Indexing and Retrieving Structured Documents

Yves Chiaramella
IMAG
BP 43
38041 Grenoble Cedex
France
chiara@imag.fr - http://www-clips.imag.fr/

Structured Documents

● Introduction

» In standard IR, documents are considered as atomic information units whatever their type or size

● Indexed as a whole

» Indexes do not express the internal organisation of the discourse set by the author(s)

● Retrieved as a whole

» Users cannot retrieve independant components of documents that might be more adapted (more focused) to their information needs
Structured Documents

- Introduction
  - Fast development of tools about structured documents
    - Database systems,
    - New standards (SGML, XML, HTML, ODA…)
    - But also MPEG for video data

Questions:

What is the impact of structure on information retrieval?

If any, then what kinds of approaches to improve retrieval performances?

Plan

- Notion of Structured Document
- The Impact of Structure
- Approaches
  - Hypermedia
  - Passage Retrieval
  - Indexing / Retrieving Hierarchical Structures
- Conclusions
structured documents

- different facets

Differently structured documents (DSi, DSj, DSk) can be represented as:

```
  DSi
  /   \
 Si,j Si,k
  |    |   |
  im   txt   txt
```

- Document structure
  - Logical structure
  - Layout structure
  - Indexing structure

```
Document = logical structure + Layout structure + Indexing Structure
```

Browsing Links (internal, external) => HYPERMEDIA DOCUMENTS
The Impact of Structure

1. Information needs often involve structure

On top of content requirements, information needs often include requirements about:

- **Attributes**
  - eg. Novels written by Hemingway
  
  involved structural information: attributes type of doc, author

- **Links**
  - eg. A biography of Hemingway illustrated by pictures of the novelist.

  involved structural information: component link between text and image

2. Interaction & User tasks

- issue a query
- activate an anchor
The Impact of Structure

- Interaction & Cognitive effort

The Impact of Structure

- Querying (1)

System's response is a closed window on the document space
The Impact of Structure

- Querying (2): moving around the window

Cognitive effort:
- evaluating responses (R_i)
- properly reformulating query (Q_{i+1}) from R_i

Disorientation:
- length of responses
- bad ranking

The Impact of Structure

- Browsing

Cognitive effort:
- stacking paths and relevance judgements from previous steps

Disorientation:
- length of paths
- loops
- redundancy
- misleading paths
The Impact of Structure

- Combining Querying and Browsing

Cognitive effort
- browsing = incremental, try and error, easier process
- may help query reformulation

Disorientation
- querying effective for topic relocation in the document space

The Impact of Structure

- Conclusion:
  - Querying and Browsing have complementary advantages & limitations
  - Both are based on explicit manipulation of structure (though at different levels and for different purposes):
    - querying: attributes, logical structure
    - browsing: links
  - From the single point of view of interaction, there is a need to make a proper use of structure
**The Impact of Structure**

- 3. Querying structured documents

  ![Diagram]

  - Negative impact of ignoring the underlying structure
    - Need for complementary browse for accessing relevant components
    - Increased cognitive load,
    - Time consuming
    - Increased cost

**The Impact of Structure**

- Querying - focusing on relevant components

  ![Diagram]

  - Notion of Document Specificity to the query
    - Need for focusing on relevant components
    - Limited cognitive load,
    - Limited time
    - Reduced cost
The Impact of Structure

● 4. The impact on document ranking

- \( S_i \) size of the whole (atomic) document
- term frequencies related to document size (concentration effect)

\[
\begin{align*}
    tf_{3i} &= f_j/S_3 \\
    tf_{2i} &= f_j/S_2 \\
    tf_{1i} &= f_j/S_1
\end{align*}
\]

- Larger documents tend to be low-ranked
- Bad Incidence on recall / precision
The Impact of Structure

● The impact on document ranking

\[ \text{tf}_{3i} = \frac{f_i}{S_3} \]
\[ \text{tf}_{2i} = \frac{f_i}{S_2} \]
\[ \text{tf}_{1i} = \frac{f_i}{S_1} \]

- similar components tend to be closely ranked (whatever the embedding document)
- Good Incidence on recall / precision

The Impact of Structure

● 5. Disorientation Problems: the example of Web pages

Logical structure (abstract)

Ex: a is the homepage of a Web site

Links implement navigation structure, not the logical structure
- page nodes are indexed separately
- index of page a does not represent the content of subtree a (only of page a)
The Impact of Structure

- Disorientation Problems: the example of Web pages

Logical structure (abstract)  System response (linear list)

- Logical structure not visible
- redundant accesses (cognitive load, disorientation)
- the user has to re-engineer The structure to some extent

System ranking

Approaches - An Integrated Model

- Browsing/Querying: integration
  - Extended IR model (hypermedia features):
    » considering the document structure
  - Extended Hypermedia Model:
    » content management
    » typed links
    » weighted links
    » link construction

content and structural knowledge (conceptual graphs)
**Approaches - An Integrated Model (hypermedia)**

- **Structure**
  - HyperIndex
  - Lattice of Concepts

- **Beam Down**
  - HyperIndex
  - Lattice of Concepts
  - Hyperbase
Approaches - An Integrated Model (hypermedia)

- Beam Up

HyperIndex
Lattice of Concepts

Hyperbase

Approaches - An Integrated Model (hypermedia)

- Abstraction levels

Hyperbase

classes

hyperdocuments

Atomic objects
**Approaches - An Integrated Model (hypermedia)**

**Conclusion**

- Strong impact of Structure on Interaction Performances (cognitive load, disorientation, efficiency)

- Strong impact of Structure on Retrieval Performances (effectiveness)

- Integration (more than combination) of Browsing and Querying capabilities needed

---

**Approaches - Indexing Structured Documents**

**2. Passage Retrieval**

![Linear sequence of text diagram]

- DSi
- Si,j
- Si,k
- txt
- p1, p2, p3, p4, p5
Approaches - Indexing Structured Documents

- Passage Retrieval - segmenting passages
  - Computing local term density within a sliding window
  - Discontinuities denote topicality changes and passage boundaries

![Diagram of passage retrieval](attachment:image)

Advantages:
- Simple, efficient, can be effective

Limitations:
- Text only
- Incidence of document types
  - Needs to be tuned to document types
3. Retrieving Structured Information

- Focus on logical structure and links
- Corpus = \{ structural entities \}
- Each unit indexed / retrieved independantly

An important case: hierarchical structure (textual documents, Web sites presentations, video..)

- Non atomic view of Document = \{ Structural Units \}

Relationship between structure - semantic content: individual indexing of structural units
Approaches - Indexing Hierarchical Structures

- Relationship between Structure and Content (indexing)
  - Aggregation (propagation of content)
  - Integration of various content models related to specific media: (text, image, speech..)

\[ C_6 = f(C_{13}, C_{14}) \]
\[ C_7 = f(C_{15}, C_{16}) \]
\[ C_3 = f(C_6, C_7) \]

Approaches - Indexing Hierarchical Structures

- Indexing Structured documents (Eg. IOTA -RIME)
  - Structural Units vs Indexing Units

<table>
<thead>
<tr>
<th>Logical Structure (TYPE ST)</th>
<th>Index levels (TYPEI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>Maximal indexing level = Chapter</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Minimal indexing level = Section</td>
</tr>
<tr>
<td>Subsection</td>
<td></td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
<tr>
<td>Atomic object</td>
<td></td>
</tr>
</tbody>
</table>
Approaches - Indexing Hierarchical Structures

- Propagation of values: Attribute Classes

  - **Goal**: Inferring information related to document components (content, attributes)
  - **Approach**: propagation of attribute values within the logical structure

\[
\begin{align*}
C_6 &= f(C_{13}, C_{14}) \\
C_7 &= f(C_{15}, C_{16}) \\
C_3 &= f(C_6, C_7)
\end{align*}
\]

Three attribute classes:
- Dynamic Descending Attributes
- Dynamic Ascending Attributes
- Static Attributes

>> Descending Dynamic Attributes

- \( V_{a,d} = V \Rightarrow V_{a,iO} = f(V) \)

- Ex: Publication date (\( f \) is identity)
**Approaches - Indexing Hierarchical Structures**

- **Ascending Dynamic Attributes**
  - \( V_{a,i} = V_1, \ldots, V_{a,n} = V_n \Rightarrow V_{a,i} = \otimes (V_1, \ldots, V_n) \)
  - Ex: Author, Content attributes

- **Static Attributes**
  - Ex: Titles
**Approaches - Indexing Hierarchical Structures**

- **Content Attribute**
  
  - Computation and assignation of values to the symbolic attribute (expressions of language $\text{Symbolic}$)

  » Indexing Units: structural components of type:

  $$\text{TYPE}: \subseteq \text{TYPE}_{sr}$$

  » Ascending Propagation and Aggregation of index expressions:

    aggregation operator $\Theta_{\text{Symbolic}}$

- **Indexing Multimedia Data based on attribute values**:

  - multivalued

  - domain values are expressions of a given language $<\text{facet}>$ $<\text{media}>$

  - Used for integrating single-media models in the framework of structured, multimedia documents:

  » Types of Content Attributes (facets):

    - Physical
    - Symbolic
    - Structural
    - Spatial
    - Perceptive

  correspond to views of single-media data
Approaches - Indexing Hierarchical Structures

- Combining Content Attributes for different media

<table>
<thead>
<tr>
<th>Media</th>
<th>Views</th>
<th>Domains</th>
<th>Languages</th>
<th>Content Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>physical structural</td>
<td>1 symbolic</td>
<td>text</td>
<td>symbolic</td>
</tr>
<tr>
<td></td>
<td>symbolic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>physical structural</td>
<td>1 symbolic</td>
<td>image</td>
<td>symbolic</td>
</tr>
<tr>
<td></td>
<td>symbolic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>spatial perceptive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphic</td>
<td>physical structural</td>
<td>1 symbolic</td>
<td>graphic</td>
<td>symbolic</td>
</tr>
<tr>
<td></td>
<td>symbolic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>spatial perceptive</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Indexing Structured Documents

  - Content Attribute: Dynamic, Ascending
  - operator $\oplus_{\text{symbolic}}$ ascending aggregation of values of content attribute for each indexing unit:

  Implication between index expressions (generalisation)

  Aggregation of index expression (specialisation)
Approaches - Indexing Hierarchical Structures

- Properties of operator \( \oplus_{\text{symbolic}} \)
  - Neutral element \( \varepsilon : g \oplus \varepsilon = g \)
  - Reflexivity \( : g \oplus g = g \)
  - Symmetry \( : g \oplus f = f \oplus g \)
  - Associativity \( : (g \oplus f) \oplus h = g \oplus (f \oplus h) \)

- Determine a strategy for aggregating values
- Determine a strategy for dynamic indexing

Example: set indexing - \( \oplus_{\text{symbolic}} = \{ \text{index terms} \} \)

ascending strategy (specialisation of index expressions)

\( \oplus_{\text{symbolic}} \) is the union operator

Could be extended to the aggregation of weighted terms

\( (a, w_a) \oplus (b, w_b) = \{(a, f(w_a)), (b, f(w_b))\} \)
Approaches - Indexing Hierarchical Structures

● Example: boolean indexing
  ascending strategy (specialisation of index expressions)

⊕_{\text{symbolic}} \text{ is the and operator}

\begin{align*}
D &= a \land b \land c \land d \land e \land f \land g \\
&= a \land b \\
&= c \land d \land e \\
&= f \land g
\end{align*}

Implication
Of index expressions
(generalisation)

Aggregation
of index expressions
(specialisation)

Approaches - Retrieving Hierarchical Structures

● What strategy for retrieving indexed units ??
  - A) Brute approach:
    » Example: boolean indexing
      
      direct application of \( D \supset Q \), where \( D \) are indexing units

- Non optimal, redundant answer
**Approaches - Retrieving Hierarchical Structures**

- What strategy for retrieving indexed units ??
  - B) Soft approach :
    » Example : boolean indexing

- Selection of indexing units based on evaluation of
  \[ D_i \supset Q \] (exhaustivity)
  and on
  \[ Q \supset D_i \] (specificity)

- If both conditions are matched, \( D \) is an exact match for \( Q \)
- Else : the process looks for the smallest units (ie. deepest in the hierarchy) such that:
  \[ D_i \supset Q \] (exhaustivity requirement)
  and
  \[ \neg (Q \supset D_i) \] (specificity limitation)

**Approaches - Retrieving Hierarchical Structures**

- **Example** : Fetch &Browse algorithm

  - **Fetch**
    
    "horizontal" **preselection** of documents \( D \) satisfying \( D \supset Q \):

    ![Diagram](diagram.png)
**Approaches - Retrieving Hierarchical Structures**

- **Browse:**
  - "vertical" selection of most specific units within preselected documents (fetch):
    - Recursive Case: \( D_i \sqsupset Q \) and \( \neg(Q \sqsupset D_i) \)
    - Stop Case: \( \{(D_i \sqsupset Q \text{ and } Q \sqsupset D_i) \Rightarrow \text{result} = D_i\} \)
    - or \( \neg(D_i \sqsupset Q) \Rightarrow \text{result} = \text{Father}(D_i)\)

**Conclusions & Perspectives**

- Considering (ie. making explicit) structure leads to better retrieval performances in terms of interactive characteristics and classical retrieval performances (precision, recall)

- Considering structure may allow for the integration of browsing vs querying as complementary ways for retrieving information

- Considering structure may allow for the integration of various media in a unified indexing/retrieval strategy

- Considering structure may help in improving focus / precision, a much needed improvement for searching the Web

- Considering structure implies better understanding of the core notions of document, user needs, document relevance