Unstructured Data Processing and Information Retrieval

Week 11

COS60009: Data Management for the Big Data Age

SWINBURNE UNIVERSITY OF TECHNOLOGY

1

Learning Objectives

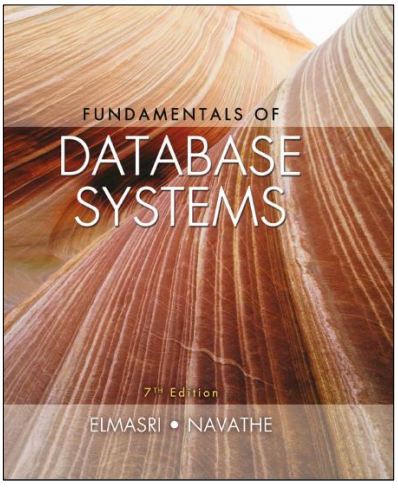
- Information Retrieval (IR) Concepts
- Retrieval Models
- Types of Queries in IR Systems
- Regular Expressions
- Text Pre-processing
- Inverted Indexing

Pearson

2

Fundamentals of Database Systems

Seventh Edition



Regular Expression

Chapter 27

Introduction to
Information Retrieval and
Web Search

3

Information Retrieval (IR) Concepts (1 of 2)

- Information retrieval
  - Process of retrieving documents from a collection in response to a query (search request)
  - Deals mainly with unstructured data
- Unstructured information
  - Does not have a well-defined formal model
  - Based on an understanding of natural language
  - Stored in a wide variety of standard formats
- Information retrieval field predates database field
  - Academic programs in Library and Information Science
- User's information need expressed as free-form search request
  - Keyword search query

4

Information Retrieval (IR) Concepts (2 of 2)

- Characterizing an IR system
  - Types of users: Expert, Layperson
  - Types of data could be domain-specific
  - Types of information needs
    - Navigational search - looking to reach a particular website
    - Informational search - looking for a specific bit of information
    - Transactional search - getting to a website where there will be more interaction, e.g. buying something, downloading something, signing up or registering etc.
- Enterprise search systems
  - Limited to an intranet
- Desktop search engines
  - Searches an individual computer system
- IR system has no fixed data model
  - Databases have fixed schemas

5

Comparing Databases and IR Systems

Table 27.1 A comparison of databases and IR systems

Databases	IR Systems
• Structured data	• Unstructured data
• Schema driven	• No fixed schema; various data models (e.g., vector space model)
• Relational (or object, hierarchical, and network) model is predominant	• Free-form query models
• Structured query model	• Rich data operations
• Rich metadata operations	• Search request returns list or pointers to documents
• Query returns data	
• Results are based on exact matching (always correct)	• Results are based on approximate matching and measures of effectiveness (may be imprecise and ranked)

6

Modes of Interactions in IR Systems

- Primary modes of interaction
  - Retrieval
    - Extract relevant information from document repository
  - Browsing
    - Exploratory activity based on user's assessment of relevance
- Web search combines both interaction modes
  - Rank of a web page measures its relevance to query that generated the result set

7

Generic IR Pipeline (1 of 2)

- Statistical approach
  - Documents analyzed and broken down into chunks of text
  - Each word or phrase is counted, weighted, and measured for relevance or importance
- Types of statistical approaches
  - Boolean
  - Vector space
  - Probabilistic

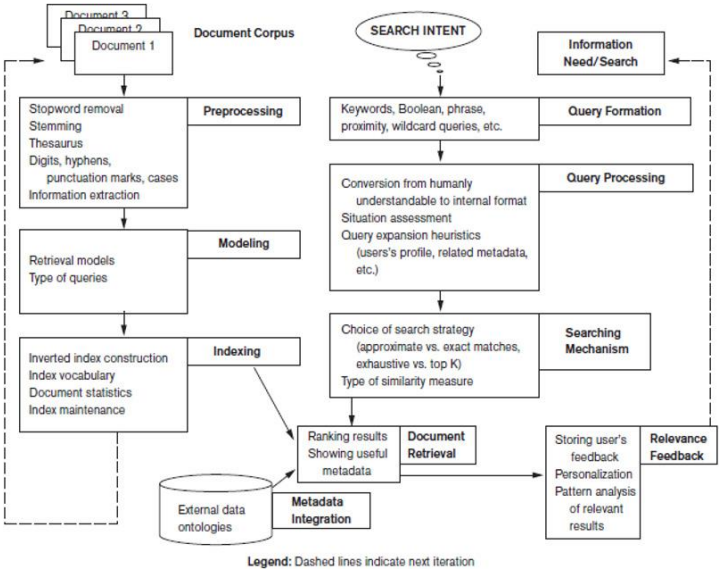
8

Generic IR Pipeline (2 of 2)

- Semantic approaches
  - Use knowledge-based retrieval techniques
  - Rely on syntactic, lexical, sentential, discourse-based, and pragmatic levels of knowledge understanding
  - Also apply some form of statistical analysis

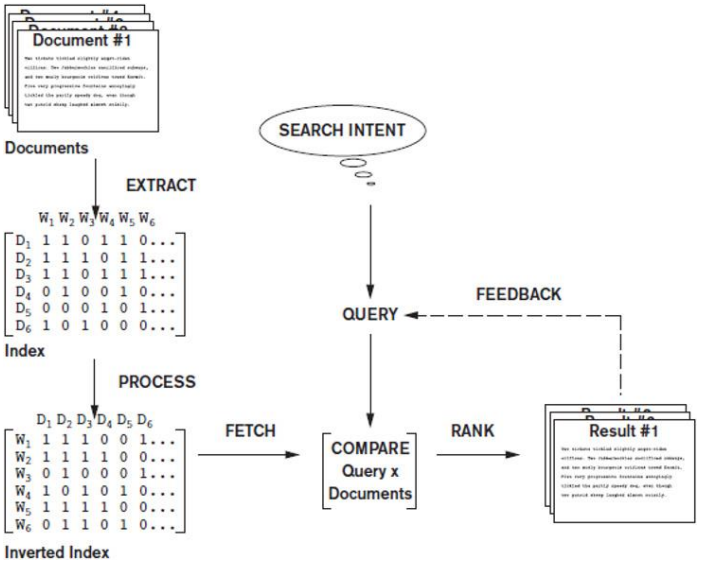
9

Figure 27.1 Generic IR Framework



10

Figure 27.2 Simplified IR Process Pipeline



11

Retrieval Models (1 of 4)

- Boolean model
  - One of earliest and simplest IR models
  - Documents represented as a set of terms
  - Queries formulated using AND, OR, and NOT
  - Retrieved documents are an exact match
    - No notion of ranking of documents
  - Easy to associate metadata information and write queries that match contents of documents

12

Retrieval Models (2 of 4)

- Vector space model
  - Weighting, ranking, and determining relevance are possible
  - Uses individual terms as dimensions
  - Each document represented by an n-dimensional vector of values
  - Features
    - Subset of terms in a document set that are deemed most relevant to an IR search for the document set
  - Different similarity assessment functions can be used
- Term frequency-inverse document frequency (TF-IDF)
  - Statistical weight measure used to evaluate the importance of a document word in a collection of documents
  - A discriminating term must occur in only a few documents in the general population

13

Retrieval Models (3 of 4)

- Probabilistic model
  - Involves ranking documents by their estimated probability of relevance with respect to the query and the document
  - IR system must decide whether a document belongs to the relevant set or nonrelevant set for a query
    - Calculate probability that document belongs to the relevant set
  - BM25: a popular ranking algorithm

14

Retrieval Models (4 of 4)

- Semantic model
  - Morphological analysis: analyze roots and affixes to determine parts of speech of search words
  - Syntactic analysis: parse and analyze complete phrases in documents
  - Semantic analysis: resolve word ambiguities and generate relevant synonyms based on semantic relationships
  - Uses techniques from artificial intelligence and expert systems

15

Types of Queries in IR Systems (1 of 4)

- Keyword queries
  - Simplest and most commonly used
  - Keyword terms implicitly connected by logical AND
- Boolean queries
  - Allow use of AND, OR, NOT, and other operators
  - Exact matches returned
    - No ranking possible

16

Types of Queries in IR Systems (2 of 4)

- Phrase queries
  - Sequence of words that make up a phrase
  - Phrase enclosed in double quotes
  - Each retrieved document must contain at least one instance of the exact phrase

17

Types of Queries in IR Systems (3 of 4)

- Proximity queries
  - How close within a record multiple search terms are to each other
  - Phrase search is most commonly used proximity query
  - Specify order of search terms
  - NEAR, ADJ (adjacent), or AFTER operators
  - Sequence of words with maximum allowed distance between them
  - Computationally expensive
    - Suitable for smaller document collections rather than the Web

18

Types of Queries in IR Systems (4 of 4)

- Wildcard queries
  - Supports **regular expressions** and pattern-based matching
    - Example 'data*' would retrieve data, database, dataset, etc.
  - Not generally implemented by Web search engines
- Natural language queries
  - Definitions of textual terms or common facts
  - Semantic models can support

19

Regular Expressions

- Regular expressions are a useful way to concisely define the syntax and 'pattern' of textural data.
- Regular expressions are strings that describe the 'pattern' or 'rules' for strings
- Regular expressions are strings that follow a set of syntax rules
- Regular expressions can be used as a concise and consistent way to test for matching patterns

20

Regular Expressions Matching

- Perform a regular expression match `preg_match()` in PHP
- Initialise a Regular Expression pattern, and test string

```
$pattern1 = "/(chapter \d+(\.\d)*)/i";
$str1 = "For more information, see Chapter
3.4.5.1";
if (preg_match($pattern1, $str1) {
    echo "A match was found.";
} else {
    echo "A match was not found.";
}
```


21

Regular Expressions - Basic Examples

/WebProg/	matches "Isn't WebProg great?"
/^WebProg/	matches "WebProg rules!", not "What is WebProg?"
/WebProg$/	matches "I love WebProg", not "WebProg is great!"
/^WebProg$/	matches "WebProg", and nothing else
/bana?na/	matches "banana" and "banna", but not "banaana".
/bana+na/	matches "banana" and "banaana", but not "banna".
/bana*na/	matches "banna", "banana", and "banaaana", but not "bnana"
/^[a-zA-z]+$/	matches any string of one or more letters and nothing else.

22

Regular Expressions – More Examples

/[A-Za-z0-9-]+/	letters, numbers and hyphens
/\d{1,2}\/\d{1,2}\/\d{4}/	date such as 19/9/2006
/\S+\.(jpg|gif|png)	jpg, gif or png filename
/^[1-9]{1}$\|^[1-2]{1}[0-9]{1}$|^30$/	any number from 1 to 30
/#?([A-Fa-f0-9]){3}(([A-Fa-f0-9]){3})?/	valid hex colour code
/(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,15}/	password string (at least 1 uppercase letter, 1 lowercase letter and 1 digit)
/^.+@.+\..{2,3}$/	possible email address (in US)
/\<(/?[^\>]+)\>/	HTML tags

Pearson

23

24

Regular Expressions - Basic Syntax

Quantifiers

*	0 or more
+	1 or more
?	0 or 1
{4}	exactly 4
{4,}	4 or more
{4,6}	4,5 or 6

Groups & Ranges

.	Any character (except \n)
(a|b)	a or b
(abc)	group of abc (a followed by b followed by c)
[abc]	set (“range”) a, b or c
[^abc]	not a, b or c
[a-g]	set range a to g
[3-6]	set range of digits 3,4,5 and 6

Pearson

24

Regular Expressions - Basic Syntax

Pattern Basics

^	Start of string
$	End of string
.	Match any single character
(a|b)	a or b
(…)	Group section
[abc]	match any character in the set
[^abc]	not match in the set
[a-z]	match the range
\d	match a single digit from 0 to 9 *shortcut for [0-9]*
\D	not a digit

Pattern Quantifiers

a?	0 or 1 of a
a*	0 or more of a
a+	one or more instance of a
a{3}	exactly 3 a's = aaa
a{3,}	3 or more a's
a{3,6}	between 3 to 6 a's
!(pattern)	"not" pattern

[\ ^ $. | ? * + ( ) are the 11 **meta-characters**, or special characters, used in the syntax.
If you want to include these, you need to escape them with \

eg. \(and \.

Pearson

Regular Expressions - Basic Syntax

Pattern Modifiers

/g	global matching
/i	case insensitive
/s	single line mode
/m	multiple-line mode
/x	allow comments and white space in pattern
/e	evaluate replacement
/U	ungreedy replacement

Assertions

(?=	look ahead positive
(?!	look ahead negative
(?<=	look behind positive
(?<!	look behind negative

There are many useful online syntax references about Regular Expressions, such as:
http://www.php.net/manual/en/reference.pcre.pattern.syntax.php

Pearson

Regular Expressions - Basic Syntax

Anchors

^	start of string
\A	start of string
$	end of string
\Z	end of string
\b	word boundary
\B	not word boundary
\<	start of word
\>	end of word

Special Characters

\n	newline
\r	carriage return
\t	tab
\v	vertical tab
\f	form feed
\xhh	hex character hh
\s	whitespace charater
\S	not a whitespace character

Swinburne Library eBooks, can provide examples of Regular Expressions in many different scripting languages, e.g. search for 'regular expressions' in http://proquest.safaribooksonline.com/search

Text Preprocessing (1 of 3)

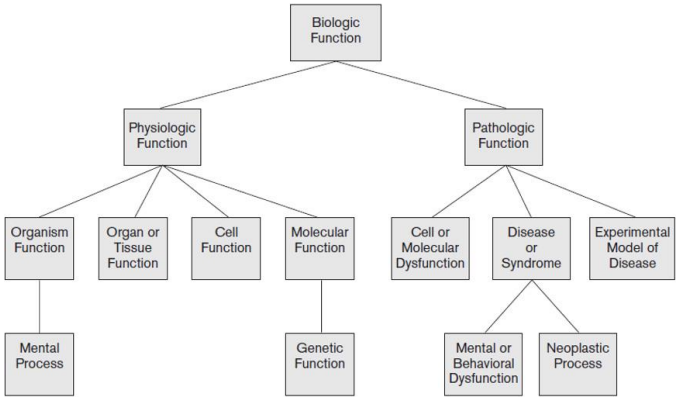
- Stopword removal must be performed before indexing
- Stopwords
  - Words that are expected to occur in 80% or more of the documents of a collection
    - Examples: the, of, to, a, and, said, for, that
  - Do not contribute much to relevance
- Queries preprocessed for stopword removal before retrieval process
  - Many search engines do not remove stopwords

Text Preprocessing (2 of 3)

- Stemming
  - Trims suffix and prefix
  - Reduces the different forms of the word to a common stem
  - Martin Porter's stemming algorithm
- Utilizing a thesaurus
  - Important concepts and main words that describe each concept for a particular knowledge domain
  - Collection of synonyms
  - UMLS

29

Figure 27.3 A Portion of the UMLS Semantic Network: "Biologic Function" Hierarchy



Source: UMLS Reference Manual, National Library of Medicine

30

Text Preprocessing (3 of 3)

- Other preprocessing steps
  - Digits
    - May or may not be removed during preprocessing
  - Hyphens and punctuation marks
    - Handled in different ways
  - Cases
    - Most search engines use case-insensitive search
- Information extraction tasks
  - Identifying noun phrases, facts, events, people, places, and relationships

31

Inverted Indexing (1 of 3)

- Inverted index structure
  - Vocabulary information
    - Set of distinct query terms in the document set
  - Document information
  - Data structure that attaches distinct terms with a list of all documents that contain the term

32

Inverted Indexing (2 of 3)

- Construction of an inverted index
  - Break documents into vocabulary terms
    - Tokenizing, cleansing, removing stopwords, stemming, and/or using a thesaurus
  - Collect document statistics
    - Store statistics in document lookup table
  - Invert the document-term stream into a term-document stream
    - Add additional information such as term frequencies, term positions, and term weights

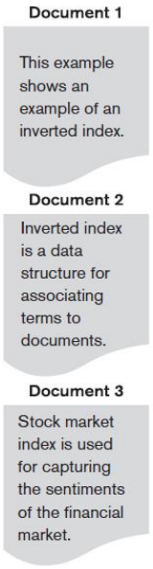
33

Inverted Indexing (3 of 3)

- Searching for relevant documents from an inverted index
  - Vocabulary search
  - Document information retrieval
  - Manipulation of retrieved information

34

Figure 27.4 Example of an Inverted Index



ID	Term	Document: position
1.	example	1:2, 1:5
2.	inverted	1:8, 2:1
3.	index	1:9, 2:2, 3:3
4.	market	3:2, 3:13

Pearson

35

Copyright

Pearson

36