Gestión de Información en la Web
Máster en Ingeniería Informática
Parte II: Recuperación de información y Sistemas de recomendación

Seminario 4: Recuperación XML

Juan Manuel Fernández Luna
Dpto. Ciencias de la Computación e Inteligencia Artificial
jmfluna@decsai.ugr.es

Motivación

Consulta:

Efectos del bálsamo de Fierabrás

Resultado:

Don Quijote de la Mancha


EL INGENIOSO
HIDALGO DON QVI-
XOTE DE LA MANCHA
Compuesto por Miguel de Cervantes Saauedra.
DIRIGIDO AL DVQVE DE BEIAR,
Marques de Gibraleon, Conde de Barcelona, y Bana-
res, Vizconde de la Puebla de Alcozer, Señor de
las villas de Capilla, Curiel, y
Burgillos.
Ano, 1605.
Con priuilegio de Castilla, Aragon, y Portugal.
EN MADRID, Por Iuan de la Cuesta.
Vendese en casa de Francisco de Robles, librero del Rey nro señor.

Motivación

Consulta:

Efectos del bálsamo de Fierabrás

Resultado:

Don Quijote de la Mancha


EL INGENIOSO
HIDALGO DON QVI-
XOTE DE LA MANCHA
Compuesto por Miguel de Cervantes Saauedra.
DIRIGIDO AL DVQVE DE BEIAR,
Marques de Gibraleon, Conde de Barcelona, y Bana-
res, Vizconde de la Puebla de Alcozer, Señor de
las villas de Capilla, Curiel, y
Burgillos.
Ano, 1605.
Con priuilegio de Castilla, Aragon, y Portugal.
EN MADRID, Por Iuan de la Cuesta.
Vendese en casa de Francisco de Robles, librero del Rey nro señor.

Capítulo 10:
De los graciosos razonamientos que entre D. Quijote y Sancho Panza su escudero

Motivación

Resultado:

Don Quijote de la Mancha

Capítulo 10: De los graciosos razonamientos que pasaron entre D. Quijote y Sancho Panza su escudero.


EL INGENIOSO
HIDALGO DON QVI-
XOTE DE LA MANCHA
Compuesto por Miguel de Cervantes Saauedra.
DIRIGIDO AL DVQVE DE BEIAR,
Marques de Gibraleon, Conde de Barcelona, y Banares, Vizconde de la Puebla de Alcozer, Señor de las villas de Capilla, Curiel, y Burgillos.
Año. 1605.
Con priuilegio de Castilla, Aragon, y Portugal.
EN MADRID, Por Iuan de la Cuesta.
Vendese en casa de Francisco de Robles, librero del Rey nro señor.

Todo esto fuera bien escusado, respondió Don Quijote, si a mí se me acordara de hacer una redoma del bálsamo de Fierabrás, que con sólo una gota se ahorraran tiempo y medicinas. ¿Qué redoma y qué bálsamo es ese? dijo Sancho Panza. De un bálsamo, respondió Don Quijote, de quien tengo la receta en la memoria, con el cual no hay que tener temor a la muerte, ni hay que pensar morir de ferida alguna; y así, cuando yo le haga y te le dé, no tienes más que hacer sino que cuando vieres que en alguna batalla me han partido por medio del cuerpo, como muchas veces suele acontecer, bonitamente la parte del cuerpo que hubiere caído en el suelo, y con mucha sutileza, antes que la sangre se hiele, la pondrás sobre la otra mitad que quedare en la silla, advirtiendo de encajallo [illegible] justo. Luego me darás a beber solos dos tragos del bálsamo que he dicho, y verásme quedar más sano q[illegible]. Si eso hay, dijo Panza, yo renuncio desde aquí el gobierno de la prometida ínsula, y no quiero otra cosa [illegible] muchos y buenos servicios, sino que vuestra merced me dé la receta de ese estremado licor, que para mí [illegible] á la onza donde quiera más de dos reales, y no he menester yo más para pasar esta vida honrada y desca[illegible]ro es de saber ahora si tiene mucha costa el hacella. Con menos de tres reales se pueden hacer tres azumb[illegible]on Quijote. ¡Pecador de mí! replicó Sancho. ¿Pues a qué aguarda vuestra merced a hacelle y a enseñármele[illegible]respondió Don Quijote, que mayores secretos pienso enseñarte, y mayores mercedes hacerte; y por ahora curémonos, que la oreja me duele más de lo que yo quisiera.

Motivación

Resultado:

Entrada de la Wikipedia: Fierabrás.

Sección: El bálsamo de Fierabrás.

El bálsamo de Fierabrás es una poción mágica capaz de curar todas las dolencias del cuerpo humano que forma parte de las leyendas del ciclo carolingio. Según la leyenda épica, cuando el rey Balán y su hijo Fierabrás conquistaron Roma, robaron en dos barriles los restos del bálsamo con que fue embalsamado el cuerpo de Jesucristo, que tenía el poder de curar las heridas a quien lo bebía.

En el capítulo X del primer volumen de Don Quijote de la Mancha de Miguel de Cervantes, después de una de sus numerosas palizas, Don Quijote menciona a Sancho Panza que él conoce la receta del bálsamo. En el capítulo XVII, Don Quijote instruye a Sancho que los ingredientes son aceite, vino, sal y romero. El caballero los hierve y bendice con ochenta padrenuestros, ochenta avemarías, ochenta salves y ochenta credos. Al beberlo, Don Quijote padece vómitos y sudores, y se siente curado después de dormir. Sin embargo, para Sancho tiene un efecto laxante.

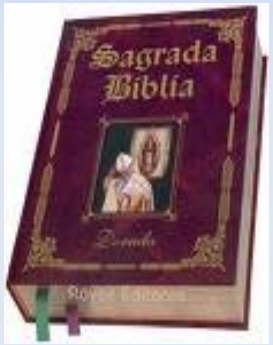
Motivación

Consulta:

Resurrección de Lázaro

Resultado:

Biblia



Motivación

Consulta:

Resurrección de Lázaro

Resultado:

Evangelio de San Juan

Motivación

Consulta:

Resurrección de Lázaro

Resultado:

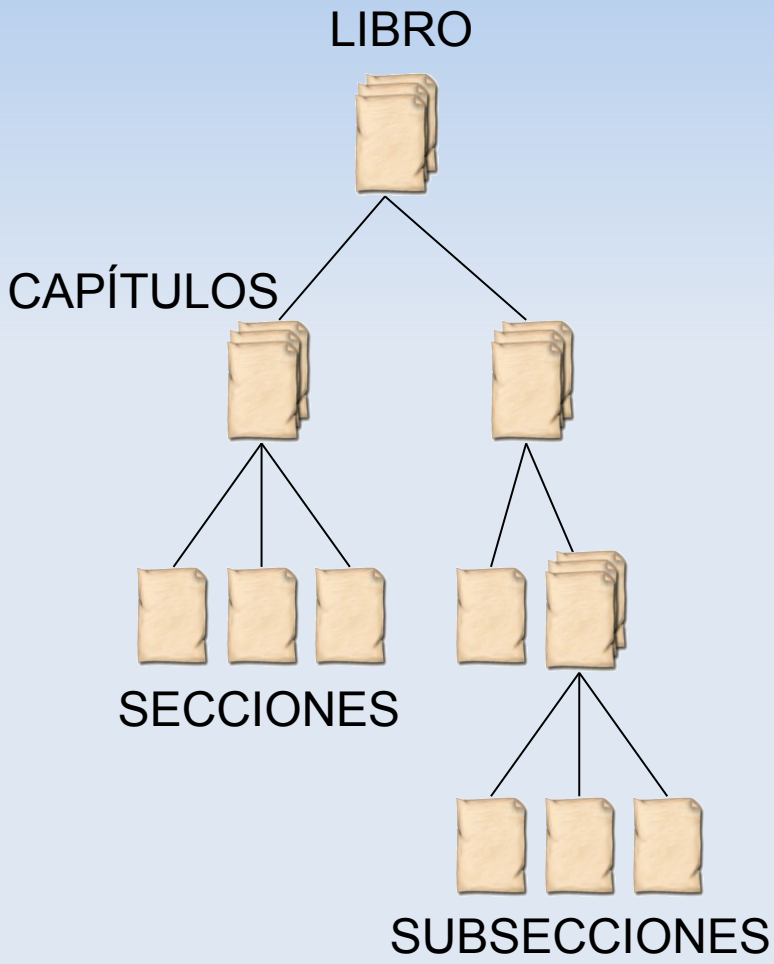
Evangelio de San Juan, Capítulo 11, versículos 41 a 44

Motivación

Los documentos suelen tener una estructura lógica.

¿Por qué no explotarla para ofrecer al usuario material relevante más concreto?

Ahrroraríamos mucho tiempo.



RECUPERACIÓN DE DOCUMENTOS ESTRUCTURADOS

Índice

1. Introducción.
2. eXtensible Markup Language.
3. Recuperación de documentos estructurados.
4. Consulta.
5. Indexación.
6. Recuperación.
7. Presentación de resultados.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

Índice

1. **Introducción.**
2. eXtensible Markup Language.
3. Recuperación de documentos estructurados.
4. Consulta.
5. Indexación.
6. Recuperación.
7. Presentación de resultados.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

1. Introducción

Documentos = contenido + estructura

- **Contenido**:
  - texto, secuencia de palabras formando frases.
- **Estructura**:
  - forma en que se organiza lógicamente.
- P.E., artículo científico: título, autores, resumen, secciones, subsecciones, párrafos.
- Forma de representarlos: mediante un lenguaje de marcado. El más usado XML.

1. Introducción

- El marcado XML permite:
  - **acceder focalizadamente** a los documentos,
  - haciendo que los S.R. XML puedan **devolver sus componentes**.
  - **Alivia el esfuerzo** del usuario para acceder al material relevante.
  - El S.R. XML proporciona las **partes más adecuadas**.

1. Introducción

- Se están desarrollando numerosa herramientas para facilitar ese acceso a documentos estructurados.
- Campos de investigación (mayormente desde 2002, INEX):

1) Consulta.
2) Indexación.
3) Recuperación.
4) Presentación.
5) Evaluación.

Índice

1. Introducción.
2. **eXtensible Markup Language.**
3. Recuperación de documentos estructurados.
4. Consulta.
5. Indexación.
6. Recuperación.
7. Presentación de resultados.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

2. eXtensible Markup Language

- Versión simplificada de **SGML**.
- **SGML = Standard Generalized Markup Language**.
- Metalenguaje usado para definir lenguajes de marcado con objeto de ofrecer una interpretación de datos (textos) independientemente de dispositivos y sistemas.

2. eXtensible Markup Language

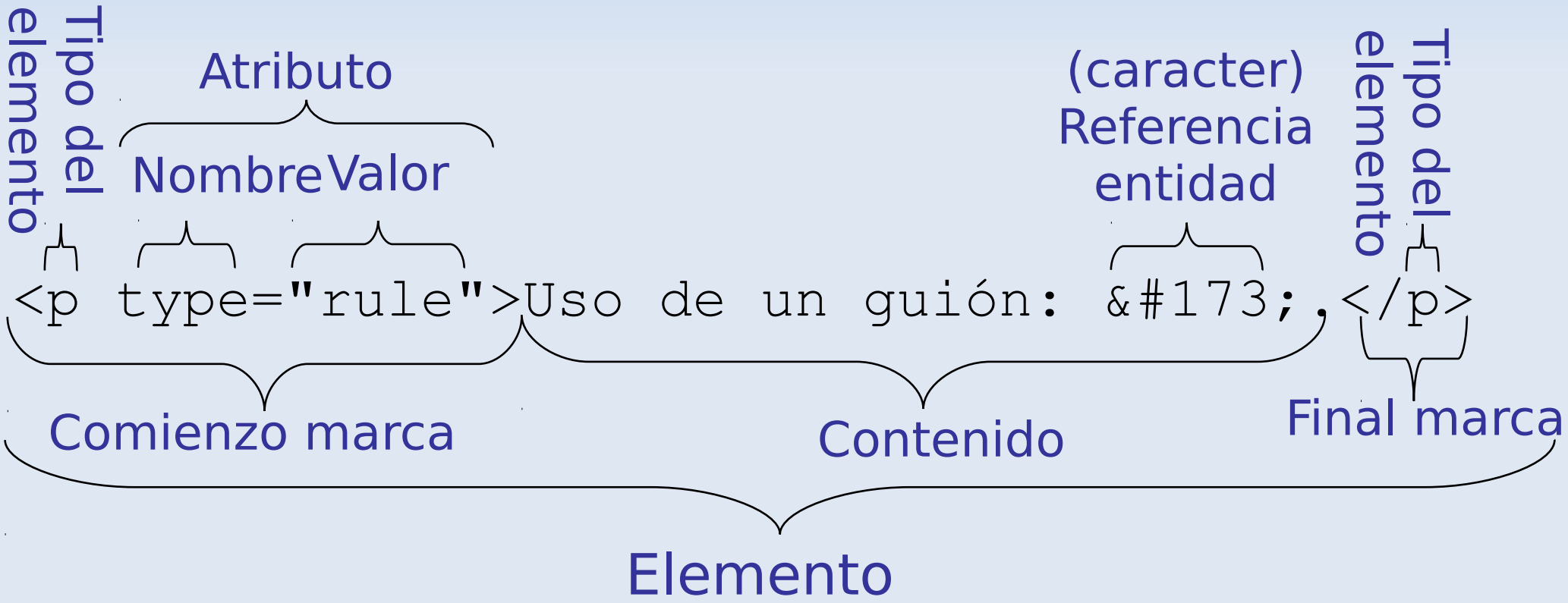
- **SGML** es más **flexible** y **potente** que XML, pero es más **complejo** y **costoso** de implementar.
- **XML** mantiene las **ventajas** de SGML sobre **extensibilidad, estructura y validación**, pero está diseñado para ser **más simple y fácil de aprender y usar**.

2. eXtensible Markup Language

- XML puede ser procesado automáticamente e intercambiado entre diversos hardware, sistemas operativos y aplicaciones.
- XML es extensible = las marcas se pueden definir para aplicaciones específicas.
- XML se centra en la estructura lógica de un documento y no en su diseño.
- **XML se centra en describir datos.**
- **HTML se centra en mostrar datos.**

2. eXtensible Markup Language

Componentes de XML



2. eXtensible Markup Language

Componentes de XML

- **Elementos**:

Componentes lógicos de un documento.

<elemento>.... </elemento>

- Puede contener texto, otros elementos o estar vacíos.
- Pueden tener elementos padre e hijos.

2. eXtensible Markup Language

Componentes de XML (ejemplo de fichero)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<breakfast_menu>
    <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>two of our famous Belgian Waffles with plenty of real maple syrup</description>
    <calories>650</calories>
</food>
<food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>light Belgian waffles covered with strawberries and whipped cream</description>
    <calories>900</calories>
</food>
</breakfast_menu>
```

(Ejemplo de w3schools.com)

2. eXtensible Markup Language

Componentes de XML

- **Atributos**:
  - Se utilizan para asociar pares (nombre, valor) a los elementos. Suelen ofrecer información que no son parte de los datos (o sí :-)).
  - Reglilla: metadatos (datos sobre los datos) en atributos, datos en sí mismos, elementos.

<elemento atributo=”valor”>.... </elemento>

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

2. eXtensible Markup Language

Algunas normas de sintaxis

- Todos los elementos XML tienen que tener una marca de cierre, salvo los que expresamente se dejen vacíos que tendrán <elemento/>.
- Las marcas XML y los atributos son sensibles a las mayúsculas/minúsculas.
- Todos los elementos XML deben estar apropiadamente anidados.
- Todos los documentos XML deben tener un elemento raíz.
- Todos los atributos deben estar entre comillas dobles.
- En XML, se mantiene el espacio en blanco.
- En XML, un fin de linea se almacena como un carácter LF.
- Comentarios XML: <!-- Esto es un comentario -->

2. eXtensible Markup Language

Validación XML

- **Documento bien formado**: aquel que es correcto desde un punto de vista sintáctico.
- **Documento válido**: aquel bien formado que sige correctamente las reglas estructurales de formación del documento y el nombre de elementos y atributos.
- Reglas = DTD o XML Schema.

2. eXtensible Markup Language

Validación XML

- **Document Type Definition (DTD):**
  - Notación EBNF (no en XML).
  - Lista los nombres de elementos que pueden aparecer en los documentos XML que sigan el DTD.
  - Qué elementos pueden aparecer en combinación con otros.
  - Cómo se pueden anidar los elementos.
  - Qué atributos están disponibles y en qué elementos.
  - ...

2. eXtensible Markup Language

Validación XML

- Ejemplo de DTD:

```
<!DOCTYPE BOOK [
   <!ELEMENT p (#PCDATA)>
   <!ELEMENT BOOK   (OPENER,SUBTITLE?,INTRODUCTION?,(SECTION | PART)+)>
   <!ELEMENT OPENER    (TITLE_TEXT)*>
   <!ELEMENT TITLE_TEXT   (#PCDATA)>
   <!ELEMENT SUBTITLE     (#PCDATA)>
   <!ELEMENT INTRODUCTION (HEADER, p+)+>
   <!ELEMENT PART         (HEADER, CHAPTER+)>
   <!ELEMENT SECTION      (HEADER, p+)>
   <!ELEMENT HEADER       (#PCDATA)>
   <!ELEMENT CHAPTER      (CHAPTER_NUMBER, CHAPTER_TEXT)>
   <!ELEMENT CHAPTER_NUMBER (#PCDATA)>
   <!ELEMENT CHAPTER_TEXT (p)+>
      ]>
```

(Ejemplo de http://www.cs.rpi.edu/~puninj/XMLJ/classes/class3/all.html)

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE BOOK SYSTEM
"http://www.library.org/book.dtd">
<BOOK>
   <OPENER>
      <TITLE_TEXT>All About Me</TITLE_TEXT>
   </OPENER>
   <PART>
      <HEADER>Welcome To My Book</HEADER>
      <CHAPTER>
         <CHAPTER_NUMBER>CHAPTER
1</CHAPTER_NUMBER>
         <CHAPTER_TEXT>
            <p>Glad you want to hear about me.</p>
            <p>There's so much to say!</p>
            <p>Where should we start?</p>
            <p>How about more about me?</p>
         </CHAPTER_TEXT>
      </CHAPTER>
   </PART>
</BOOK>
```

- a+ - Una o más ocurrencias de a: <!ELEMENT BOOK (CHAPTER)+>
- a* - Cero o más ocurrencias de a: <!ELEMENT List (Object)*>
- a? - a o nada: <!ELEMENT Table (plate)?>
- a, b - a seguido de b: <!ELEMENT SUM (op1, op2)>
- a | b – a ó b pero no ambos: <!ELEMENT POINT (COORDINATES | POLAR)>
- (expresión) - expresión tratada como una unidad: <!ELEMENT CHAPTER (INTRODUCTION, (P | QUOTE | NOTE)*, DIV*)>
- PCDATA – Cualquier tipo de texto (se analiza).
- CDATA – Cualquier tipo de texto (no se analiza).

2. eXtensible Markup Language

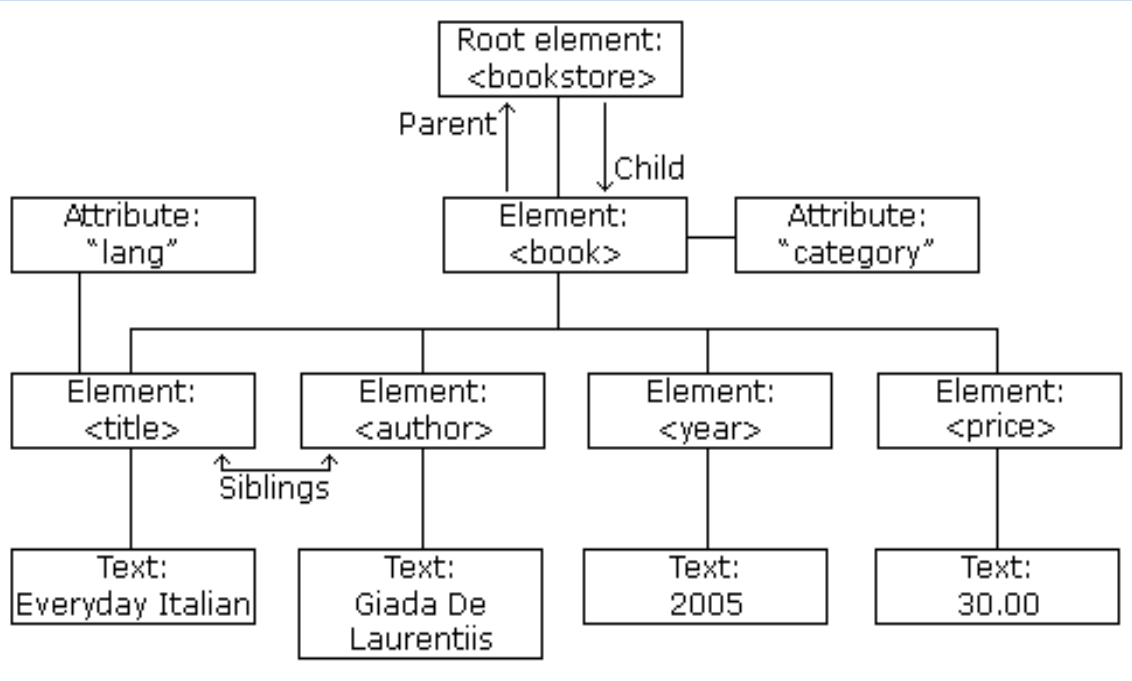
Validación XML

- **XML Schema**:
  - El DTD tiene limitaciones:
    - No está escrito en XML.
    - Es muy pobre con los tipos de datos.
  - Ventajas: las contrarias a las limitaciones.
    - Se utilizan espacios de nombres.
    - El XML Schema es a su vez un documento XML.

2. eXtensible Markup Language

Documentos XML como árboles

```
<bookstore>
 <book category="COOKING">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
 </book>
 <book category="CHILDREN">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
 </book>
 <book category="WEB">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
 </book>
</bookstore>
```

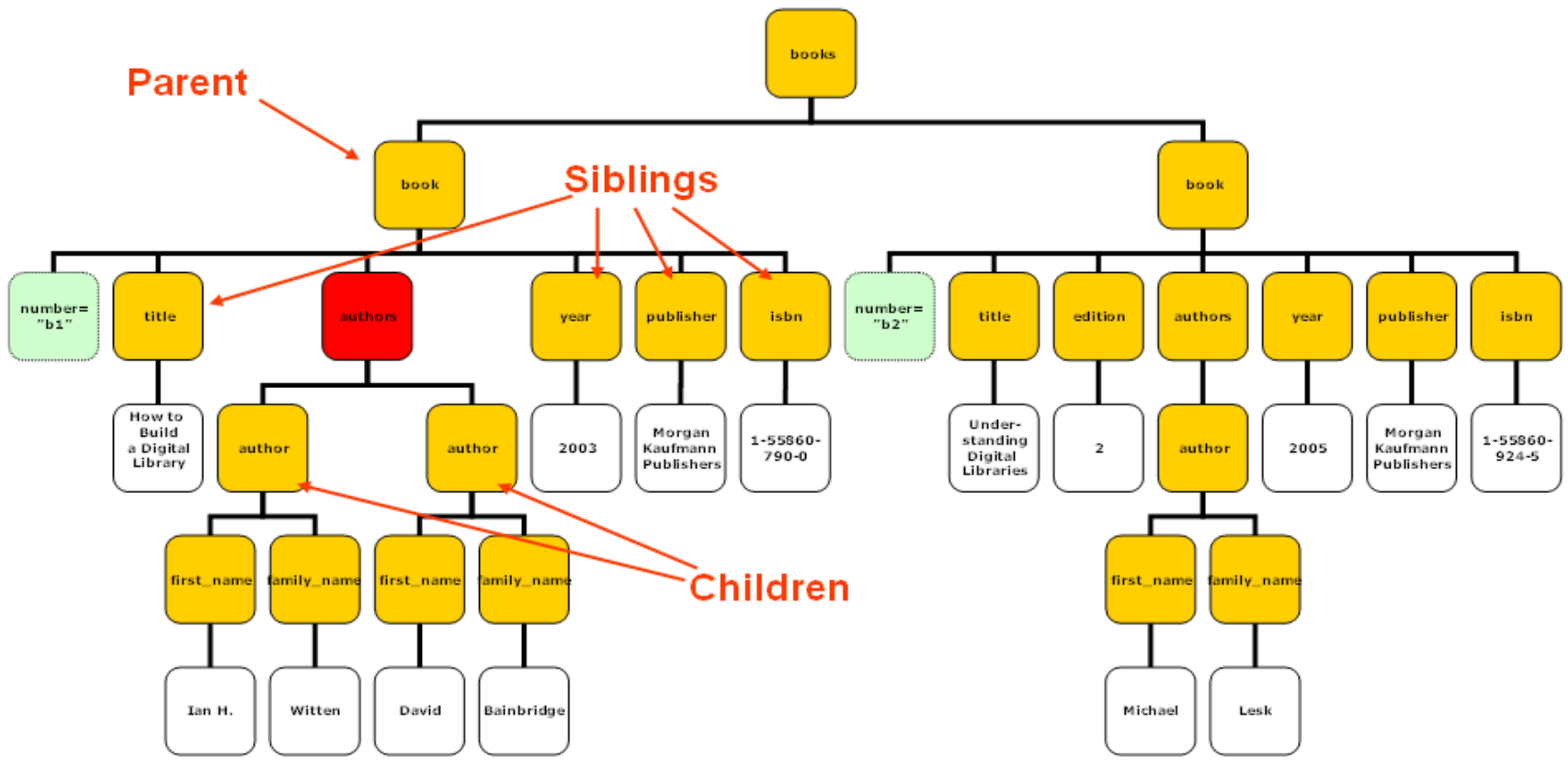


http://www.w3schools.com/xml/xml_tree.asp

2. eXtensible Markup Language

Documentos XML como árboles

http://www.jbi.hio.no/bibin/dill/summer_school/2009/xml-wellformed.html



Índice

1. Introducción.
2. eXtensible Markup Language.
3. **Recuperación de documentos estructurados**.
4. Consulta.
5. Indexación.
6. Recuperación.
7. Presentación de resultados.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

3. Recuperación de dococumentos estructurados

”La recuperación de información trata la representación, análisis, almacenaje, acceso y presentación de ítems de información”

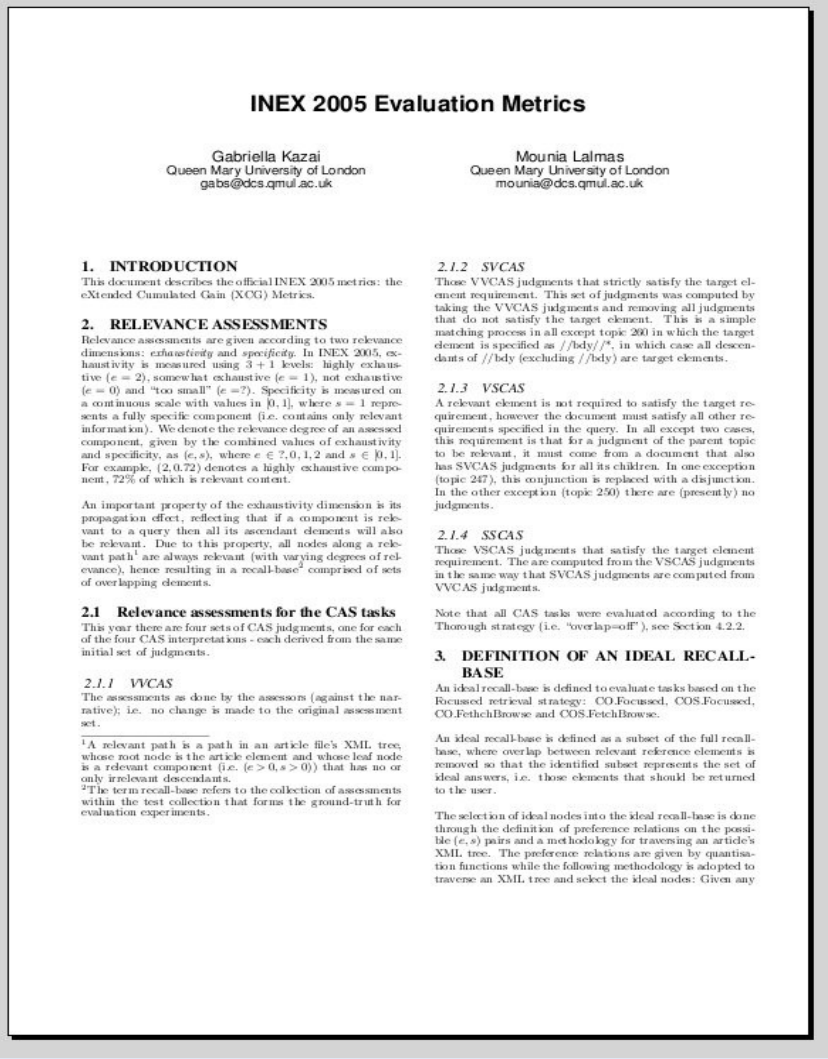
Información textual = no estructurada y estructurada.

- No estructurada = secuencia de palabras, puesta en forma de frases, dispuestas en párrafos, con objeto de transmitir algún significado (R.I. Clásica).

3. Recuperación de documentos estructurados

- Pero, a menudo, los documentos ofrecen información estructural: diseño y lógica.
  - **Estructura de diseño**: cómo se muestra el documento (a dos columnas con letra times 12).
  - **Estructura lógica**: describe las partes de un documento y sus relaciones.
- Ambos tipos de estructura se pueden expresar mediante un lenguaje de marcado: **HTML** y **XML**, respectivamente.
- Pero XML puede emplearse para especificar **información semántica sobre el contenido almacenado**.

3. Recuperación de documentos estructurados



INEX 2005 Evaluation Metrics

Gabriella Kazai
Queen Mary University of London
gabs@dcs.qmul.ac.uk

Mounia Lalmas
Queen Mary University of London
mounia@dcs.qmul.ac.uk

1. INTRODUCTION

This document describes the official INEX 2005 metrics: the eXtended Cumulated Gain (XCG) Metrics.

2. RELEVANCE ASSESSMENTS

Relevance assessments are given according to two relevance dimensions: *exhaustivity* and *specificity*. In INEX 2005, exhaustivity is measured using $3+1$ levels: highly exhaustive ($e=2$), somewhat exhaustive ($e=1$), not exhaustive ($e=0$) and "too small" ($e=?$). Specificity is measured on a continuous scale with values in $]0,1]$, where $s=1$ represents a fully specific component (i.e. contains only relevant information). We denote the relevance degree of an assessed component, given by the combined values of exhaustivity and specificity, as (e,s), where $e \in ?,0,1,2$ and $s \in]0,1]$. For example, $(2,0.72)$ denotes a highly exhaustive component, 72% of which is relevant content.

An important property of the exhaustivity dimension is its propagation effect, reflecting that if a component is relevant to a query then all its ascendant elements will also be relevant. Due to this property, all nodes along a relevant path[1] are always relevant (with varying degrees of relevance), hence resulting in a recall-base[2] comprised of sets of overlapping elements.

2.1 Relevance assessments for the CAS tasks

This year there are four sets of CAS judgments, one for each of the four CAS interpretations - each derived from the same initial set of judgments.

2.1.1 VVCAS

The assessments as done by the assessors (against the narrative); i.e. no change is made to the original assessment set.

2.1.2 SVCAS

Those VVCAS judgments that strictly satisfy the target element requirement. This set of judgments was computed by taking the VVCAS judgments and removing all judgments that do not satisfy the target element. This is a simple matching process in all except topic 260 in which the target element is specified as //bdy//*, in which case all descendants of //bdy (excluding //bdy) are target elements.

2.1.3 VSCAS

A relevant element is not required to satisfy the target requirement, however the document must satisfy all other requirements specified in the query. In all except two cases, this requirement is that for a judgment of the parent topic to be relevant, it must come from a document that also has SVCAS judgments for all its children. In one exception (topic 247), this conjunction is replaced with a disjunction. In the other exception (topic 250) there are (presently) no judgments.

2.1.4 SSCAS

Those VSCAS judgments that satisfy the target element requirement. The are computed from the VSCAS judgments in the same way that SVCAS judgments are computed from VVCAS judgments.

Note that all CAS tasks were evaluated according to the Thorough strategy (i.e. "overlap=off"), see Section 4.2.2.

3. DEFINITION OF AN IDEAL RECALL-BASE

An ideal recall-base is defined to evaluate tasks based on the Focussed retrieval strategy: CO.Focussed, COS.Focussed, CO.FetchBrowse and COS.FetchBrowse.

An ideal recall-base is defined as a subset of the full recall-base, where overlap between relevant reference elements is removed so that the identified subset represents the set of ideal answers, i.e. those elements that should be returned to the user.

The selection of ideal nodes into the ideal recall-base is done through the definition of preference relations on the possible (e,s) pairs and a methodology for traversing an article's XML tree. The preference relations are given by quantisation functions while the following methodology is adopted to traverse an XML tree and select the ideal nodes: Given any

[1] A relevant path is a path in an article file's XML tree, whose root node is the article element and whose leaf node is a relevant component (i.e. $(e>0, s>0)$) that has no or only irrelevant descendants.
[2] The term recall-base refers to the collection of assessments within the test collection that forms the ground-truth for evaluation experiments.

```
<article>
   <title> INEX 2005 Evaluation Metrics </title>
   <authors>
     <author> Gabriella Kazai </author>
     <author> Mounia Lalmas </author>
   </authors>
   <abstract/>
   <section>
     <title> 1. INTRODUCTION </title>
     <parragraph>
          This document describes the official INEX'05
           Metrics...
     </paragraph>
   </section>
   <section>
     <title> 2. RELEVANCE ASSESSMENTS </title>
     <paragraph>... </paragraph>
     <paragraph>... </paragraph>
   </section>
   …
</article>
```

3. Recuperación de documentos estructurados

- Un documento marcado en XML tiene una **estructura** que explícitamente **identifica partes** del documento **semánticamente separadas**.
- Esta propiedad puede ser explotada para ofrecer un acceso más potente a la información.
- Explotar esta estructura hace que lo que se encuentra el usuario sea de diferente granularidad (una frase, un párrafo, un conjunto de párrafos, una sección,..., un documento completo).

Recuperación de documentos estructurados
”Structured document retrieval”

3. Recuperación de documentos estructurados

Recuperación de documentos estructurados

Desarrollo de estrategias de recuperación, donde componentes de documentos, normalmente marcados en XML, en lugar de los documentos completos, son devueltos en respuesta a una consulta.

3. Recuperación de documentos estructurados

Vista de los documentos XML

- **Vista centrada en los datos (visión BB.DD.).**
  - Estructura bastante regular. Sin contenido anidado.
  - XML como un formato de intercambio para datos estructurados.
  - Se usa fundamentalmente entre aplicaciones de empresa.
  - Básicamente, una nueva representación del esquema relacional.
  - Datos ”semi-estructurados”.

3. Recuperación de documentos estructurados

Vista de los documentos XML

- **Vista centrada en el contenido (visión R.I.):**
  - Diseñados para consumo humano.
  - Estructura más irregular.
  - XML como el formato para representar la estructura lógica de los documentos.
  - Rico en textos.
  - Demanda la funcionalidad de las herramientas de recuperación de texto.

3. Recuperación de documentos estructurados

Vista de los documentos XML (Vista de datos)

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

3. Recuperación de documentos estructurados

Vista de los documentos XML (Vista contenido)

```
<CLASS name="DCS317" num_of_std="100">
    <LECTURER lecid="111">John</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>
```

3. Recuperación de documentos estructurados

INitiative for the Evaluation of XML retrieval (INEX)

Índice

1. Introducción.
2. eXtensible Markup Language.
3. Recuperación de documentos estructurados.
4. **Consulta**.
5. Indexación.
6. Recuperación.
7. Presentación de resultados.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

4. Consulta

- Con la aparición de XML se desarrollaron varios lenguajes de consulta, fundamentalmente, a documentos con una orientación en datos:
  - XML-QL, XSLT, XQL, Xpath, Quilt, XQuery.
- De ahí se pasó a abrirse la investigación en recuperación basada en contenido.
  - NEXI, ELIXIR, XIRQL, XQuery Full-Text.
- Nos centraremos en el segundo, pero añadiendo restricciones estructurales.

4. Consulta

Restricciones estructurales

- Deseo una **sección sobre evaluación de la recuperación en XML** contenida en un **capítulo que hable sobre iniciativas de evaluación**.
- Se imponen restricciones sobre **qué obtener** y **dónde buscar**, además de las restriciones de contenido.

4. Consulta

Restricciones estructurales

- Tres tipos:
  - Especificación del resultado objetivo: ***secciones sobre recuperación XML***.
  - Especificación de dónde buscar (contexto): *secciones sobre recuperación XML en documentos con* ***resúmenes sobre iniciativas de evaluación***.
  - Construcción de resultados: *el título de una sección,* ***junto a su primer y último párrafo agrupados*** (común en BB.DD.).

4. Consulta

Tipos de lenguajes contenido y estructura

- **Basados en marcas**:
  - *sección: recuperación xml*
  - XSEarch.
- **Basados en caminos**: (basados en XPath)
  - *//document[about(.,recuperación de información)]*
    //section [about(., recuperación xml)]
  - Xpath, XIRQL, NEXI.

4. Consulta

Tipos de lenguajes contenido y estructura

- **Basados en cláusulas** (muy parecidos a SQL):

  for &X in /document/section

  where $x/title = ”recuperación XML”

  return $x

  - XQuery Full-Text.

4. Consulta

XPath

- Objetivo: acceder o navegar entre los componentes del documento XML.
- Location paths: secuencia de pasos de navegación.
  - **libro/editor/@isbn** (/= descendiente directo).
  - **libro//editor** (// = cualquier camino entre *libro* y *editor*).
  - **//titulo** (cualquier elemento *título*).
  - **.//titulo** (cualquier elemento *titulo* del elemento actual).
  - **../editor** (cualquier elemento editor del elemento padre).
  - **//libro[@año = 2002]** (cualquier elemento *libro* con *año* 2002).
  - **//articulo//autor[1]** (el primer autor que cuelgue de un artículo). Predicados posicionales.
  - Función **contains(elemento, texto)**: verdadero o falso, según contenga el elemento el texto.

4. Consulta

NEXI

- **Narrowed Extended XPath I.**
- Diseñado para INEX como un lenguaje para evaluación de recuperación XML orientada a contenido.
- Usado para formular consultas de contenido y estructura.
- Subconjunto de XPath pero mejorado.
- Se incluye la **cláusula about**: un elemento trate de sobre un contenido.
- Se eliminan, por ejemplo, predicados posicionales, relaciones padre-hijo, hermanos, etc.

4. Consulta

NEXI

Consulta: //A[B]
Elemento objetivo: A
Significado: Devuelve elementos A sobre B.

Consulta: //A[B]//C
Elemento objetivo: A//C
Significado: Devuelve descendientes C de A donde A trate de B.

Consulta: //A[B]//C[D]
Elemento objetivo: A//C
Significado: Devuelve descendientes C de A donde A trate de B y C, descendiente de A, trate de B.

4. Consulta

NEXI

- //article[about(.//body, ”information retrieval”)]//
  section[about(.,xml) and about (.,retrieval)]

- //article[about(.//bdy, "artificial intelligence") and
  .//yr<=2000]//bdy[about(., chess) and about(., algorithm)]

- //article//(abs|sec)[about(.,+multimedia "document models"
  "content authoring")]

Índice

1. Introducción.
2. eXtensible Markup Language.
3. Recuperación de documentos estructurados.
4. Consulta.
5. **Indexación**.
6. Recuperación.
7. Presentación de resultados.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

5. Indexación

- En general:
  - Primera tarea de un S.R.I.
- **Objetivo**: obtener una representación del documento.
- **Resultado**: **Índice**.

Será empleado por el S.R.I. para calcular el grado de relevancia del documento dada una cierta consulta y generar una lista ordenada.

5. Indexación

En Recuperación XML:

- No existe una unidad concreta de recuperación predefinida.
- Todo puede ser una respuesta potencial a una consulta.

Por tanto...

¿Qué se indexa?

5. Indexación

- Otra cuestión a tener en cuenta: El **cálculo de las pesos de los términos**.
- En documentos planos: *tf* e *idf*.
- En doc. XML (la estrategia más simple): a nivel de elemento:
  - **etf**: frecuencia del término en el elemento.
  - **ief**: ”inverse element frequency”.

¿Cómo se hace?

5. Indexación

¿Qué indexar y cómo calcular estadísticas?

Estrategias de indexación:

- Basada en elementos.
- Sólo hojas.
- Basada en agregación.
- Selectiva.
- Distribuida.

5. Indexación

Indexación basada en elementos.

- Se indexa cada elemento asumiendo el texto contenido en él directamente y el de sus descendientes.

```
<article>
  <title>XXX</title>
  <abstract>YYY</abstract>
   <body>
     <sec>ZZZ</sec>
     <sec>ZZZ</sec>
   </body>
  </article>
```

```
<article>XXX YYY ZZZ ZZZ </article>
<title>XXX</title>
<abstract>YYY</abstract>
<body>ZZZ ZZZ</body>
<sec>ZZZ</sec>
<sec>ZZZ</sec>
```

5. Indexación

Indexación basada en elementos.

- Ventajas: se transforma en un problema de R.I. clásico (bien entendido).
- Se le puede asignar un valor de relevancia como si fuera un documento plano.

- Desventaja: índice muy redundante y grande.

Ej. INEX 2002 – 2004: 12.000 artículos, 8.000.000 elementos.

5. Indexación

Indexación basada en elementos: pesos.

- **etf** e **ief** se calculan en cada elemento sin tener en cuenta la naturaleza anidada de éstos.

Alternativas:

- **ief** calculado sólo considerando elementos del mismo tipo.
- **ief** calculado através de los documentos (idf).

No hay demostración empírica de cuál es mejor.

5. Indexación

Indexación de hojas sólo.

Sólo se indexan los elementos hojas.

```
<article>
 <title>XXX</title>
<abstract>YYY</abstract>
<body>
   <sec>ZZZ</sec>
   <sec>ZZZ</sec>
</body>
</article>
```

```
<title>XXX</title>
<abstract>YYY</abstract>
<sec>ZZZ</sec>
<sec>ZZZ</sec>
```

5. Indexación

Indexación de hojas sólo.

- Ventajas: índice más pequeño y sin redundancias.
  - Se calcula el grado de relevancia de los nodos hoja y se propaga hacia el resto de nodos (combinación).

- Desventajas: requiere algoritmos de propagación eficientes en tiempo de consulta.

Pesos:

ief se calcula através de los nodos hojas.

5. Indexación

Indexación basada en agregación.

Sólo se indexan los elementos hojas.

```
<article>
 <title>XXX</title>
 <abstract>YYY</abstract>
 <body>
  <sec>ZZZ</sec>
  <sec>ZZZ</sec>
 </body>
 </article>
```

```
<title>XXX</title>
<abstract>YYY</abstract>
<sec>ZZZ</sec>
<sec>ZZZ</sec>
```

5. Indexación

Indexación basada en agregación: Pesos.

- Se agregan las estadísticas de los términos en el texto de la unidad de indexación (si lo tienen) con los términos que aparecen en las unidades hijas.
- Se puede establecer la **importancia de una unidad en otra** a la hora de calcular las estadísticas.
  - Por ejemplo, es más importante un resumen de un artículo que sus conclusiones.

5. Indexación

Indexación selectiva.

- Sólo se indexan aquellos tipos de elementos que se configuren como unidades de recuperación con interés desde el punto de vista de la recuperación.
- Reduce el tamaño del índice (no tanto como la basada en nodos hojas).
- Se utiliza en combinación con alguna otra.

5. Indexación

Indexación selectiva. Dos estrategias:

- Eliminar elementos pequeños (con dos o tres palabras). Por ejemplo, negritas, cursivas, etc.
  - Aunque no tengan interés desde el punto de vista de la recuperación, pueden influenciar la relevancia de otra unidad que los contenga.
- Seleccionar elementos de marcas concretas. Por ejemplo, artículos, resúmenes, secciones pero no párrafos.
  - Selección por diseñador del S.R.I. o basada en juicios de relevancia.

5. Indexación

Indexación distribuida.

- Se construye un índice por cada tipo de elemento.
- Los pesos de los términos se calculan en cada índice.
- La consulta se lanza a cada índice y se mezclan los resultados.

5. Indexación

Indexación distribuida.

Indexación separada – consulta en paralelo a cada índice

5. Indexación

- No está clara qué estrategia es la mejor.
- La elección dependerá de la colección, los tipos de elementos y sus relaciones.
- La elección de la estrategia tendrá efectos en la estrategia de obtención del RSV.

Índice

1. Introducción.
2. eXtensible Markup Language.
3. Recuperación de documentos estructurados.
4. Consulta.
5. Indexación.
6. **Recuperación**.
7. Presentación de resultados.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

6. Recuperación

Estimación del grado de relevancia de cada elemento a partir su contenido textual y ordenación de éstos.

- Muchos modelos de la R.I. plana se han exportado a la recuperacion XML.

- Técnicas principales:

  Contextualización, propagación, agregación, mezcla y consultas de contenido y estructura.

6. Recuperación

Un ejemplo de función de estimación del grado de relevancia (RSV) de un elemento dada una consulta: Language Models.

- $q = (t_1, t_2, ..., t_n)$.
- Un elemento e y su modelo de lenguaje, M_e.
- Los elementos se ordenan según su P(e|q).

6. Recuperación

$$P(e|q) \propto P(e)\,P(q|M_e)$$

$$P(t_1, t_2, \ldots, t_n|M_e) = \prod_{i=1}^{n} \lambda \cdot P(t_i|e) + (1-\lambda) \cdot P(t_i|C)$$

Donde

- P(e) es la probabilidad a priori de relevancia de e.
- $P(q|M_e)$ es la probabilidad de que la consulta q sea generada por el modelo de lenguaje de M_e.
- $P(t_i|e)$ es la probabilidad del término t_i en el elemento e.
- $P(t_i|C)$ es la probabilidad del término t_i en la colección.
- λ es un parámetro de suavizado.

6. Recuperación

Contextualización

- El hecho de que *un elemento no contenga todos los términos de la consulta* pero *esté contenido en un documento que sí los contiene* hace que sea **más relevante** que si *está en un documento que no contiene a todos* los elementos de la consulta.
- **Considerar el contexto de un elemento para calcular su relevancia**.

6. Recuperación

Contextualización

- **Contexto = elemento padre y/o (alguno de) sus ancestros**, incluyendo el documento completo.
- Lo más común: contexto = elemento raíz (documento).
- Combinación del peso de elemento en cuestión con el del raíz (o con todos los de su contexto).
- Mejora el rendimiento de la recuperación.

6. Recuperación

Contextualización

Formas:

- Media de los RSVs.
- Media ponderada de los RSVs: se enfatiza la importancia de uno con respecto al otro.

6. Recuperación

Propagación

- Se emplea cuando se indexan sólo las hojas.
- Se estima la relevancia de los elementos hojas.
- La relevancia del resto de elementos, incluído el raíz, se calcula mediante un proceso de propagación.
- Lo más habitual: una suma ponderada de los RSVs.

6. Recuperación

Propagación (Ejemplo I)

- Sea **e** un elemento interior y $\mathbf{e_h}$ un elemento hoja de **e**.
- **rsv(·,q)** la función de estimación de relevancia para un elemento y una consulta **q**. Si es elemento hoja se calcula directamente.
- $\mathbf{d(e,e_h)}$ es la distancia entre **e** y $\mathbf{e_h}$ en el camino que los une.

6. Recuperación

Propagación (Ejemplo I)

- Se pondera por la distancia: cuanta más distancia haya entre un elemento y sus elementos hojas, con menos fuerza contribuirán a la relevancia del elemento.

$$rsv(e,q)=\sum_{e_h}\left(1-2\lambda\frac{d(e,e_h)}{d(e,e_h)+d(e_{raiz},e_h)}\right)^2\times score(e_h,q)$$

- **λ** es una cte. que regula la aportación de los descendientes en la propagación.

6. Recuperación

Propagación (Ejemplo II: GPX)

- Se pondera por el número de hijos de un elemento.
- **m** = número de hijos de e recuperados.
- $\mathbf{e_h}$ es un elemento hijo de e.
- **D(m)** = factor de decaimiento.
- Si m=1, D(1)= 0.49 → **e** estará más bajo en la ordenación que e_h.
- Para varos hijos recuperados, D(m)= 0,99 → **e** estará más alto que sus hijos.

6. Recuperación

Propagación (Ejemplo II: GPX)

$$rsv(e,q) = D(m) \sum_{e_h} rsv(e_h, q)$$

Por ejemplo:

- Una sección con un único párrafo relevante recuperado se considerará menos relevante que su hijo (después de recuperar el párrafo, recuperar la sección no ofrece nada)

pero...

- una sección con varios párrafos relevantes recuperados será dispuesta más alta en la ordenación (a partir de dicha sección se podrá acceder a dichos párrafos).

6. Recuperación

Propagación (Ejemplo III: XFIRM)

- Ponderación por medio de varios pesos más contextualización:

$$rsv(e,q) = \rho \times m \times \sum_{e_h} \alpha^{d(e,e_h)-1} + \beta(e_h) \times rsv(e_h,q) + (1-\rho) \times rsv(e_{raiz},q)$$

- **m** = número de nodos hoja recuperados contenidos en **e**.
- **ρ** parámetro que mide la distancia entre la relevancia del elemento propagado y el del raíz.
- **$\beta(e_h)$** mide la importancia del de los elementos hojas en la propagación (mayor importancia a títulos, negritas itálicas,etc.).

6. Recuperación

Agregación

Idea subyacente:

- la representación de un elemento XML se puede ver como una agregación de su propio contenido (si existe) y de la representación del contenido de elementos relacionados (si existen – generalmente elementos hijos).

Agregación vs Propagación:

- En la primera, la combinación se realiza en las representaciones, mientras que en la segunda, en los valores de relevancia.

6. Recuperación

Agregación

- La representación de elementos hojas se obtiene en tiempo de indexación.
- Una función de agregación se emplea para generar la representación de los elementos interiores.
- Pueden existir parámetros para determinar cómo la representación de un elemento puede venir determinada por la de sus hijos.

6. Recuperación

Agregación (Tipos)

- **Global**: en tiempo de idexación (no tiene buena escalabilidad y puede llegar a ser ineficiente).
- **Local**: en tiempo de consulta (más usado y mejores resultados).

6. Recuperación

Agregación (Ejemplo basado en LMs)

$$P(t_i|M_{e\,propio}) = \lambda \cdot P(t_i|C) + (1-\lambda) \cdot P(t_i|e_{propio})$$

$$P(t_i|M_e) = \omega_0 P(t_i|M_{e\,propio}) + \sum_j \omega_j P(t_i|M_{ej})$$

$$Ordenación \Rightarrow P(t_1, t_2, \ldots, t_n|M_e) = \prod_{i=1}^{n} \lambda \cdot P(t_i|M_e)$$

- $\mathbf{e_{propio}}$ = contenido propio del elemento **e**.
- $\mathbf{e_j}$ = elemento hijo de e.
- $\mathbf{M_{e\ propio}}$ y $\mathbf{M_{ej}}$ = Modelos del lenguaje basados en el propio contenido de e y $\mathbf{e_j}$, respectivamente.
- $\mathbf{\omega_0 + \Sigma_j\ \omega_j = 1}$. Los parámetros ω miden la contribución de cada modelo.

6. Recuperación

Mezcla

- Se calcula rsv(q,q) en cada índice, es decir, el rsv de la consulta como si fuera un elemento del índice.
- Cada rsv de cada elemento queda normalizado por el rsv(q,q) en su índice.
- Los valores son comparables y se pueden ordenar todos los elementos en una única lista.

6. Recuperación

Mezcla

Otro enfoque:

- Varias lista obtenidas a partir de la aplicación de varios modelos de R.I. para XML.
- Fusión de ordenaciones.

6. Recuperación

Procesamiento de restricciones estructurales

- Consultas que además de expresar un interés por un contenido textual, lo hacen indicando restricciones sobre qué tipos de elementos obtener y dónde buscar.
- En INEX, se formulan mediante **NEXI**.

6. Recuperación

Procesamiento de restricciones estructurales

En INEX, se ven como pistas que indican dónde buscar información relevante.

Razones:

Si ya es difícil que el usuario pueda expresar de manera correcta una consulta por contenido, más aún se acentuará esta dificultad con consultas estructuradas *(pedir párrafos y devolver secciones)*.

Se piensa en la comunidad XML que es más importante satisfacer el criterio de contenido que el estructural (pedir secciones y devolver un resumen).

6. Recuperación

Procesamiento de restricciones estructurales

Enfoques:

- Construir un diccionario de elementos sinónimos.
  - Basado en sintaxis (<p> y <p1> son equivalentes).
  - Basado en semántica (<city> y <town>).
  - Si se pide <section> todos sus sinónimos podrán ser devueltos (si son relevantes).

6. Recuperación

Procesamiento de restricciones estructurales

Enfoques:

- Potenciación de estructura: el rsv de cada elemento se calcula independientemente de las restricciones estructurales, pero se potencia con respecto a cómo cada elemento las satisface.
- Se suele utilizar en la propagación.

6. Recuperación

Procesamiento de restricciones estructurales

- Resolución de consultas con estructura:

”Recupera párrafos sobre modelos de recuperación contenidos en secciones sobre recuperación XML”.

Dos consultas:

- párrafos sobre modelos de recuperación, y
- secciones sobre recuperación XML.

6. Recuperación

Procesamiento de restricciones estructurales

- Se procesa **cada subconsulta separadamente**.
- Se obtiene una **ordenación** de elementos **por subconsulta**.
- **Se obtiene sólo una ordenación**: sólo los elementos párrafos cuyos ancestros *estén contenidos* en una sección devuelta en la segunda ordenación se devolverán como resultado.
- *Estar contenidos* de forma estricta o difusa.

Índice

1. Introducción.
2. eXtensible Markup Language.
3. Recuperación de documentos estructurados.
4. Consulta.
5. Indexación.
6. Recuperación.
7. **Presentación de resultados**.
8. Evaluación.
9. Un prototipo de recuperación XML: Seda.

7. Presentación de resultados

- Un S.R.I. para XML devuelve una ordenación de elementos según su grado de relevancia, pero...

  ¿Es esto lo que más le interesa al usuario?

- Posiblemente **no** porque los elementos XML no son independientes:
  - Existen ***solapamientos***: p.e., una sección y sus párrafos.
  - Existen **elementos "hermanos"**: p.e., dos párrafos consecutivos de una sección.

7. Presentación de resultados

Según estudios de usuario:

- A los usuarios no le gusta que le devuelvan mucha información redundante → métodos para **eliminar solapamientos (focused retrieval task)**.
- No sólo quieren tener acceso a los elementos relevantes sino al contexto de estos → métodos para presentar **resultados en contexto (relevant in context task)**.
- Les gusta un punto de entrada para empezar a leer el documento → métodos para identificar los **mejores puntos de entrada (best entry points task)**.

7. Presentación de resultados

Manejo de solapamientos

- Cuando un elemento haya sido estimado como relevante es probable que sus ancestros también lo sean (aunque con un grado diferente).
- Incluso puede tener varios descendientes también relevantes.
- Pueden estar todos en la lista de elementos relevantes → Puede distraer a los usuarios.
- Objetivo:

¿Qué elementos dar al usuario y cuáles no?

7. Presentación de resultados

Manejo de solapamientos (Método I)

- El método más sencillo: **elegir un tipo de elemento y eliminar el resto**.
- Desventajas:
  - No se elimina del todo porque puede haber elementos del mismo tipo que se solapen.
  - Hay que conocer la colección para decidir cuáles son los elementos más útiles para todas las consultas.
  - No tiene flexibilidad: esos elementos más útiles dependerán de la consulta.

7. Presentación de resultados

Manejo de solapamientos (Método II)

- **Filtrado de fuerza bruta**:
  - Tomar el elemento más alto de la ordenación.
  - Eliminar todos sus descendientes y ascendientes.
  - Se aplica iterativamente.
- Desventaja:
  - Se depende de los elementos que el S.R.I. haya colocado en la parte de arriba de la ordenación.

7. Presentación de resultados

Manejo de solapamientos (Método III)

- **Tolerar algo de solapamiento**.
- Se modifica la ordenación inicial, actualizando el rsv de los elementos que están contenidos o contienen otros elementos más arriba en la ordenación como forma de reflejar que son redundantes.
- ¿Cómo? Reduciendo la importancia de los términos que aparecen en elementos que ya han sido vistos en la fórmula de estimación del grado de relevancia.
- **No elimina realmente el solapamiento**, sino que baja los elementos solapados. Puede ser graduada.

7. Presentación de resultados

Manejo de solapamientos (Método IV)

- También basada en una reordenación.
- El rsv de cada elemento se recalcula teniendo en cuenta los rsv de sus elementos descendientes.
  - Máximo o media.
- La nueva ordenación se filtra seleccionando los elementos ordenados más arriba y eliminando todos los ancestros o todos los descendientes de esos elementos, no ambos.
- **Nueva ordenación sin solapamiento**.

7. Presentación de resultados

Manejo de solapamientos (Método V)

- Ordenación sin solapamientos.
- Está basada en el concepto de utilidad de un elemento. Función de utilidad que depende de:
  - el rsv del elemento,
  - su tamaño y
  - de la cantidad de texto contenido en elementos hijos no recuperados (información irrelevante).
- Si $u(e) > \Sigma_j u(e_j)$, entonces se selecciona **e** y se borran sus hijos. Si $u(e_j) > \sigma$, se selecciona y **e** se elimina.

7. Presentación de resultados

Manejo de solapamientos

Eliminación de solapamientos se hace **en tiempo de ejecución**.

Los métodos tienen que ser **eficientes**.

7. Presentación de resultados

Presentación de elementos en contexto

- Dos consideraciones iniciales:
  - Los usuarios quieren **contexto**.
  - No quieren elementos de un mismo documento diseminados a lo largo de la ordenación.
- Dos posibles soluciones:
  - Presentar un **índice** del documento.
  - Devolver una **ordenación de documentos**, dispuestos según su relevancia, y dentro de cada documento, identificar los elementos más relevantes (Tarea de INEX "**Relevant in Context**").

7. Presentación de resultados

Presentación de elementos en contexto (Método I)

1) Partir de una ordenación libre de solapamientos.
2) Agrupar los elementos por documentos.
3) Calcular un valor de relevancia para cada documento.
   - Máximo valor de los elementos o algún tipo de agregación.
4) Reordenar los documentos y ofrecer al usuario dicha ordenación (conteniendo una sub-ordenación de los elementos de cada documento).

7. Presentación de resultados

Presentación de elementos en contexto (Método II)

1) Partir de una ordenación original del S.R.I.
2) Agrupar los elementos por documentos.
3) Calcular un valor de relevancia para cada documento.
4) Realizar una eliminación de solapamiento dentro de cada documento.
5) Reordenar los documentos y ofrecer al usuario dicha ordenación.

- Ordenación más precisa de documentos pues considera todos los contenidos relevantes.

7. Presentación de resultados

Puntos de entrada

- En algunas aplicaciones, más interesante ofrecer un punto de entrada a un documento que ofrecer una ordenación de elementos.

¿Punto de entrada? Lugar para empezar a leer un documento según una consulta.

- **Best in context task de INEX**: identificar documentos relevantes ordenados según su grado de relevancia, y para cada uno, un puntero a donde comienza el texto más relevante.

7. Presentación de resultados

Puntos de entrada

¿De dónde se puede partir?

(1)La ordenación inicial.

(2)La ordenación libre de solapamientos.

(3)La ordenación ”relevant in context”.

- Para (1) y (2) se selecciona el elemento con mayor relevancia como mejor punto de entrada.
- Se ordenan los documentos de acuerdo a la relevancia de su punto de entrada. Sólo un resultado por documento.

7. Presentación de resultados

Puntos de entrada

- Para (3):
  - Opción 1: la ordenación es la dada por la agrupación ya hecha.
  - Opción 2: se reordenan los documentos, por ejemplo, sumando los rsv de sus elementos. Documentos con muchos elementos relevantes irán a la parte alta.
- En ambos casos también se devuelve sólo el mejor elemento por documento.

Índice

1. Introducción.
2. eXtensible Markup Language.
3. Recuperación de documentos estructurados.
4. Consulta.
5. Indexación.
6. Recuperación.
7. Presentación de resultados.
8. **Evaluación**.
9. Un prototipo de recuperación XML: Seda.

8. Evaluación

- Es importante evaluar nuestros sistemas para determinar su efectividad recuperadora.
- El enfoque predominante es el uso de colecciones de prueba: Documentos, consultas, juicios y medidas de evaluación.
- En Recuperación XML no había nada utilizable para la evaluación.

INitiative for the Evaluation of XML retrieval (INEX)

8. Evaluación

Colecciones de INEX

- **Hasta INEX 2004:**
  - 12.107 artículos en XML de 12 revistas y 6 transactions de la IEEE Computer Society, desde 1995 a 2002.
  - 494 MB y 8.000.000 de elementos.
  - 1.532 elementos en media por documento.
  - 6,9 elementos de profundidad media por documento.

8. Evaluación

Colecciones de INEX

- **INEX 2005**:
  - Se añaden 4.712 artículos del período 2002-2004.
  - Total de 16.819 artículos.
  - 764 MB.
  - 11 millones de elementos.
  - 176 marcas únicas.

8. Evaluación

Colecciones de INEX

- **Desde INEX 2006**:
  - 659.388 artículos de la Wikipedia en inglés.
  - 4,6 GB sin imágenes.
  - 52 millones de elementos.
  - 161,3 elementos en media por documento XML.
  - 1.241 marcas únicas.

8. Evaluación

Colecciones de INEX

- **INEX 2009**:
  - 50,7 GB.
  - 2.666.190 artículos de la Wikipedia en inglés.

8. Evaluación

Consultas de INEX

Están compuestas por:

- **Título**: explicación breve de la necesidad de información.
- **Descripción**: definición de una o dos frases de la necesidad de información.
- **Narrativa**: explicación detallada y descripción de qué hace que un documento (elemento XML) sea relevante.

8. Evaluación

Consultas de INEX (Tipos)

- **Content-Only (CO)**: consultas que ignoran la estructura del documento. Resultado: elementos de cualquier granularidad.
- **Content-and-structure (CAS)**: expresan tanto contenido como estructura de los elementos. NEXI.
  - **Contexto**: dónde buscar.
  - **Objetivo**: qué tipos de elementos recuperar.
  - Ej. Resúmenes de artículos sobre R.I. que tengan secciones que hablen sobre evaluación XML.

8. Evaluación

Consultas de INEX (Tipos)

- **Content-Only (CO)**: usuarios que no tienen conocimiento de la estructura o no quieren hacer uso de ella. Gran mayoría de usuarios.
- **Content-and-structure (CAS)**: usuarios que conocen la estructura de la colección y quieren emplearla como forma de mejorar la precisión y obtener información más completa.

8. Evaluación

Consultas de INEX (CO)

<title>
Open standards for digital video in distance learning
</title>

<description>
Open technologies behind media streaming in distance learning projects
</description>

<narrative>
I am looking for articles/components discussing methodologies of digital video production and distribution that respect free access to media content through internet or via CD-ROMs or DVDs in connection to the learning process. Discussions of open versus proprietary standards of storing and sending digital video will be appreciated.
</narrative>

<keywords>
media streaming,video streaming,audio streaming, digital video,distance learning,open standards,free access
</keywords>

8. Evaluación

Consultas de INEX (CAS)

<title>
//article[about(.,'formal methods verify correctness aviation systems')]//sec//* [about(.,'case study application model checking theorem proving')]
</title>
<description>
Find documents discussing formal methods to verify correctness of aviation systems. From those articles extract parts discussing a case study of using model checking or theorem proving for the verification.
</description>
<narrative>
To be considered relevant a document must be about using formal methods to verify correctness of aviation systems, such as flight traffic control systems, airplane- or helicopter- parts. From those documents a section-part must be returned (I do not want the whole section, I want something smaller). That part should be about a case study of applying a model checker or a theorem proverb to the verification.
</narrative>
<keywords>
SPIN, SMV, PVS, SPARK, CWB
</keywords>

8. Evaluación

Consultas de INEX (Tipos)

- Con respecto a las CAS, hay **dos interpretaciones de las condiciones estructurales**:
  - **Estricta**: correspondencia exacta. Más cercana a la visión de las bases de datos (orientada a datos).
  - **Vaga**: no se necesita que haya tal correspondencia. Más cercana a la visión de la R.I. (orientada a contenido).
- Generalmente sólo se aplican al objetivo, ya que no parecen ser relevante cómo se interprete el contexto.

8. Evaluación

Consultas de INEX (Tipos)

- Consultas creadas por los grupos participantes.
- Se seleccionan varias de ellas (organización).
- Se hacen los juicios de relevancia (grupos).

Año	Número de consultas seleccionadas	Número de consultas enjuiciadas
2002	60	54
2003	66	62
2004	71	60
2005	87	63
2006	125	114
2007	130	99
2008	135	70
2009	114	

8. Evaluación

Juicios de relevancia

- En el contexto de la recuperación XML, conjunto de elementos relevantes a cada consulta.
- Teniendo en cuenta que:
  - los elementos recuperados pueden tener cualquier tipo de granularidad,
  - un elemento y uno de sus hijos pueden ser relevantes,
  - el elemento hijo suele estar más focalizado en la consulta y el padre puede tener información irrelevante.
- Entonces, el elemento hijo es preferible porque no sólo es relevante sino que es más específico.

8. Evaluación

Juicios de relevancia

- En INEX 2002 y 2003, la relevancia se estableció en dos dimensiones:
  - **Relevancia sobre la consulta (topical relevance)**: grado con el que el elemento satisface la necesidad de información (**exahustividad**).
  - **Covertura del elemento (component coverage)**: grado con el que el elemento está focalizado en la necesidad de información (**especificidad**).

8. Evaluación

Juicios de relevancia

- **Grados de exahustividad**:
  - **No exahustivo (0)**: no contiene información sobre la consulta.
  - **Marginalmente exahustivo (1)**: menciona el tópico pero de pasada.
  - **Moderadamente exahustivo (2)**: habla sobre el tópico aunque no exahustivamente.
  - **Altamente exahustivo (3)**: se centra en la necesidad de información exahustivamente.

8. Evaluación

Juicios de relevancia

- **Grados de especificidad**:
  - **No específico (0)**: la consulta no constituye el asunto central del elemento.
  - **Marginalmente específico (1)**: la consulta sólo es un asunto menor.
  - **Bastante específico (2)**: la consulta es el asunto central, pero el elemento es demasiado pequeño como para ser una unidad de información con sentido.
  - **Altamente específico (3)**: la consulta es el tema principal del elemento y el único.

8. Evaluación

Juicios de relevancia

- El objetivo: los **altamente específicos y exhaustivos**.
- Realizar los juicios de relevancia era una tarea muy tediosa y larga. En INEX 2005 se cambia el método a uno con dos fases:
  - Se seleccionan fragmentos de texto que son relevantes.
    - La especificidad se calcula automáticamente en una escala [0,1] como la porción relevante del elemento.
  - Determinar la exahustividad de los elementos con texto seleccionado, así como sus padres.

8. Evaluación

Juicios de relevancia

- **Grados de exahustividad (modificados en INEX'05)**:
  - **No exahustivo (0)**: no habla de la consulta.
  - **Parcialmente exahustivo (1)**: menciona pocos aspectos de la consulta.
  - **Altamente exahustivo (2)**: trata todos, o la mayoría, de los aspectos de la consulta.
  - **Demasiado pequeño (3)**: contiene material relevante pero es demasiado pequeño como para ser relevante en sí mismo.

8. Evaluación

Juicios de relevancia

- Se pide a los participantes que envíen sus resultados para cada consulta (1500).
- Se crea un fondo de resultados con todos los resultados.
- Se seleccionan 500 documentos para ser enjuiciados por consulta.
- Cada grupo realiza los juicios de varias consultas.
- Finalmente, se evalúa la salida de cada envío con respecto a los juicios realizados.

8. Evaluación

Medidas de evaluación: XCG

- **eXtension of the Cumulated Gain.**
- Está basada en la acumulación de la ganancia asociada a los resultados devueltos. No en P-R. Está pensada para una relevancia multivaluada.
- La efectividad recuperadora de un sistema se obtiene comparando los valores de ganancia obtenidos por el sistema y los obtenidos por un sistema ideal.
- El sistema ideal se obtiene ordenando los documentos según sus valores de ganancia.
- En la recuperación XML, se puede realizar la ordenación considerando la combinación de las dos dimensiones de relevancia: el más exahustivo y específico. Funciones de cuantificación.
- Adecuado para la recuperación donde hay solapamiento.

8. Evaluación

Medidas de evaluación (Funciones de cuantificación)

- ¿Cómo diferenciar entre (1,3) y (3,3), …? (exh,esp).
- Varios modelos de usuario:
  - **Experto e impaciente**: sólo considera elementos altamente exahustivos y específicos (3,3).
  - **Experto e impaciente**: sólo elementos altamente específicos (3,3), (2,3) (1,3).
  - **Con mucho tiempo**: considera, en diferente grado, todos los relevantes (todo excepto (0,0)).

8. Evaluación

Medidas de evaluación (Funciones de cuantificación)

- Experto e impaciente:

$$f_{strict}(exh, esp) = \begin{cases} 1 \text{ if } exh = 3 \text{ and } esp = 3 \\ 0 \text{ en caso contrario} \end{cases}$$

- Con mucho tiempo:

$$f_{generalised}(exh, esp) = \begin{cases} 1.00 & if \quad (exh, esp) = (3,3) \\ 0.75 & if \quad (exh, esp) \in \{(2,3), (3,2), (3,1)\} \\ 0.50 & if \quad (exh, esp) \in \{(1,3), (2,2), (2,1)\} \\ 0.25 & if \quad (exh, esp) \in \{(1,1), (1,2)\} \\ 0.00 & if \quad (exh, esp) = (0,0) \end{cases}$$

8. Evaluación

Col = ⟨d4,d5,d2,d3,d1⟩

G = ⟨3,0,1,3,2⟩

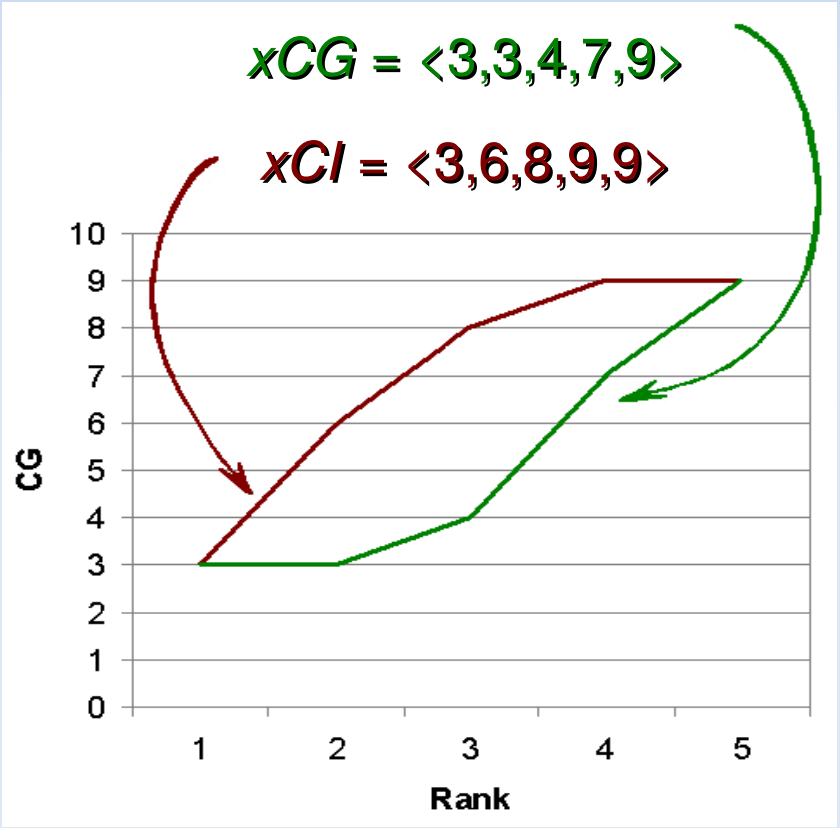
I = ⟨3,3,2,1,0⟩

Medidas de evaluación: XCG

- Vector de ganancia (***G***) a partir de la lista de documentos relevantes.
- Vector de ganancia ideal (***I***).
- Ganancia Acumulada (CG).

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

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



8. Evaluación

Medidas de evaluación: XCG

- Para la tarea focalizada (se elimina el solapamiento), se intenta premiar los elementos cercanos a los relevantes (near-missed).
- Se distingue entre el conjunto de elementos que deberían ser recuperados y aquellos que están estructuralmente relacionados con ellos.
- **Ideal recall-base**: el conjunto ordenado de elementos en los juicios de relevancia que quedan una vez se quitan solapamientos.

8. Evaluación

Medidas de evaluación: (HiXEVAL)

- Utilizado en las últimas ediciones de INEX, donde se indica qué texto es relevante.
- Objetivo: devolver aquellas unidades que tengan el máximo posible de texto señalado.
- Reformulación de las medidas de recall y precision.

$$Precision = \frac{cantidad\ de\ información\ relevante\ recuperada}{cantidad\ total\ de\ información\ recuperada}$$

$$Recall = \frac{cantidad\ de\ información\ relevante\ recuperada}{cantidad\ total\ de\ información\ relevante}$$

8. Evaluación

Medidas de evaluación: (HiXEVAL)

- e_i un elemento en la posición i-ésima.
- rsize(e_i) = cantidad de texto marcado (relevante) contenido en e_i para una consulta (medido en caracteres).
- Trel = cantidad total de texto señalado (relevante) en la colección para una consulta dada.
- size(e_i) = número total de caracteres contenidos en e_i.

$$P@r = \frac{\sum_{i=1}^{r} rsize(e_i)}{\sum_{i=1}^{r} size(e_i)}$$

$$R@r = \frac{1}{Trel} \sum_{i=1}^{r} rsize(e_i)$$

8. Evaluación

Otras tareas en XML (INEX)

- **Mutimedia**: cómo se explota la estructura XML para poder mejora la recuperación de objetos multimedia en un S.R.I. Multimedia.
- **Heterogénea**: recuperación con colecciones heterogéneas.
- **Minería de documentos**: clasificación, agrupamiento,...
- **Link-the-Wiki**: descubrimiento de enlaces automático.
- **Ordenación de entidades**: recuperar entidades que aparecen en una colección.
- **Interactiva**: estudio del comportamiento de los usuarios cuando tratan con S.R. XML.

Índice

9. Un prototipo de recuperación XML: Seda

Antecedentes:

- Grupo UTAI de la UGR.
- Comenzamos la investigación en R.I en 1998.
- Diseño y evaluación de varios modelos de R.I. Basados en Redes Bayesianas.
- En 2001 empezamos una nueva línea en recuperación XML.
- Modelo CID (redes bayesianas y diagramas de influencia).
- Garnata = Sistema de Recuperación de documentos estructurados.

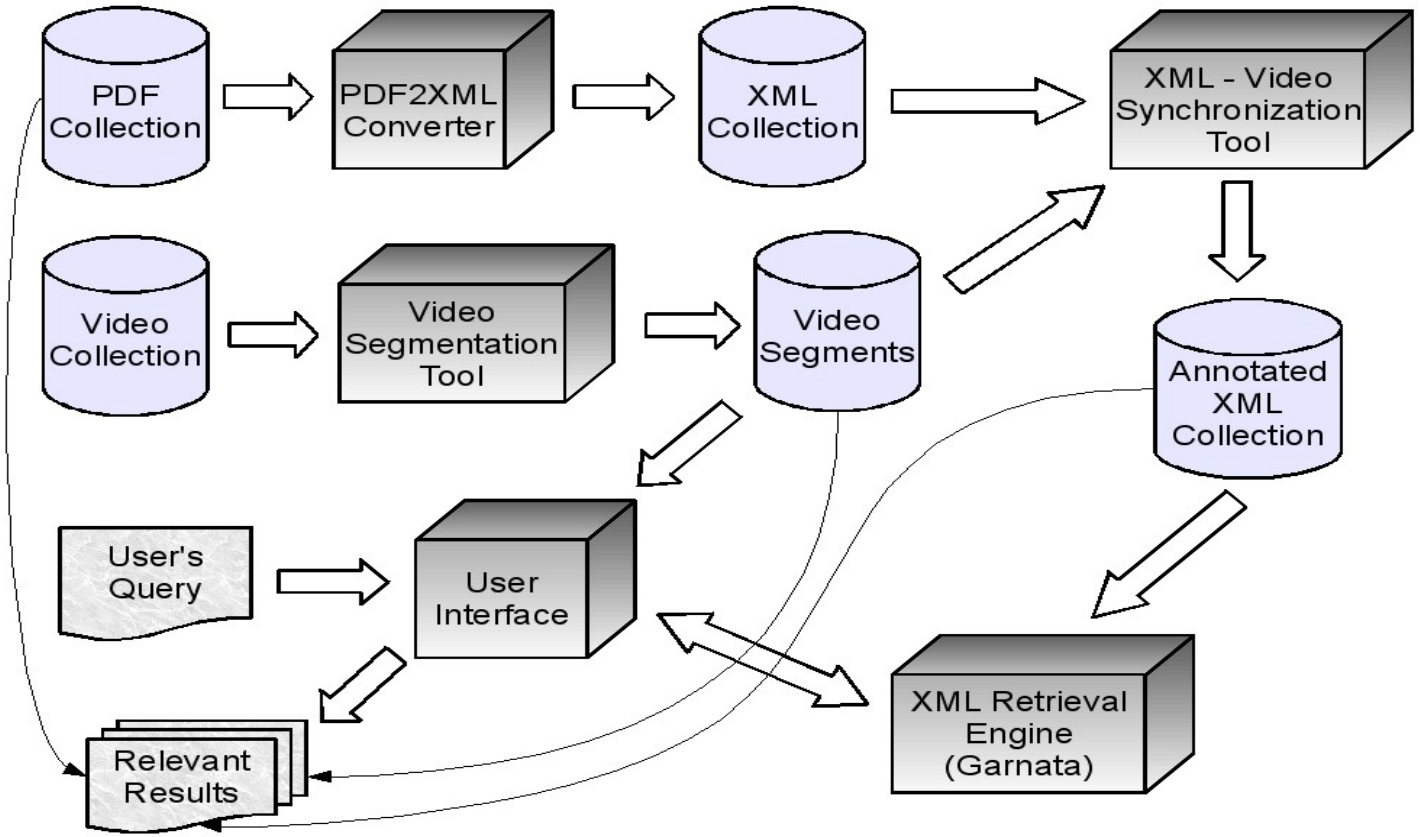
9. Un prototipo de recuperación XML: Seda

Antecedentes:

- Participación en INEX desde 2007 enviando resultados.
- Interés de aplicar Garnata a un entorno real.
- Contactos con el Parlamento de Andalucía.
- Proyecto de Excelencia de la Junta de Andalucia: ”Desarrollo de un sistema inteligente para el acceso a las colecciones documentales del Parlamento de Andalucía” (http://irutai.ugr.es/WebParlamento/).
- Colaboración con el Servicio de Publicaciones Oficiales.
- Colección: Diarios de sesiones y Boletines Oficales del P.A.
- Resultado: ***Seda***

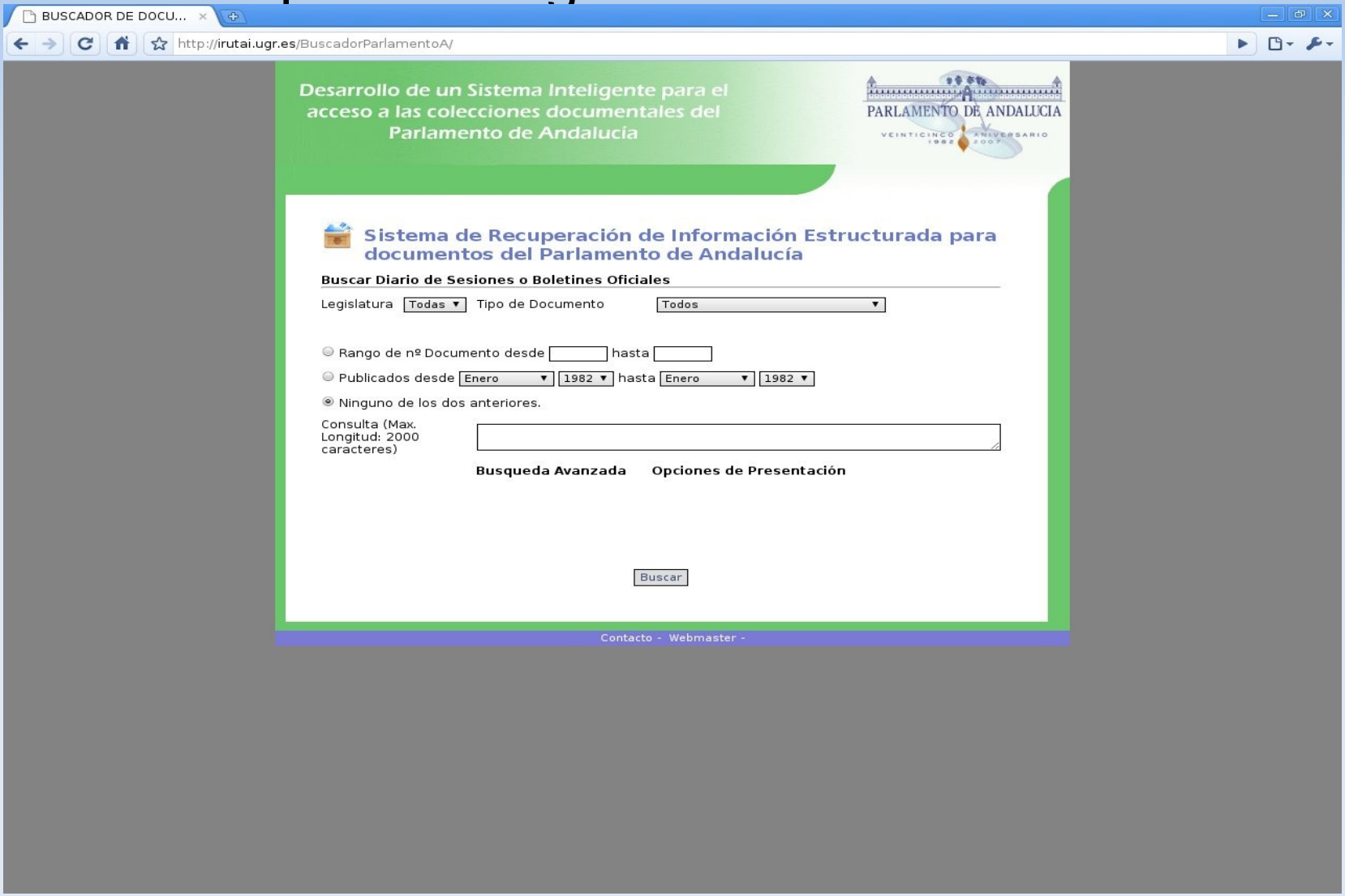
9. Un prototipo de recuperación XML: Seda

Arquitectura:



9. Un prototipo de recuperación XML: Seda

URL: http://irutai.ugr.es/BuscadorParlamentoA/



Bibliografía Básica

XML Retrieval

Mounia Lalmas

Morgan & Claypool Publishers

2009

Se acabo...

¡Gracias por vuestra atención!

:-)

¿Preguntas?