Modern Information Retrieval

Chapter 6

Document and Query Properties and Languages

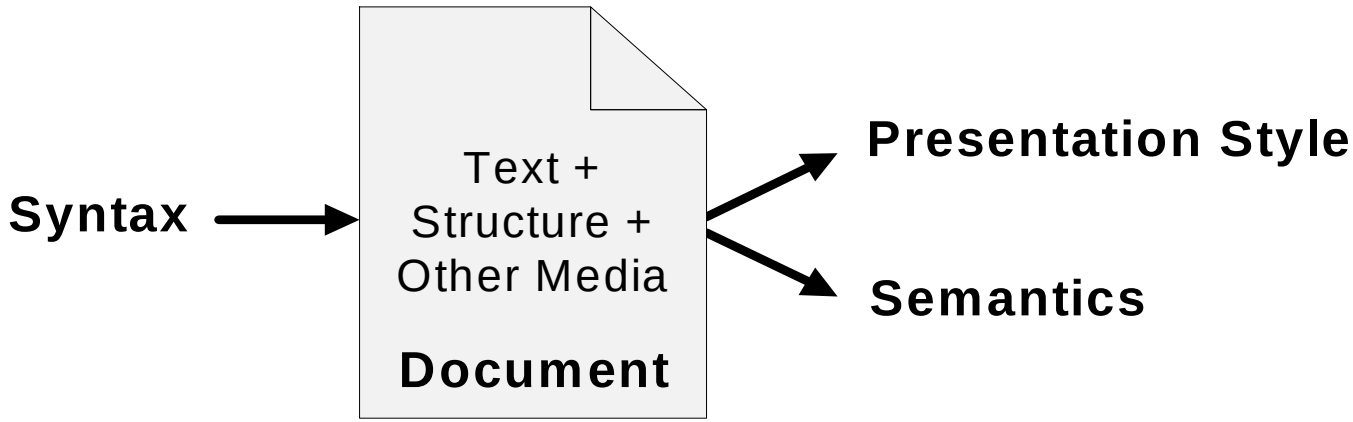
Metadata
Text Properties
Document Formats
Markup Languages
Query Properties
Query Languages
Trends and Research Issues

Introduction

- The term **document** is used to denote a single unit of information
- A document has a given **syntax and structure**, which is usually dictated by the application or by the person who created it
- A document also may have a **presentation style** associated with it, which specifies how it should be displayed or printed
  - Such style is usually given by the document syntax and structure and related to a specific application
- It also has a **semantics**, specified by the author of the document

Introduction

- Figure below depicts all cited relations



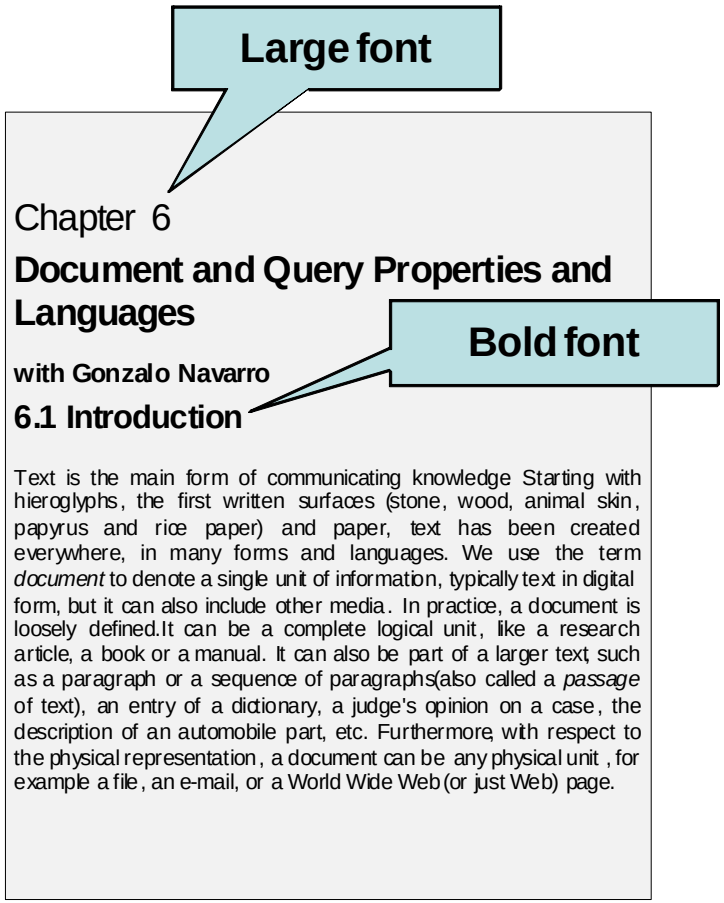
- The syntax of a document can express structure, presentation style, semantics, or even external actions
  - In many cases one or more of these elements are implicit or are given together
  - For example, a structural element (e.g. a section) can have a fixed formatting style

Introduction

- The syntax of a document can be **implicit** in its content, or expressed in a **simple declarative language**, or even in a **programming language**
  - Many syntax languages are proprietary and specific
  - Open and generic languages are more flexible
- Text can also be written in **natural language**
- However, the semantics of natural language is still not easy to understand by a computer
  - The current trend is to use languages which provide information on the document structure, format and semantics while being readable by humans as well as computers

Introduction

- The **style of a document** defines how the document is visualized in a computer window or a printed page
- Most documents have a particular formatting style
- The presentation style can be embedded in the document, as in TeX or Rich Text Format (RTF)
- Style can be complemented by macros
  - For example, LaTeX in the case of TeX



Introduction

- **Queries in search engines** can also be considered as short pieces of text
- The characteristics of queries do differ from normal text, and understanding them is very important
- The query semantic is many times ambiguous due to polysemy, and hence it is not so simple to infer the user intent behind a query

Metadata

Metadata

- Most documents and text collections have associated with them what is known as **metadata**
- Metadata is information on the organization of the data, the various data domains, and the relationship between them
- In short, metadata is **data about the data**
  - For instance, in a database management system, the name of the relations, the fields or attributes of each relation, the domain of each attribute are metadatas

Descriptive Metadata

- Common forms of metadata include the author of the text, the date of publication, the source of the publication, the document length, etc
- Dublin Core Metadata Element Set proposes 15 fields to describe a document
- Marchionini refers this type of information as **Descriptive Metadata**
- Descriptive Metadatas are external to the meaning of the document, and pertains more to how it was created

Semantic Metadata

- **Semantic Metadata** characterizes the subject matter that can be found within the document's contents
- Semantic Metadata is associated with a wide number of documents and its availability is increasing
- An important metadata format is the **Machine Readable Cataloging Record (MARC)** which is the most used format for library records
  - MARC has several fields for the different attributes of a bibliographic entry such as title, author, etc
  - In the U.S.A., a particular version of MARC is used: **USMARC**

Metadata in Web Documents

- With the increase of data in the Web, there are many initiatives to add metadata information to Web documents
- In the Web, metadata can be used for many purposes
  - Some of them are cataloging, content rating, intellectual property rights, digital signatures, privacy levels, applications to electronic commerce, etc
- The new standard for Web metadata is the **Resource Description Framework (RDF)**
  - This framework allows to describe Web resources to facilitate automated processing of the information

Metadata in Web Documents

- RDF does not assume any particular application or semantic domain
- It consists of a description of nodes and attached **attribute/value pairs**
  - Nodes can be any Web resource, that is, any **Uniform Resource Identifier (URI)**, which includes the **Uniform Resource Locator (URL)**
  - Attributes are properties of nodes, and their values are text strings or other nodes (Web resources or metadata instances)

Text Properties

Information Theory

Information Theory

- It is difficult to formally capture **how much information** is there in a given text
- However, the distribution of symbols is related to it
  - For example, a text where one symbol appears almost all the time does not convey much information
  - Information theory defines a special concept, **entropy**, to capture information content

Entropy

- If the alphabet has σ symbols, each one appearing with probability p_i in a text, the entropy of this text is defined as

$$E = -\sum_{i=1}^{\sigma} p_i \log_2 p_i$$

- We say that the amount of information in a text can be quantified by its entropy
- Entropy is also a limit on how much a text can be compressed

Text Properties

Modeling Natural Language

Modeling Natural Language

- We can divide the symbols of a text in two disjoint subsets:
  - symbols that separate words; and
  - symbols that belong to words
- It is well known that symbols are not uniformly distributed in a text
  - For instance, in English, the vowels are usually more frequent than most consonants

Modeling Natural Language

- A simple model to generate text is the **Binomial model**
- However, the probability of a symbol depends on previous symbols
  - For example, in English, a letter **f** cannot appear after a letter **c**
- We can use a finite-context or **Markovian model** to reflect this dependency
- More complex models include finite-state models, and grammar models
- However, finding the right grammar for natural language is still a difficult open problem

Modeling Natural Language

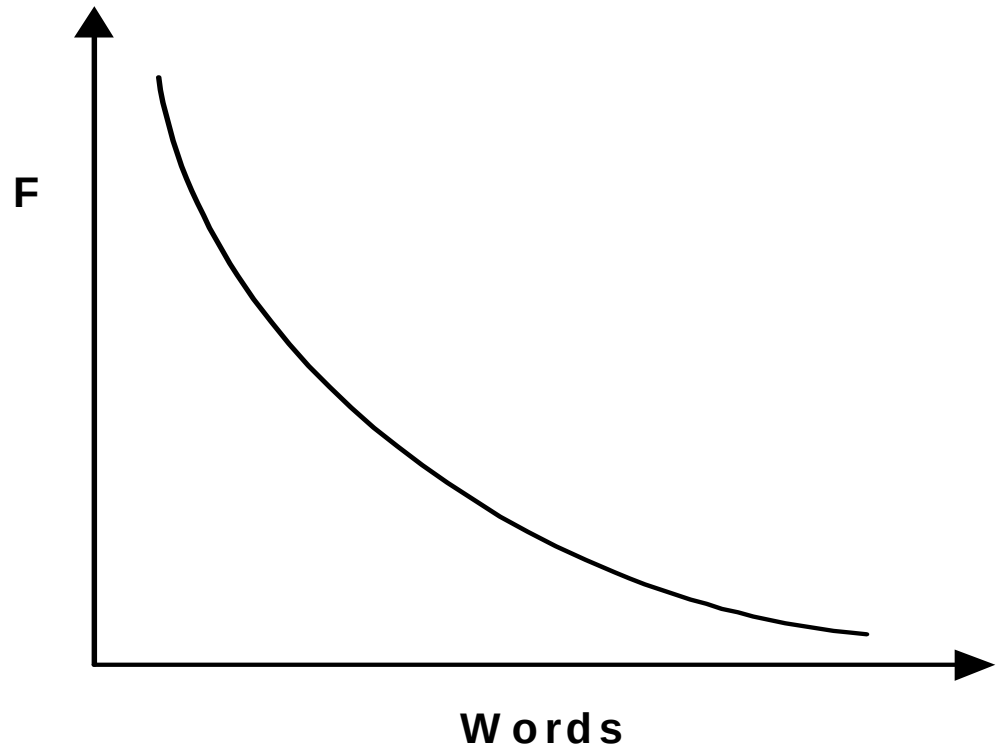
- The second issue is **how the different words are distributed** inside each document
- An approximate model is the **Zipf's Law**
- This law states that in a text of n words with a vocabulary of V words, the i-th most frequent word appears $n/(i^{\theta} H_V(\theta))$ times
- $H_V(\theta)$ is the harmonic number of order θ of V, defined as

$$H_V(\theta) = \sum_{j=1}^{V} \frac{1}{j^{\theta}}$$

  - The value of θ depends on the text

Modeling Natural Language

- The Figure below illustrates the distribution of frequencies of the terms in a text
- The words are arranged in decreasing order of their frequencies



Modeling Natural Language

- Since the distribution of words is very skewed, words that are too frequent, such as **stopwords**, can be disregarded
- A stopword is a word which does not carry meaning in natural language
  - Examples of stopword in english: *a*, *the*, *by*, *and*
- Fortunately the most frequent words are stopwords
  - Therefore, half of the words appearing in a text do not need to be considered

Modeling Natural Language

- A third issue is the **distribution of words** in the documents of a collection
- A simple model is to consider that each word appears the same number of times in every document (not true in practice)
- A better model is to consider a negative binomial distribution, which says that the fraction of documents containing a word k times is

$$F(k) = \binom{\alpha + k - 1}{k} p^k (1 + p)^{-\alpha - k}$$

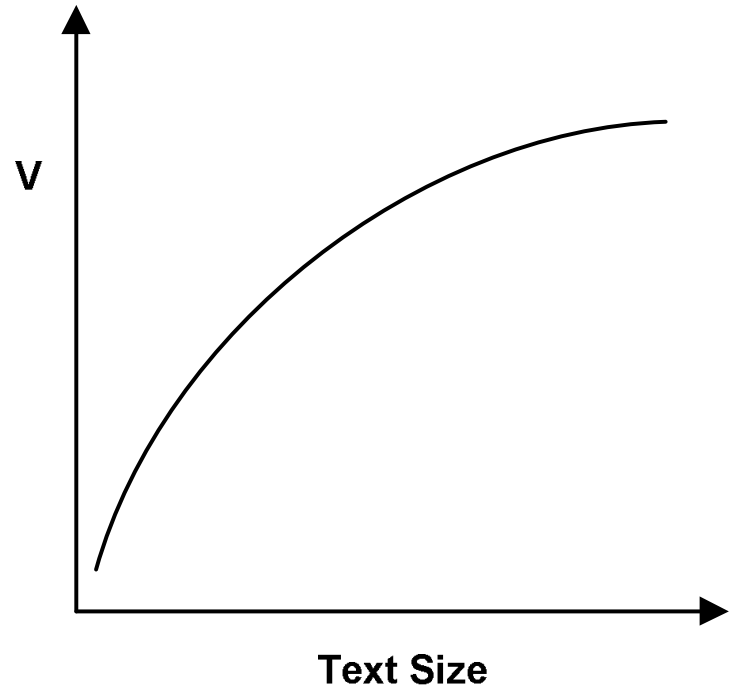
where p and α are parameters that depend on the word and the document collection

Modeling Natural Language

- The fourth issue is the **number of distinct words** in a document (the **document vocabulary**)
- To predict the growth of the vocabulary size in natural language text, we use the so called **Heaps' Law**
- This law states that the vocabulary of a text of n words is of size $V = Kn^{\beta} = O(n^{\beta})$, where K and β depend on the text
  - K is normally between 10 and 100, and β is a positive value less than one
- Some experiments on the TREC-2 collection show that the most common values for β are between 0.4 and 0.6.

Modeling Natural Language

- The figure below illustrates how the vocabulary size varies with the text size



- Hence, the vocabulary of a text grows sub-linearly with the text size, in a proportion close to its square root

Modeling Natural Language

- Notice that the set of different words of a language is fixed by a constant
- However, the limit is so high that it is common assume that the size of the vocabulary is $O(n^{\beta})$ instead of $O(1)$
- Many authors argue that the number keeps growing anyway because of the typing or spelling errors
- The Heap's law also applies to collections of documents because, as the total text size grows, the predictions of the model become more accurate
- Furthermore, this model is also valid for the World Wide Web

Modeling Natural Language

- A last issue is the **average length of words**
- This relates the text size in words with the text size in bytes
- In the different sub-collections of TREC-2 collection, the average word length is very close to 5 letters
- If we remove the stopwords, the average length of a word increases to a number between 6 and 7 (letters)

Text Properties

Similarity Models

Similarity Models

- Similarity is measured by a **distance function**
- If we have strings of the same length, we can define the distance between them as the number of positions that have different characters
  - For instance, the distance is 0 if they are equal
  - This is called the **Hamming distance**
- A distance function should also be **symmetric**
  - In this case, the order of the arguments does not matter
  - This distance satisfy the triangle inequality:

$$distance(a, c) \leq distance(a, b) + distance(b, c)$$

Similarity Models

- An important distance over strings is the **Edit or Levenshtein distance**
- Edit distance: the minimal number of characters insertions, deletions, and substitutions that we need to perform in any of the strings to make them equal
- For instance:
  - The edit distance between `color` and `colour` is one
  - The edit distance between `survey` and `surgery` is two
- Extensions to this concept: different weights for each operation, adding transpositions, etc

Similarity Models

- Another measure is the **longest common subsequence** (LCS)
- After all non-common characters have been deleted of two (or more) strings, the remaining sequence of characters is the LCS of both strings
- For example, the LCS of `survey` and `surgery` is `surey`

```
sur ve y
surg ery
---------
sur  e y
```

Similarity Models

- Similarity can be extended to documents
- For example, we can consider lines as single symbols and compute the longest common sequence of lines between two files
- This is the measure used by the `diff` command in Unix-like operating systems
- Problems with this approach
  - Very time consuming
  - Not consider lines that are similar
- The later drawback can be fixed by taking a weighted edit distance between lines

Document Formats
Text

Text

- With the advent of the computer, it was necessary to **code characters** in binary digits
  - Examples of Coding schemes: EBCDIC (7 bits), ASCII (8 bits) and UNICODE (16 bits)
- Beyond characters, there is no single standard **format for a text document**
- An IR system should be able to retrieve information from many text formats (doc, pdf, html, txt, etc)
- Current IR systems have filters that can handle most popular documents
  - Even though, good filters might not be possible if the format is proprietary

Text

- Other text formats were developed for document interchange
  - Like **Rich Text Format (RTF)**
- Other important formats were developed for displaying or printing documents
  - Like **Portable Document Format (PDF)** and **Postscript**
- Other interchange formats are used to encode electronic mail
  - Like **Multipurpose Internet Mail Exchange (MIME)**
- Nowadays many files are compressed
  - They include **Compress** (Unix), **ARJ** (PCs) and **ZIP** (for example `gzip` in Unix and `Winzip` in Windows)

Document Formats
Multimedia

Multimedia

- Multimedia usually stands for applications that handle different types of digital data originated from distinct types of media
- The most common types of media in multimedia applications are text, sound, images, and video
- Different types of formats are necessary for storing each type of media
- In contrast with text formats, most formats for multimedia are partially binary and hence can only be processed by a computer

Images

- There are several formats for images
- The simplest formats are direct representations of a bit-mapped display such as XBM, BMP or PCX
- Images of these formats have a lot of redundancy and can be compressed efficiently
  - Example of format that incorporate compression: **Compuserve's Graphic Interchange Format (GIF)**

Images

- To improve compression ratios for higher resolutions, lossy compression was developed
  - That is, uncompressing a compressed image does not give the original
- This is done by the **Joint Photographic Experts Group (JPEG)** format
  - JPEG tries to eliminate parts of the image that have less impact in the human eye
  - This format is parametric, in the sense that the loss can be tuned

Images

- Another common image format is the **Tagged Image File Format** (TIFF)
  - This format is used to exchange documents between different applications and different computer platforms
  - TIFF has fields for metadata and also supports compression as well as different number of colors
- Yet another format is **Truevision Targa image file (TGA)**, which is associated with video game boards
- There are many more image formats, many of them associated to particular applications ranging from fax to fingerprints and satellite images

Audio

- Audio must be digitalized first in order to be stored properly
- The most common formats for small pieces of digital audio are **AU**, **MIDI** and **WAVE**
- MIDI is an standard format to interchange music between electronic instruments and computers
- For audio libraries other formats are used such as **RealAudio** or **CD formats**

Movies

- There are several formats for animations or moving images
- The main one is **Moving Pictures Expert Group (MPEG)** which is related to JPEG
  - MPEG works by coding the changes with respect to a base image which is given at fixed intervals
  - In this way, MPEG profits from the temporal image redundancy that any video has
  - This format also includes the audio signal associated to the video
- Other video formats are **AVI**, **FLI** and **QuickTime**
  - AVI may include compression (CinePac), as well as QuickTime, which was developed by Apple
  - As for MPEG, the audio is also included

3D Graphics

- There are many formats for three dimensional graphics
  - Examples of formats: **Computer Graphics Metafile (CGM)** and **Virtual Reality Modeling Language (VRML)**
- VRML is also intended to be a universal interchange format for integrated 3D graphics and multimedia
- It may be used in a variety of application areas such as
  - engineering and scientific visualization
  - multimedia presentations
  - entertainment and educational titles
  - web pages and shared virtual worlds
- VRML has become the *de facto* standard Modeling Language for the Web

Markup Languages

Markup Languages

- Markup is defined as extra textual syntax that can be used to describe formatting actions, structure information, text semantics, attributes, etc
- Examples of Markup Languages
  - **SGML**: Standard Generalized Markup Language
  - **XML**: eXtensible Markup Language
  - **HTML**: Hyper Text Markup Language
- All these languages and examples of them will be presented here

Markup Languages

SGML

SGML

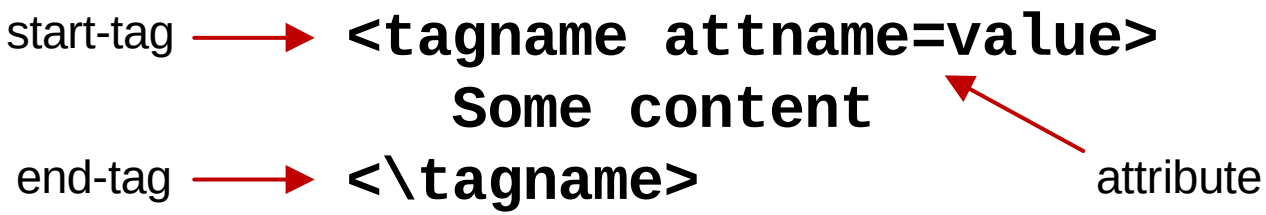
- SGML (ISO 8879) stands for **Standard Generalized Markup Language**
- It is a meta-language for tagging text (it provides rules for defining a markup language based on tags)
- Each instance of SGML includes a description of the document structure called **document type definition**
- Hence, an SGML document is defined by:
  1. a document type definition; and
  2. the text itself marked with tags which describe the structure

SGML

- The **document type definition** is used to
  - describe and name the pieces that a document is composed of
  - define how those pieces relate to each other
- Part of the definition can be specified by an **SGML document type declaration (DTD)**
- Other parts, such as the semantics of elements and attributes, or application conventions, cannot be expressed formally in SGML
  - Comments can be used, however, to express them informally
  - More complete information is usually present in separate documentation

SGML

- Tags are denoted by angle brackets



- They are used to identify the **beginning** and **ending** of elements of the document
- Ending tags are specified by adding a slash before the tag name
- The **attributes** are specified inside the beginning tag, after the tagname

SGML

- Example of a SGML DTD for electronic messages

```
<!ELEMENT e-mail                - - (prolog, contents) >
<!ELEMENT prolog                - - (sender, address+, subject?, Cc*) >
<!ELEMENT (sender | address | subject | Cc) - O (#PCDATA) >
<!ELEMENT contents              - - (par | image | audio)+ >
<!ELEMENT par                   - O (ref | #PCDATA)+ >
<!ELEMENT ref                   - O EMPTY >
<!ELEMENT (image | audio)       - -  (#NDATA) >

<!ATTLIST e-mail
          id            ID                      #REQUIRED
          date_sent     DATE                    #REQUIRED
          status        (secret | public )      public >
<!ATTLIST ref
          id            IDREF                   #REQUIRED >
<!ATTLIST (image | audio )
          id            ID                      #REQUIRED >
```

SGML

- Example of use of previous DTD

```
<!DOCTYPE e-mail SYSTEM "e-mail.dtd">
<e-mail id=94108rby date_sent=02101998>
  <prolog>
  <sender> Pablo Neruda </sender>
  <address> Federico García Lorca </address>
  <address> Ernest Hemingway </address>
  <subject> Pictures of my house in Isla Negra
  <Cc> Gabriel García Marquez </Cc>
  </prolog>
  <contents>
      <par>
         As promised in my previous letter...
      </par>
  </contents>
</e-mail>
```

SGML

- The document description does not specify how a document is printed on paper or displayed on a screen
- Therefore, output specifications are often added to SGML documents
- Output specification standards were devised:
  - **DSSSL**: Document Style Semantic Specification Language
  - **FOSI**: Formatted Output Specification Instance
- Both of these standards define mechanisms for associating style information with SGML document instances
- They are used for defining, for instance, that the data identified by a tag should be typeset in italics

SGML

- One important use of SGML is in the **Text Encoding Initiative (TEI)**
- The TEI includes several US associations related to the humanities and linguistics
- The main goal is to generate guidelines for the preparation and interchange of electronic texts for scholarly research, as well as the industry
- In addition to the guidelines, TEI provides several document formats through SGML DTDs
- One of the most used formats is **TEI Lite**

Markup Languages

HTML

HTML

- HTML stands for **HyperText Markup Language** and is an instance of SGML
- It was created in 1992 and has evolved during the last years, being 4.0 the latest version
- Most documents on the Web are stored and transmitted in HTML
- HTML is a simple language well suited for hypertext, multimedia, and the display of small and simple documents
- Although there is an HTML DTD, most HTML instances do not explicitly make reference to the DTD
- The HTML tags follow all the SGML conventions and also include formatting directives

HTML

- HTML documents can have other media embedded within them, such as images or audios
- HTML also has fields for **metadata**, which can be used for different applications and purposes
- If we also add programs (for example, using Javascript) inside a page some people call it **dynamic HTML**

HTML

- Example of an HTML document

```
<html><head>
<title>HTML Example</title>
<meta name=rby content="Just an example">
</head>
<body>
<h1>HTML Example</h1>
<p><hr><p>HTML has many <i>tags</i>, among them:
<ul>
<li> links to other <a href=example.html>pages</a> (a from anchor),
<li> paragraphs (p), headings (h1, h2, etc), font types (b, i),
<li> horizontal rules (hr), indented lists and items (ul, li),
<li> images (img), tables, forms, etc.
</ul>
<p><hr><p><img align=left src="at_work.gif">
This page is <b>always</b> under construction.
</body></html>
```

HTML

- How the HTML document is seen in a browser

HTML Example

HTML has many *tags*, among them:

- links to other pages (a from anchor),
- paragraphs (p), headings (h1, h2, etc), font types (b, i),
- horizontal rules (hr), indented lists and items (ul, li),
- images (img), tables, forms, etc.

This page is **always** under construction.

HTML

- Because HTML does not fix the presentation style of a document, in 1997, the **Cascade Style Sheets** (CSS) were introduced
- CSS offers a powerful and manageable way for authors to improve the aesthetics of HTML pages
- Style sheets separate information about presentation from document content
- On the other hand, CSS support in current browsers is still modest

HTML

- The evolution of HTML implies support for backward compatibility and also for forward compatibility
- HTML 4.0 has been specified in three flavors: strict, transitional, and frameset
  - **Strict HTML** only worries about non-presentational markup, leaving all the displaying information to CSS
  - **Transitional HTML** uses all the presentational features for pages that should be read for old browsers that do not understand CSS
  - **Frameset HTML** is used when you want to partition the browser window in two or more frames
- HTML 4.0 includes support for style sheets, internationalization, frames, richer tables and forms, and accessibility options for people with disabilities

HTML

- Typical HTML applications use a fixed small set of tags in conformance with a single SGML specification
- Fixing a small set of tags makes the language specification much easier to build applications
- But this advantage comes at the cost of severely limiting HTML in several important aspects
- In particular, HTML does not
  - allow users to specify their own tags
  - support the specification of nested structures needed to represent database schemas
  - support the kind of language specification that allows consuming applications to check data for structural validity on importation

Markup Languages

XML

XML

- XML stands for **eXtensible Markup Language** and is a simplified subset of SGML
- XML is not a markup language, as HTML is, but a meta-language, as SGML
- It allows to have human-readable semantic markup, which is also machine-readable
- XML makes it easier to develop and deploy new specific markup languages

XML

- XML does not have many of the restrictions imposed by HTML
- On the other hand, imposes a more rigid syntax on the markup:
  - In XML, ending tags cannot be omitted
  - XML also distinguishes upper and lower case
  - All attribute values must be between quotes
- Parsing XML without a DTD is easier
  - The tags can be obtained while the parsing is done

XML

- XML allows any user to define new tags, define more complex structures and data validation capabilities
- On other side, XML is a profile of SGML that eliminates many of the difficulties to implement things
- The **Extensible Style sheet Language (XSL)** is the XML counterpart of Cascading Style Sheets (CSS)
- XSL is designed to transform and style highly-structured, data-rich documents written in XML
  - For example, with XSL it would be possible to automatically extract a table of contents from a document
- The syntax of XSL has been defined using XML

XML

- Another extension to XML, defined using XML, is the **Extensible Linking Language (XLL)**
- XLL defines different types of links, including external and internal links
- Recent uses of XML include:
  - Mathematical Markup Language (MathML)
  - Synchronized Multimedia Integration Language (SMIL)
  - Resource Description Format
- The XML movement is one indication that a parseable, hierarchical object model will play an increasingly major role in the evolution of HTML
- The next generation HTML should be based in a suite of XML tag sets

Markup Languages

HyTime

HyTime

- The **Hypermedia/Time-based Structuring Language (HyTime)** is a standard (ISO/IEC 10744) defined for multimedia documents markup
- HyTime is an SGML architecture that specifies the generic hypermedia structure of documents
- The hypermedia concepts directly represented by HyTime include
  - complex locating of document objects
  - relationships (hyperlinks) between document objects
  - numeric, measured associations between document objects

HyTime

- The HyTime architecture has three parts:
  - the base **linking and addressing architecture**
  - the **scheduling architecture** (derived from the base architecture)
  - the **rendition architecture** (which is an application of the scheduling architecture)
- HyTime does not directly specify graphical interfaces, user navigation, user interaction, or the placement of media on time lines and screen displays
- These aspects of document processing are rendered from the HyTime constructs in a similar manner as style sheets in SGML documents

Query Properties

Characterizing Web Queries

- The notion of **information seeking** encompasses a broad range of **information needs**
- People have different search needs at different times and in different contexts
- A number of researchers have attempted to taxonymize and tally the types of information needs
- These efforts are summarized in the following

Characterizing Web Queries

- Most web search engines **record information about the queries** that searchers write
  - This information includes the **query** itself, the **time**, and the **IP address**
  - Some systems also record which **search results were clicked** on for a given query
- These logs are a valuable resource for understanding the kinds of information needs that users have
- Statistics on query length and composition have been recorded since early in the Web's existence

Characterizing Web Queries

- These logs have shown that the **average query length** has grown over time, and the **percentage of one-word queries** have shrunk
- Distribution of query lengths found by Jansen et al., using 1.5M queries gathered in May 2005, from the Dogpile.com search engine:

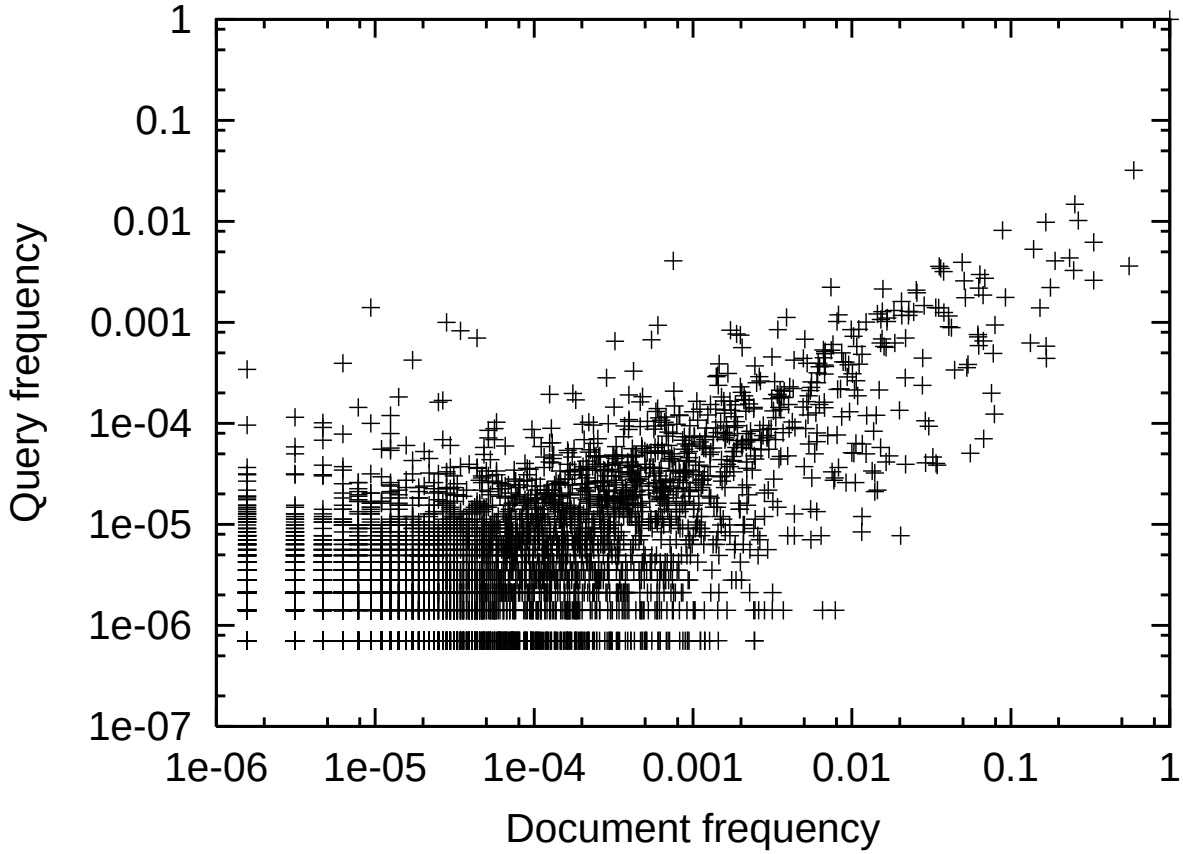
Length	Occurrences	Percent
1	281,000	18.5
2	491,000	32.3
3	373,000	24.5
4	194,000	12.7
5	95,000	6.3
6	45,000	3.0
7	22,000	1.5
8	12,000	0.8
9	6,000	0.4

Characterizing Web Queries

- Queries, as words in a text, follow a biased distribution
- In fact, the frequency of query words follow a **Zipf's law** with parameter α
  - The value of α ranges from 0.6 to 1.4, perhaps due to language and cultural differences
- However, this is less biased than Web text, where α is closer to 2
- The standard correlation among the frequency of a word in the Web pages and in the queries also varies
  - These values range from 0.15 to 0.42

Characterizing Web Queries

- This implies that **what people search is different from what people publish in the Web**
- Figure below shows this fact



Characterizing Web Queries

- Some logs also registers the **number of answer pages seen** and the **pages selected** after a search
- Many people refines the query adding and removing words, but most of them see very few answer pages
- The table below shows the comparison of four different search engines with different scopes and languages

Measure	AltaVista	Excite	AlltheWeb	TodoCL
Words per query	2.4	2.6	2.3	1.1
Queries per user	2.0	2.3	2.9	–
Answer pages per query	1.3	1.7	2.2	1.2
Boolean queries	$<$40%	10%	–	16%

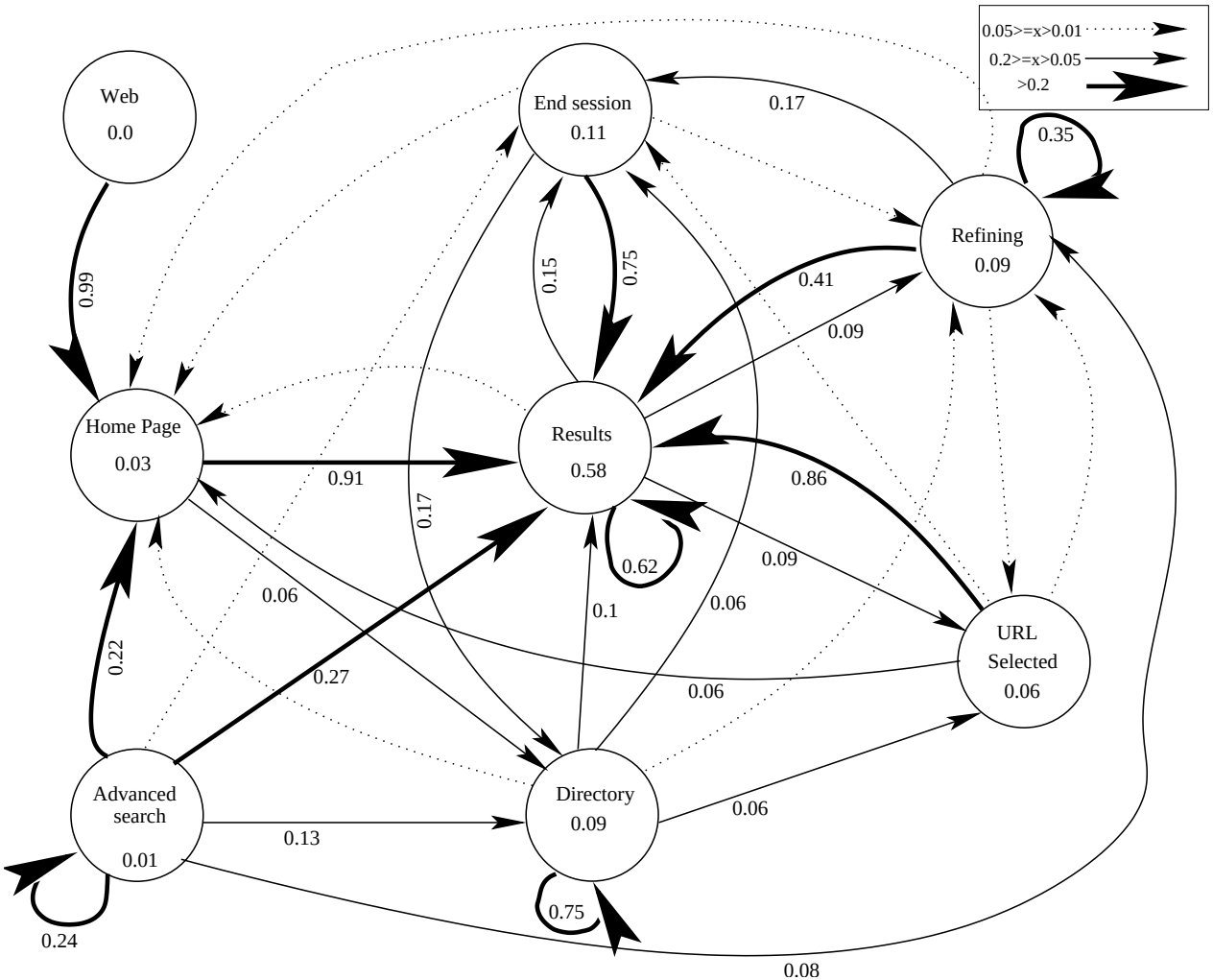
Characterizing Web Queries

- In addition, the average number of pages clicked per answer is very low (around 2 clicks per query)
- Further studies have showed that the focus of the queries has shifted from leisure to e-commerce

Query Properties User Search Behavior

User Search Behavior

- Figure below shows an example, where the transitions between different states, indicating the proportion of users that took that path



User Search Behavior

- The number in each state of the previous figure represents the probability of a user being in that state
- From the diagram we could infer that:
  - Advanced search is not used (but we must have it!)
  - Few people refines the query
  - Few people browse the directory
- This means that instead of posing a better query, a trial and error method is used

Query Intent

- Prior to the Web, search engine designers could assume that searchers had an informational goal in mind
- This was due in part to:
  - The search population (students, legal analysts, scientific researchers, business analysts)
  - And the kind of data that could be searched (newswire, legal cases, journal article abstracts)
- The queries to these systems were often long, or contained Boolean operators
- Users had to carefully craft their queries, because careless query formulation was expensive

Query Intent

- Broder created a taxonomy of web search goals
- The three types of **need behind the query** that he identified were:
  - **Navigational**: The immediate intent is to reach a particular site (24.5% survey, 20% query log)
  - **Informational**: The intent is to acquire some information (39% survey, 48% query log)
  - **Transactional**: The intent is to perform some web-mediated activity (36% survey, 30% query log)
- This taxonomy has been heavily influential in discussions of query types on the Web

Query Intent

- Rose & Levinson developed a taxonomy that differed somewhat from Broder's
- They noted that much of what happens on the Web is the acquisition and consumption of online resources
- Thus they replace Broder's transactions category with a broader category of **resources**
- They also introduced subcatgories for the three main categories, including the subcategory of **advice seeking**

Query Intent

- Rose & Levinson manually classified a set of 1,500 AltaVista search engine log queries
  - For two sets of 500 queries, the labeler saw just the query and the retrieved documents
  - For the third set the labeler also saw information about the clicks of the searcher
- Conclusions: using the extra information about clickthrough did not change the results
- They found a smaller proportion of navigational queries (13%) than did Broder (22%)
- They also find that informational queries were 61%, a much higher proportion than Broder's average (45%)

Query Intent

- Recent research have dealt with the automatic prediction of these classes
- These works use machine learning over different query attributes such as:
  - Anchor-text distribution of the words in the queries
  - Past click behavior
- The main problem is that many queries are **inherently ambiguous**
  - They can be classified in more than one class when the context of the search is not known

Query Topic

- Queries from web query logs can be classified according to the **topic of the query**, independent of the type of information need
- For example, a search involving the topic of weather can consist of:
  - the simple information need of looking at today's forecast, or
  - the rich and complex information need of studying meteorology

Query Topic

- Spink & Jansen et al. have manually analyzed samples of query logs
- In one article, they compared a manual analysis on Altavista logs from 1997 with queries from the DogPile metasearch engine in 2005
  - They found that queries relating to sex and pornography declined from 16.8% in 1997 to just 3.6% in 2005
  - In another article, they show that Commerce-related queries now dominate the query logs, claiming 30.4% in this study, up from 13.3% in 1997

Query Topic

- Topics manually assigned to 2,500 queries against the DogPile metasearch engine in 2005

Rank	Topic	Number	Percent
1	Commerce, travel, employment, or economy	761	30.4
2	People, places, or things	402	16.0
3	Unknown or other	331	13.2
4	Health or sciences	224	8.9
5	Entertainment or recreation	177	7.0
6	Computers or Internet	144	5.7
7	Education or humanities	141	5.6
8	Society, culture, ethnicity, or religion	119	4.7
9	Sex or pornography	97	3.8
10	Government or legal	90	3.6
11	Arts	14	0.5

Query Topic

- Shen et al. described an algorithm for automatically classifying web queries into a set of pre-defined topics
- The idea is to use the results of a query in a search engine, and look at the categories that have been manually associated with those results in the past
- Their results: F-score of about .45 on 63 categories
- The table below shows the results for five queries

Query	Top category	Second category
chat rooms	Computers/Internet	Online Community/Chat
lake michigan lodges	Info/Local & Regional	Living/Travel & Vacation
stephen hawking	Info/Science & Tech	Info/Arts & Humanities
dog shampoo	Shopping/Buying Guides	Living/Pets & Animals
text mining	Computers/Software	Information/Companies

Query Topic

- More recently, Broder et al. presented a highly accurate method for classifying short, rare queries into a taxonomy of 6,000 categories
  - This is an important problem because rare or infrequent queries are approximately half of all queries
- They trained a set of text classifiers using a commercial taxonomy contained many documents assigned to each category
- They classified the results of a query in the classifier, and then used a voting algorithm to determine which class(es) best categorize the query

Query Ambiguity

- Ambiguous queries are those queries that can have two or more distinct meanings
  - a query on `apple` may refer to the fruit or the computer manufacturer or the record label
- A number of search interface ideas turn out to be effective mainly for ambiguous queries
- For this reason, a few researchers have tried to estimate what proportion of queries truly are ambiguous

Query Ambiguity

- Wen et al. found that identical query terms produced nearly identical clicks
  - They speculated that users were self-disambiguating by their choice of terms
  - Song et al. describe an algorithm for estimating how many queries in a query log are ambiguous
- They achieved 87% accuracy on a test set of 253 queries
- Applying this algorithm to a sample, they estimated that the sample contained 16% ambiguous queries

Query Session Boundaries

- To use query log data, it is important to distinguish the **boundaries of search sessions**
- A session is defined as a sequence of requests made by a single user for a single navigation purpose
- The simplest method is to set a **time threshold**
- He & Goker find a range of 10 to 15 minutes inactivity is optimal for creating session boundaries for web logs
- Silverstein et al. use five minutes, Anick [**?**] uses 60 minutes, and Catledge & Pitkow recommend 25.5 minutes

Query Session Boundaries

- Chen et al. recognize that timing information varies with both user and task, and so propose an adaptive timeout method
- They define the **age of an action** to be the interval between an action and the current time
- A session boundary is assigned when the age of an action is two times older than the average age of actions in the current session
- Thus the time to elapse before a session boundary is declared varies with the degree of activity the user is engaged in at different points in time

Query Session Boundaries

- Jansen et al. examined 2,465,145 interactions from 534,507 users of a search engine
- They found that the best method for determining session duration made use of **IP address**, a **cookie**, and **query reformulation patterns**
- Using this method, they found that 93% of sessions consisted of 3 or fewer queries, with a mean of 2.31 queries per session
- The mean session length using this method was 5 minutes and 15 seconds

Query Re-access Patterns

- Many searches are characterized by people **re-accessing information** that they have seen in the past
- This can be accomplished by saving previously visited information via bookmarks
- However, there is ample evidence that people use search engines as re-finding instruments
- Teevan et al. found that 40% of the queries led to a click on a result that the same user had clicked on in a past search session

Query Properties Classifying Search Behaviors

Classifying Search Behaviors

- O'Day and Jeffries studied 15 business analysts who worked with search intermediaries (professional searchers) to access information important to their work
- They observed three main kinds of information seeking tasks:
  - monitoring a well-known topic over time (such as researching competitors' activities each quarter)
  - following a plan or stereotyped series of searches to achieve a particular goal (such as keeping up to date on good business practices)
  - exploring a topic in an undirected fashion (such as when getting to know an unfamiliar industry)

Classifying Search Behaviors

- Kellar et al. made a study about the categories of information seeking tasks:
  - **Fact Finding**: Looking for specific facts or pieces of information; usually short lived tasks, completed over a single session
  - **Information Gathering**: A task that involves the collection of information, often from multiple sources
  - **Browsing**: A task where web pages are visited with no particular goal other than entertainment or *to see what's new*
  - **Transactions**: Online actions.
  - **Other**: Other activities, such as web page maintenance.

Classifying Search Behaviors

- The table below shows the Web page task usage statistics obtained from the study by Kellar et al.

Task	% of total Web use	% that were repeats
Fact finding	18.3	55.5
Info. Gathering	13.5	58.5
Browsing	19.9	84.4
Transactions	46.7	95.2
Other	1.7	–

Query Languages

Query Languages

- We cover now the **different kind of queries** normally posed to text retrieval systems
- This is in part dependent on the retrieval model the system adopts
  - That is, a full-text system will not answer the same kind of queries as those answered by a system based on keyword ranking or on a hypertext model

Query Languages

- There is a difference between **information retrieval** and **data retrieval**
- Languages for information retrieval allow the answer to be ranked
  - For the basic information retrieval models, keyword-based retrieval is the main type of querying task
- For query languages not aimed at information retrieval, the concept of ranking cannot be easily defined
  - We consider these languages as languages for data retrieval
  - Some query languages are not intended for final users

Query Languages

- Most query languages try to use the **content** and the **structure of the text** to find relevant documents
- In that sense, the system may fail to find the relevant answers
- For this reason, a number of techniques meant to enhance the usefulness of the queries exist
- Some examples are the expansion of a word to the set of its synonyms or the use of a thesaurus
- Some words which are very frequent and do not carry meaning (called ***stop-words***) may be removed
- When we want to emphasize the difference between words that can be retrieved by a query and those which cannot, we call **keywords** the first ones

Query Languages

- Another issue is the subject of the **retrieval unit** the information system adopts
- The retrieval unit is the basic element which can be retrieved as an answer to a query
- The retrieval unit can be a file, a document, a Web page, a paragraph, etc
- From this point on, we will call simply **documents** those retrieval units, although this can have different meanings

Query Languages Keyword Based Querying

Keyword Based Querying

- A **query** is the formulation of a user information need
- In its simplest form, a query is only composed of keywords
- Keyword based queries are popular because they are intuitive, easy to express, and allow for fast ranking
- A query can also be a more complex combination of operations involving several words
- We refer to single-word and multiple-word queries as **basic queries**

Single-Word Queries

- The most elementary query that can be formulated in a text retrieval system is the **word**
- Some models are also able to see the internal division of words into letters
- The alphabet is split into **letters** and **separators**
- And a word is a sequence of letters surrounded by separators
- Some models allow specifying that some characters are not letters but do not split a word
  - E.g. the hyphen in `on-line`
- It is good practice to leave the choice of what is a letter and what is a separator

Single-Word Queries

- The division of the text into words is not arbitrary, since words carry a lot of meaning in natural language
- The result of word queries is the set of documents containing at least one of the words of the query
- Further, the resulting documents are ranked according to a degree of similarity to the query
- To support ranking, two common statistics on word occurrences inside texts are commonly used
  - The first is called **term frequency** and counts the number of times a word appears inside a document
  - The second is called **inverse document frequency** and is based on a counting of the number of documents in which a word appears

Context Queries

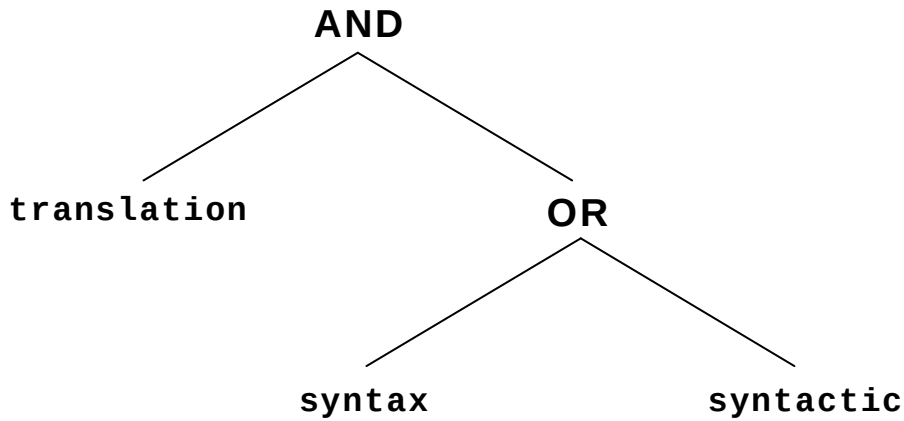
- Many systems complement single-word queries with the ability to search words in a given **context**
- Words which appear near each other may signal higher likelihood of relevance than if they appear apart
- We may want to form phrases of words or find words which are proximal in the text
  - Phrase
    - Is a sequence of single-word queries
    - An occurrence of the phrase is a sequence of words
    - Can be ranked in a fashion somewhat analogous to single words
  - Proximity
    - Is a more relaxed version of the phrase query
    - A maximum allowed distance between single words or phrases is given
    - The ranking technique can be depend on physical proximity

Boolean Queries

- The oldest mean to combine keyword queries is to use boolean operators
- A **boolean query** has a syntax composed of
  - **atoms**: basic queries that retrieve documents
  - **boolean operators**: work on their operands (which are sets of documents) and deliver sets of documents
- This scheme is in general **compositional**, i.e., operators can be composed over the results of other operators

Boolean Queries

- A **query syntax** tree is naturally defined
- The figure below shows an example of a query syntax tree



- It will retrieve all the documents which contain the word `translation` as well as either the word `syntax` or the word `syntactic`

Boolean Queries

- The operators most commonly used, given two basic queries or boolean sub-expressions e_1 and e_2, are:
  - e_1 **OR** e_2: the query selects all documents which satisfy e_1 or e_2
  - e_1 **AND** e_2: selects all documents which satisfy both e_1 and e_2
  - e_1 **BUT** e_2: selects all documents which satisfy e_1 but not e_2
  - **NOT** e_2: the query selects all documents which not contain e_2

Boolean Queries

- With classic boolean systems, no ranking of the retrieved documents is provided
- A document either satisfies the boolean query or it does not
- This is quite a limitation because it does not allow for partial matching between a document and a user query
- To overcome this limitation, the condition for retrieval must be relaxed
  - For instance, a document which partially satisfies an AND condition might be retrieved

Boolean Queries

- Users not trained in mathematics can find the meaning of boolean operators difficult to grasp
- With this problem in mind, a **fuzzy-boolean** set of operators has been proposed
- The idea is that the meaning of AND and OR can be relaxed, so that they retrieve more documents
- The documents are ranked higher when they have a larger number of elements in common with the query

Query Languages

Beyond Keywords

Pattern Matching

- A **pattern** is a set of syntactic features that must be found in a text segment
- Those segments satisfying the pattern specifications are said to **match** the pattern
- We can search for documents containing segments which match a given search pattern
- Each system allows specifying some types of patterns
- The more powerful the set of patterns allowed, the more involved queries can the user formulate, in general

Pattern Matching

- The most used types of patterns are:
  - **Words**: a string which must be a word in the text
  - **Prefixes**: a string which must form the beginning of a text word
  - **Suffixes**: a string which must form the termination of a text word
  - **Substrings**: a string which can appear within a text word
  - **Ranges**: a pair of strings which matches any word which lexicographically lies between them
  - **Allowing errors**: a word together with an error threshold
  - **Regular expressions**: a rather general pattern built up by simple strings
  - **Extended patterns**: a more user-friendly query language to represent some common cases of regular expressions

Natural Language

- Pushing the fuzzy boolean model even further, the distinction between AND and OR could be completely blurred
- In this case, a query becomes simply an enumeration of words and context queries
- All the documents matching some query are retrieved, giving more weight to those matching more parts of the query
- The negation can be handled by letting the user express that some words are not desired
  - Then the documents containing them are penalized in the ranking computation

Natural Language

- Under this scheme we have completely eliminated any reference to boolean operations and entered into the field of **natural language** queries
- The search criterion can be reexpressed using a different model, where documents and queries are considered just as a vector of **term weights**
- The algorithms for this model are totally different from those based on searching patterns

Query Languages

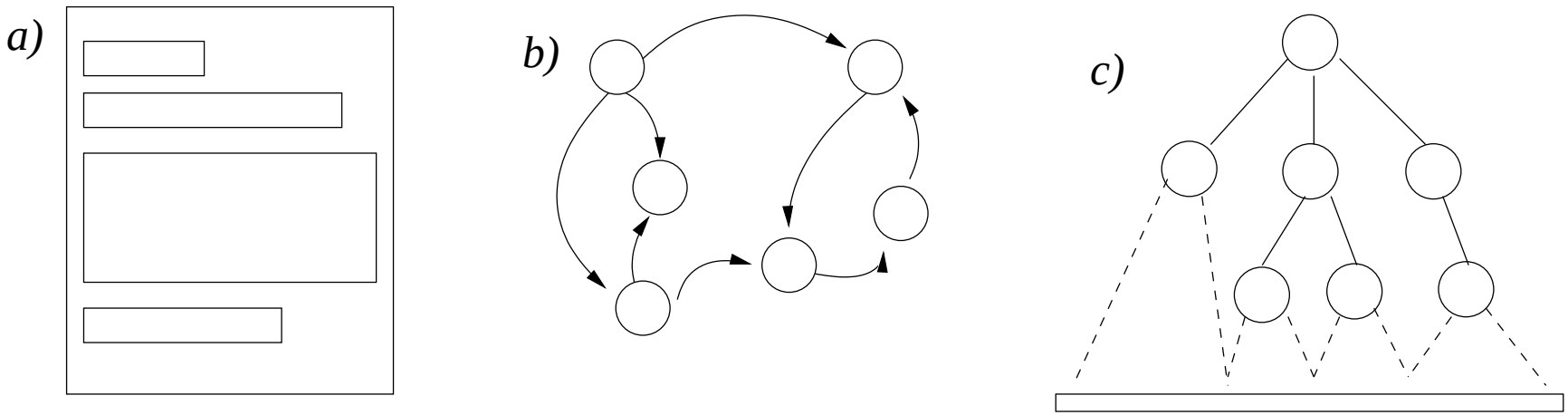
Structural Queries

Structural Queries

- The text collections tend to have some structure built into them
- Be able to query those texts based on their structure is becoming attractive
- The standardization of languages to represent structured texts such as HTML has pushed forward in this direction
- Mixing contents and structure in queries allows posing very powerful queries
- Queries can be expressed using containment, proximity or other restrictions on the structural elements present in the documents

Structural Queries

- The three main types of structures:
  - $a)$ form-like fixed structure
  - $b)$ hypertext structure
  - $c)$ hierarchical structure



Fixed Structure

- The structure allowed in texts was traditionally quite restrictive
- The documents had a fixed set of fields, and each each field had some text inside
  - Some fields were not present in all documents
  - Other fields could appear repeatedly across the documents
  - Some documents could have text not classified under any field
- They were not allowed to nest or overlap
- Retrieval activity allowed: specifying that a given basic pattern was to be found only in a given field

Fixed Structure

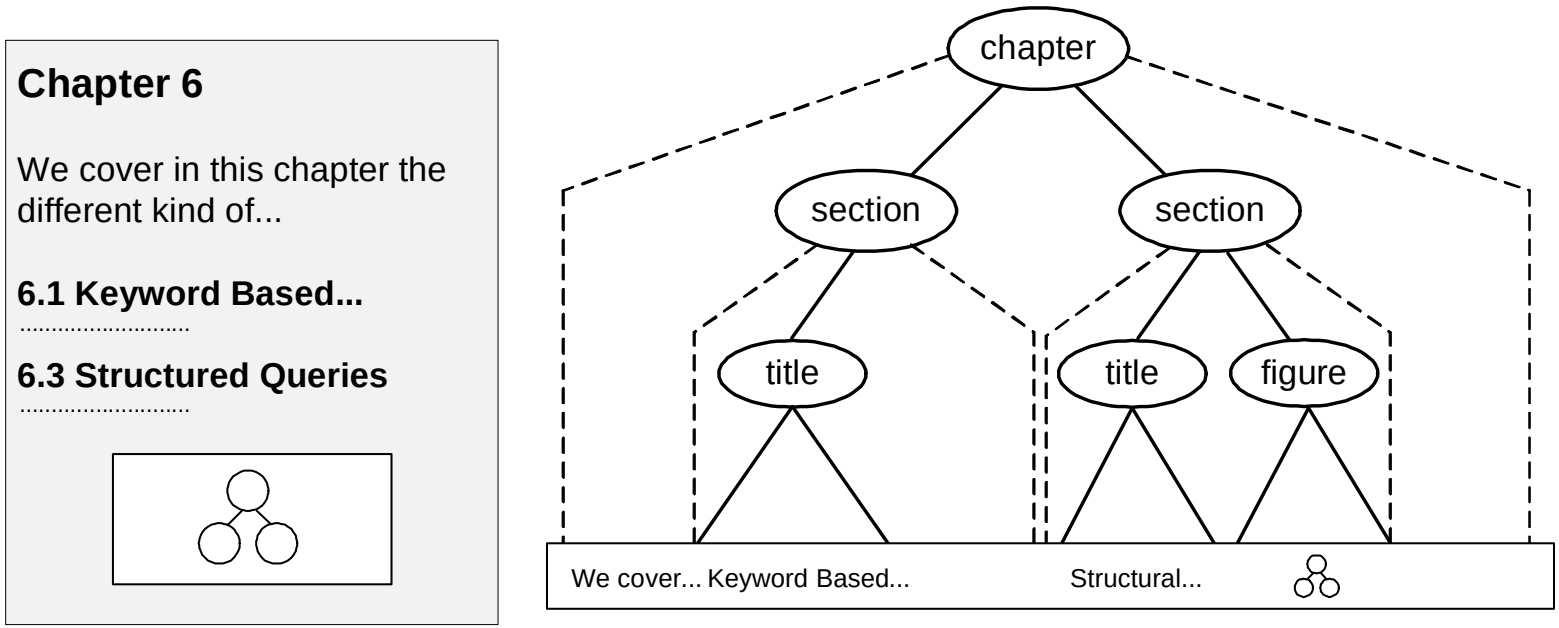
- Sometimes, the structure of the text is very rigid
- In this case, the content of some fields can even be interpreted not as text but as numbers, dates, etc
- This model allows different queries to be posed on the texts
  - E.g. month ranges in dates
- This idea leads naturally to the relational model, each field corresponding to a column in the database table
- There are several proposals that extend SQL to allow full-text retrieval
  - Among them we can mention proposals by the leading relational database vendors such as Oracle and Sybase, as well as SFQL

Fixed Structure

- Hypertexts probably represent the opposite trend with respect to structuring power
- A hypertext is a directed graph where
  - the nodes hold some text
  - the links represent connections among nodes or among positions inside the nodes
- Retrieval from hypertext began as a merely navigational activity
- That is, the user had to manually traverse the hypertext nodes following links to search what he/she wanted
- Currently, some query tools have appeared that achieve the goal of querying hypertext based on their content and their structure

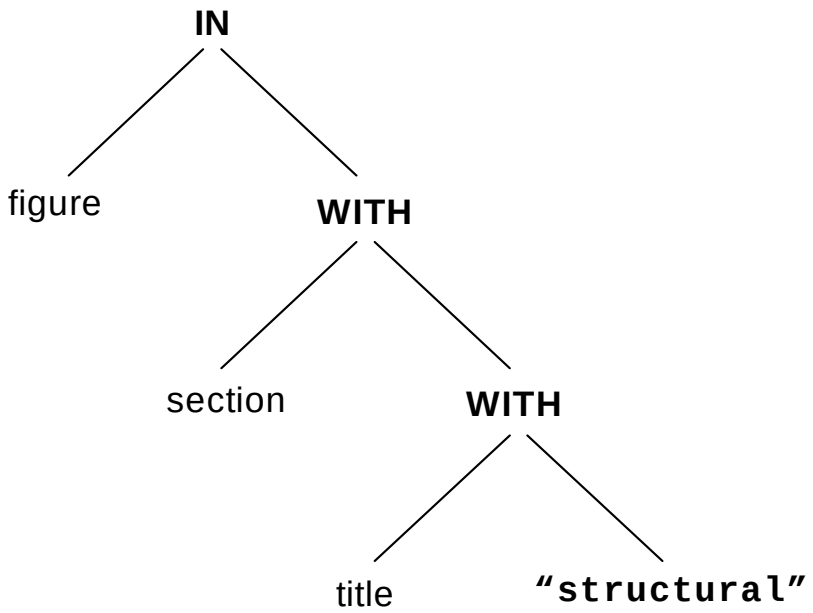
Fixed Structure

- An intermediate structuring model which lies between fixed structure and hypertext is the **hierarchical structure**
- This is a natural model for many text collections
- An example of a hierarchical structure: the page of a book and its schematic view

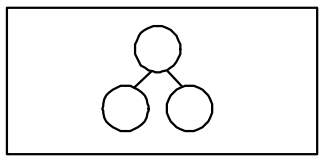


Fixed Structure

- An example of a query to the hierarchical structure presented



- This parsed query returns the image below



Query Languages

Query Protocols

Query Protocols

- We now present some query languages used by applications to query text databases
- Because they are not intended for human use, we refer to them as **protocols** rather than languages
- The most important are:
  - **Z39.50**
  - **Wide Area Information Service (WAIS)**

Query Protocols

- In the CD-ROM publishing arena there are several proposals for query protocols
- The main goal of these protocols is to provide **disc interchangeability**
- We can cite three of them:
  - **Common Command Language (CCL)**
  - **Compact Disk Read only Data exchange (CD-RDx)**
  - **Structured Full-text Query Language (SFQL)**
- SFQL is based on SQL and also has a client-server architecture
- The language does not define any specific formatting or markup

Query Protocols

- For example, a query in SFQL is:

```
Select abstract from journal.papers
where title contains "text search"
```

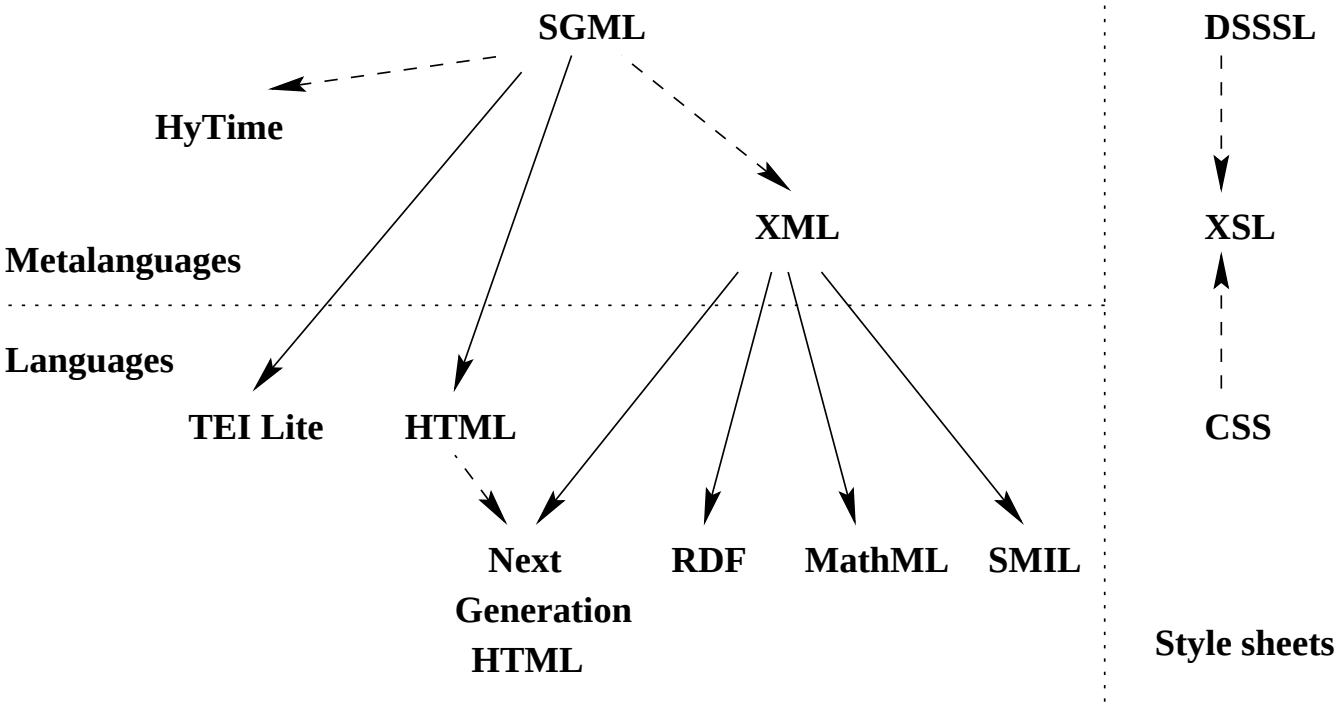
- The language supports boolean and logical operators, thesaurus, proximity operations and some special characters as wild-cards and repetition
- For example:

```
...  where paper contains "retrieval"
or like "info %" and date > 1/1/98
```

Trends and Research Issues

Trends and Research Issues

- Many changes and proposals are happening and very rapidly
- Figure below illustrates a taxonomy of the main languages considered



- Solid lines indicate instances of a meta-language, while dashed lines indicate derived languages

Trends and Research Issues

- The main trend is the convergence and integration of the different efforts, being the Web the main application
- A European alternative to SGML is the **Open Document Architecture** (**ODA**, ISO 8613)
- ODA was designed to share documents electronically without losing control over the content, structure and layout of those documents
- ODA defines a logical structure (like SGML), a layout and the content (including vector and raster graphics)

Trends and Research Issues

- An object model is being defined: the **Document Object Model (DOM)**
  - DOM will provide an interoperable set of classes and methods to manipulate HTML and XML objects from programming languages
- Recent developments include proposes to integrate VRML and Dynamic HTML
  - This integration provides a set of evolving features and architecture extensions to HTML
  - Such features include cascading style sheets and document object models.

Trends and Research Issues

- Recent developments also include:
  - Integration between the **Standard Exchange for Product Data format** (**STEP**, ISO 10303) and SGML
    - STEP covers product data from a broad range of industries
    - provides extensive support for modeling, automated storage schema generation, life-cycle maintenance, and other management facilities
  - Efforts to convert MARC to SGML by defining a DTD, as well as converting MARC to XML
    - This has potential possibilities for enhanced access and navigation and presentation of MARC record data and the associated information

Trends and Research Issues

- Several new proposals have appeared:
  - **Signed Document Markup Language (SDML)**
  - **Vector Markup Language (VML)**
  - **Precision Graphics Markup Language (PGML)**

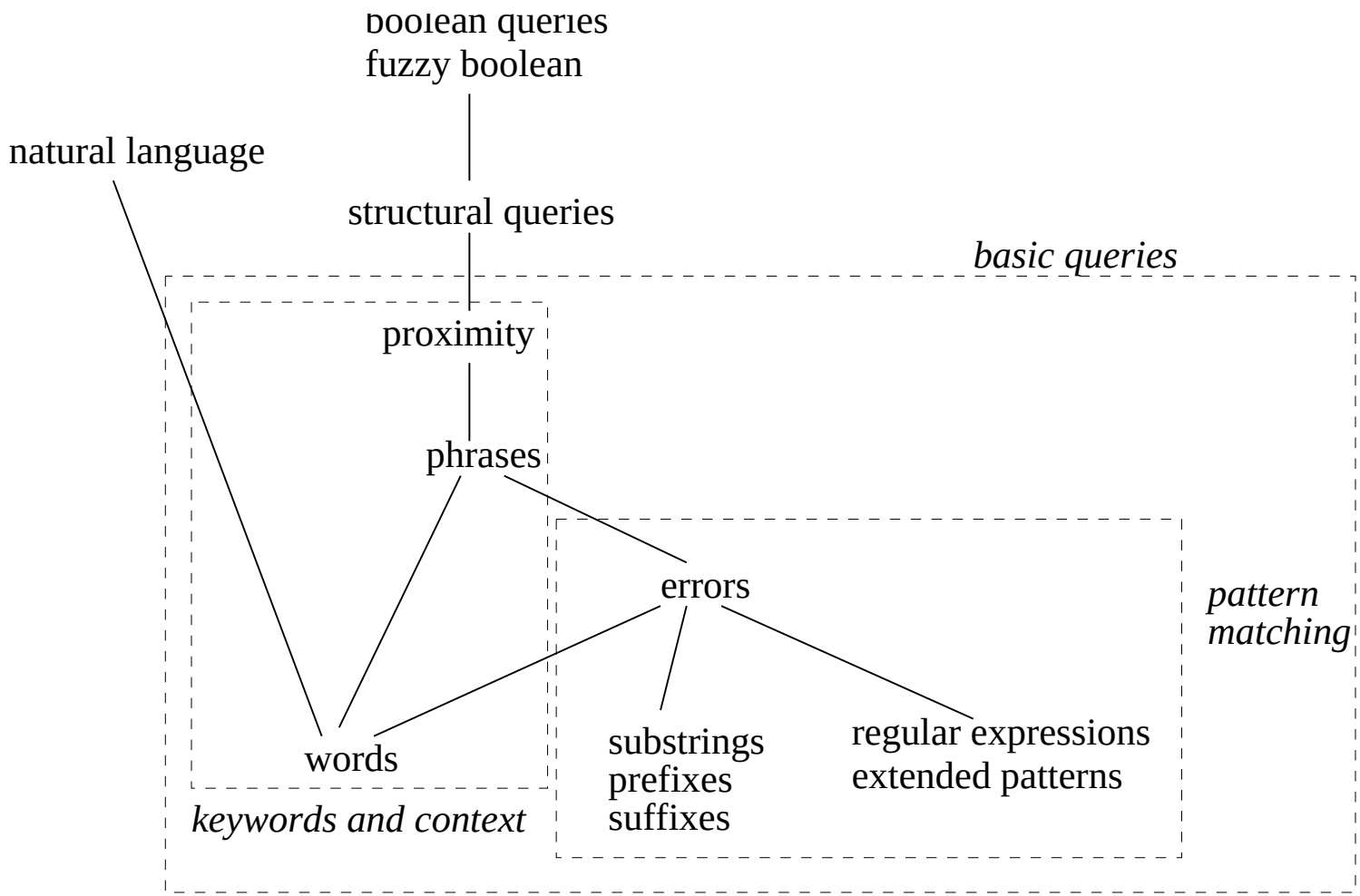
Trends and Research Issues

- We have also reviewed here the main aspects of the query languages that retrieve information from textual databases
- We ranged from the most classical tools to the most novel capabilities that are emerging
- The table below shows the different queries allowed in the different models

Model	Queries allowed
Boolean	word, set operations
Vector	words
Probabilistic	words
BBN	words

Trends and Research Issues

- The figure below present the types of operations we covered and how can they be structured



Trends and Research Issues

- The subject of query languages for text databases is definitely moving to a higher flexibility
- The text models are moving to the goal of achieving a better understanding of the user needs
- The query languages are allowing more and more power in the specification of the query
- Another important research topic is visual query languages
  - Visual metaphors can help non experienced users to pose complex boolean queries
  - A visual query language can also include the structure of the document