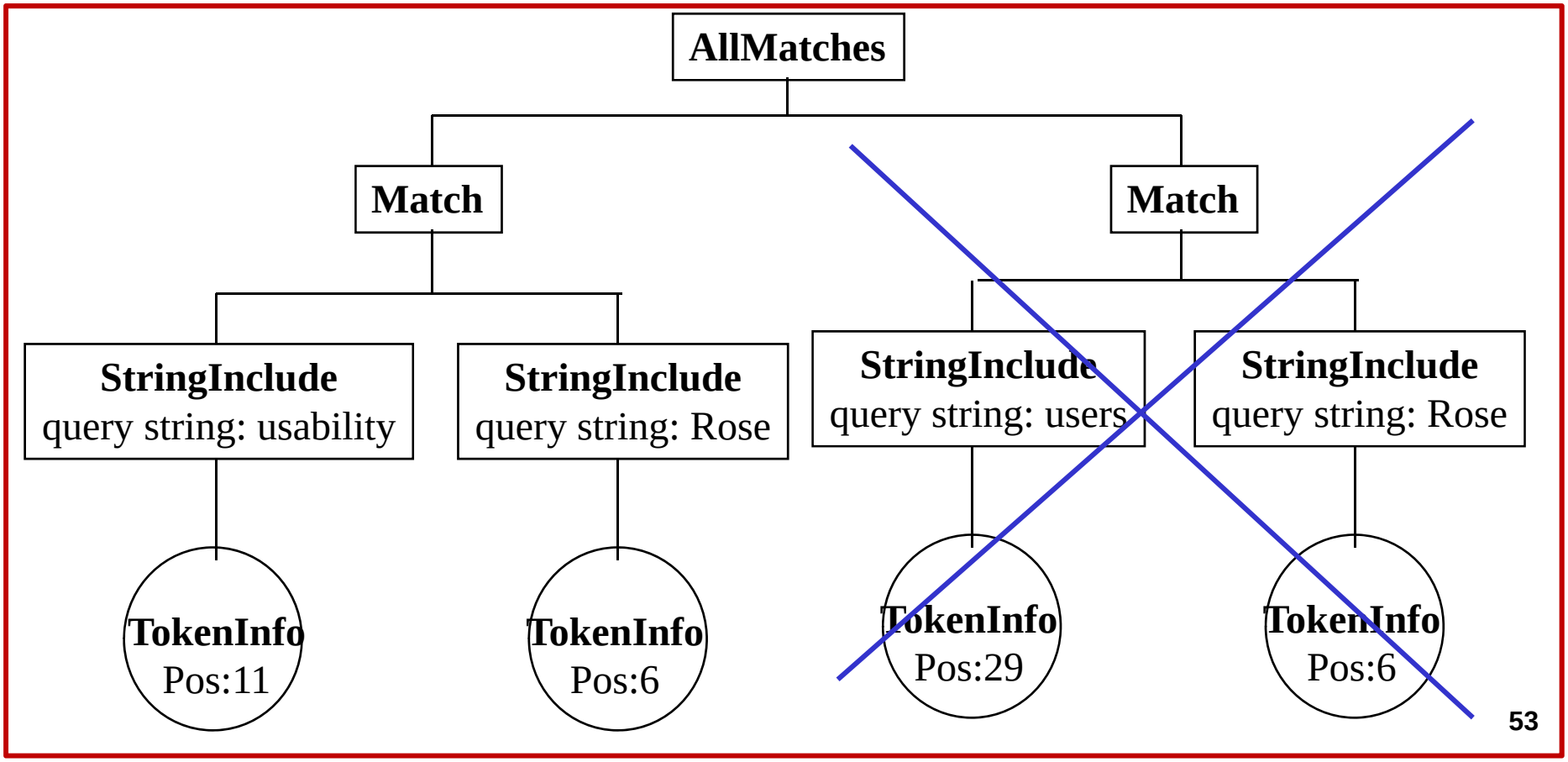
Exemple - suite

```
$doc contains text ('usability' using steming
                    ftand
                    'Rose')
                    window at most 10
```



Exemple - suite

```
$doc contains text ('usability' using steming
                    ftand
                    'Rose')
                    window at most 10
```



Exemple: segmentation

```
$doc contains text ('usability' using steming
                    ftand
                    'Rose')
                    window at most 10
```

<book id="1000">

<author> Gerald(1) Bruce(2) and(3) Elina(4) F.(5) **Rose(6)**</author>

<content>

 Here(7) we(8) present(9) the(10) **usability(11)** of(12)
software(13) which(14) measures(15) how(16)
well(17) the(18) software(19) provides(20)
support(21) for(22) quickly(23) achieving(24)
specified(25) goals(26).

But(27) the(28) **users(29)** ...

</content>

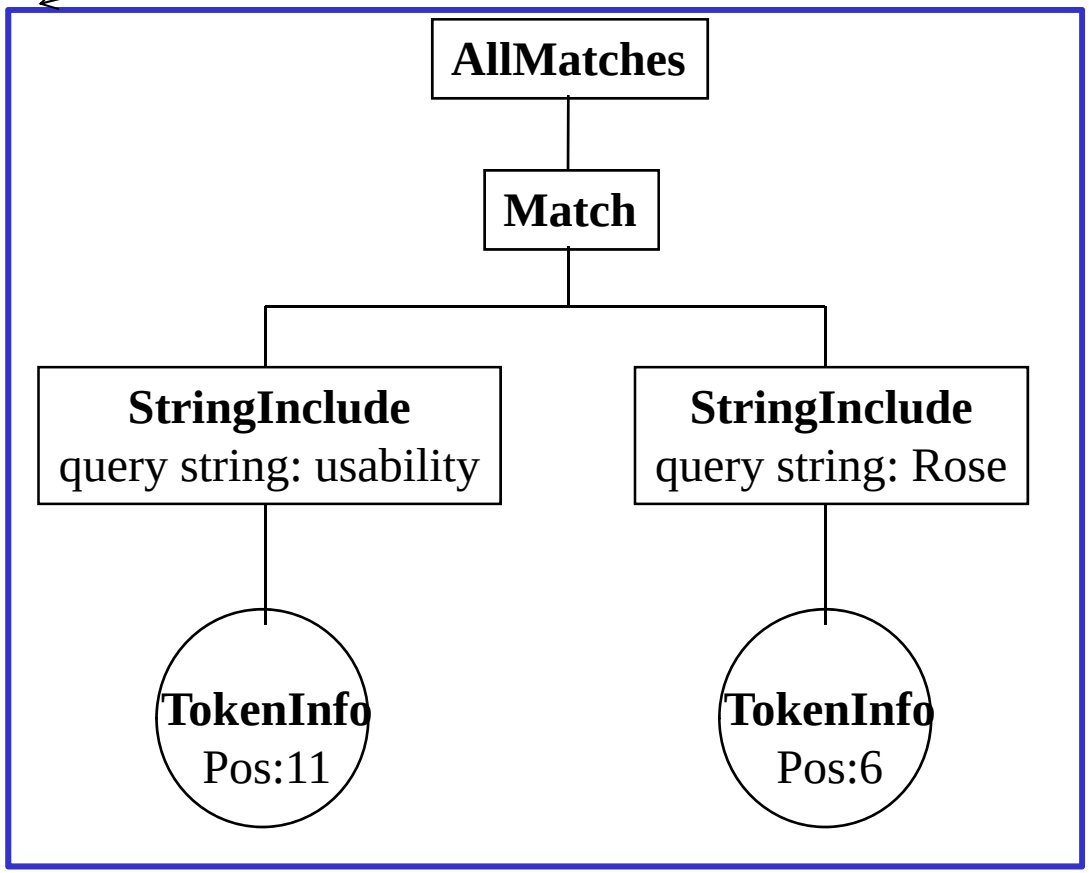
</book>

Distance > 10

Distance < 10

Exemple - suite

```
$doc contains text ('usability' using steming
                    ftand
                    'Rose')
                    window at most 10
```



Synthèse sur XQuery FT

- Prise en charge de "rechercher" de typed full-text dans le contexte de Xquery
- Combiner la recherche structurée avec du IR
- Le calcul de pertinence est encore une question ouverte