Text Analysis and Processing

Berlin Chen

Department of Computer Science & Information Engineering
National Taiwan Normal University

References:

1. *Modern Information Retrieval*, Chapters 6 & 7
2. *Search Engines: Information Retrieval in Practice,* Chapter 4

Documents

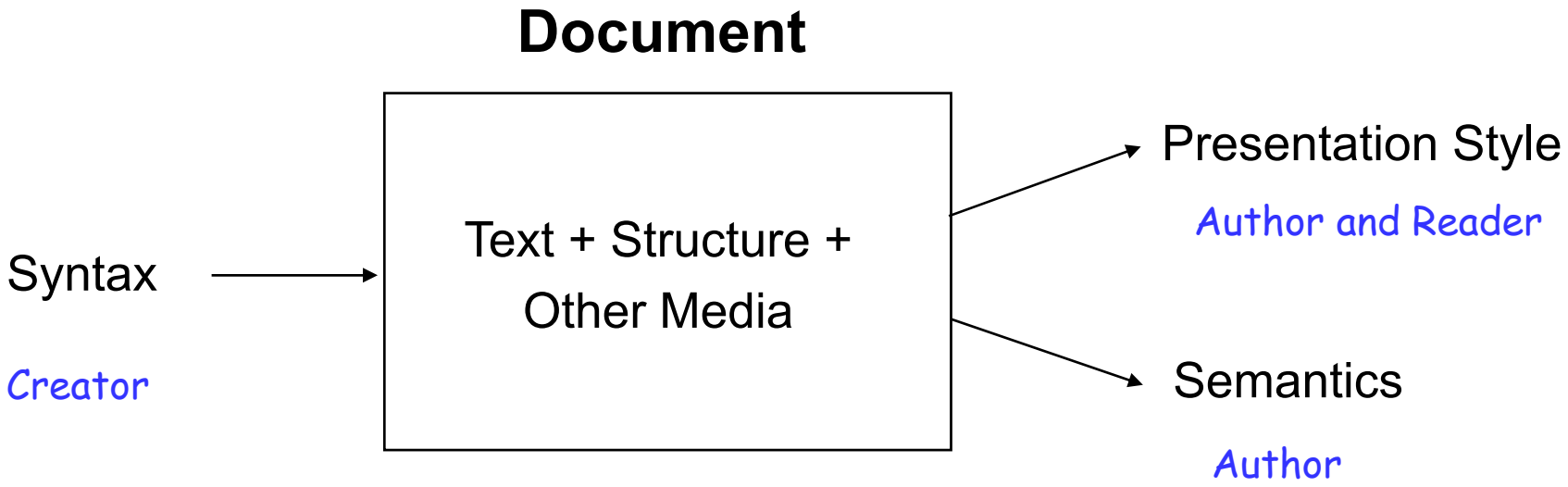
- A document is a single unit of information
  - Typical text in digital form, but can also include other media
- Two perspectives
  - Logical View
    - Complete: A unit like a research article, a book or a manual
    - Incomplete: A paragraph or a sequence of paragraphs (passage)
  - Physical View
    - A unit like a file, an email, or a Web page

Syntax of a Document

- Syntax of a document can express structure, presentation style, semantics, or even external actions
  - A document can also have information about itself, called **metadata**
- The syntax of a document can be explicit in its content, or expressed in a simple declarative language or in a programming language
  - But the conversion of documents in one language to other languages (or formats) is very difficult !
  - How to flexibly interchange between applications is becoming important

Many syntax languages are proprietary and specific !

Characteristics of a Document



- The **presentation style** of a document defines how the document is visualized in a computer window or a printed page
  - But can also includes **treatment of other media such as audio or video**

Metadata (1/2)

- Metadata: “data about data”
  - Is information on the organization of the data, the various data domains, and the relationship between them

- Descriptive Metadata
  - Is external to the meaning of the document and pertains more to how document was created
  - Information including author, date, source, title, length, genre (book, article, memo, etc.), …
  - E.g., Dublin Core Metadata Element Set
    - 15 fields to describe a doc

Metadata (2/2)

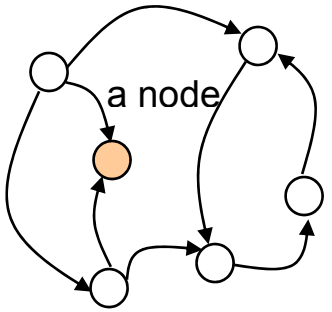
- Semantic Metadata
  - Characterize the subject matter about the document's contents
  - Information including subject codes, abstract, keywords (key terms)
  - To standardize semantic terms, many areas use specific **ontologies**, which are hierarchical taxonomies of terms describing certain knowledge topics
  - E.g., Library of Congress subject codes

(semantic metadata for ACM papers)

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*
General Terms: Algorithms, Performance, Theory
Additional Key Words and Phrases: Hidden Markov models, Mandarin spoken documents, syllable-level Indexing features

Web Metadata

- Used for many purposes, e.g.,
  - Cataloging
  - Content rating
  - Intellectual property rights
  - Digital signatures
  - Privacy levels
  - Electronic commerce



- RDF (Resource Description Framework)
  - A new standard for Web metadata which provides interoperability between applications
  - Allow the description of Web resources to facilitate automated processing of information

Metadata for Non-textual Objects

- Such as images, sounds, and videos
  - A set of keywords used to describe them
    - Meta-descriptions
  - These keywords can later be used to search for these media using classical text IR techniques
  - The emerging approach is content-based indexing
    - Content-Based Image Retrieval
    - Content-Based Speech Retrieval
    - Content-Based Music Retrieval
    - Content-Based Video Retrieval
    - ….

Text

- What are the possible formats of text ?
  - Coding schemes for languages
    - E.g., EBCDIC, ASCII, Unicode(16-bit code)

- What are the statistical properties of text ?
  - How the information content of text can be measured
  - The frequency of different words
  - The relation between the vocabulary size and corpus size

Factors affect IR performance and term weighting and other aspects of IR systems

Text: Formats (1/2)

- Text documents have no single format, and IR systems deal with them in two ways
  - Convert a document to an internal format
    - Disadvantage: the original application related the document is not useful any more
  - Using filters to handle most popular documents
    - E.g., word processors with some binary syntax like Word, WordPerfect, …
    - But some formats are proprietary and thus can't be filtered
    - Documents in human-readable ASCII form are more portability than those in binary form

Text: Formats (2/2)

- Other text formats developed for document interchange
  - **Rich Text Format (RTF):** is used for interchange between word processors and has ASCII syntax
  - **Portable Document Format (PDF)** and **Postcript:** is used for display or printing documents
  - **MIME (Multipurpose Internet Mail Exchange):** supports multiple character sets, multiple languages, and multiple media

Text: Information Theory (1/2)

- Written text contains semantics for information communication
  - E.g., a text where only one symbol appears almost all the time does not convey much information
- Information theory uses **entropy** to capture information context (uncertainty) of text

Entropy: the amount of information in a text

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

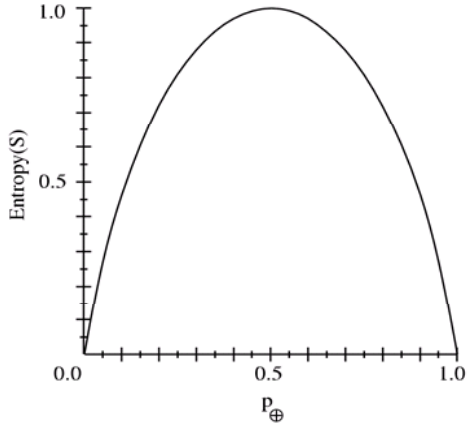
σ : number of symbols

  - Given σ =2, and the symbols coded in binary
    - Entropy is 1 if both symbols appear the same number of times
    - Entropy is 0 if only one symbol appears

Text: Information Theory (2/2)

- The calculation of **entropy** depends on the probabilities of symbols which were obtained by a text (probabilistic) model
  - The amount of information in a text is measured with regard to the text model
  - E.g., in text compression
    - Entropy is a limit on how much the text can be compressed, depending on the text model

- For Boolean classification (0 or 1)



$$Entropy(X) = -p_1 \log_2 p_1 - p_2 \log_2 p_2$$

Text: Modeling Natural Languages (1/7)

- **Issue1**: Text of natural languages composed of symbols from a finite alphabet set
  - **Word-level** (within word)
    - Symbols **separating words** or **belonging to words,** and symbols are not uniform distributed
    - Vowel letters (e.g., a, **e**, i, o, u) are more frequent than most constant letters in English (**e** is the most frequent)
    - The simple binominal model (0-order Markovian model) was used to generate text
    - However, dependency for letters' occurrences was observed
      - *k*-order Markovian model further is used (the probability a symbol depends on previous words)
      - E.g., "f" cannot appear after "c"

Text: Modeling Natural Languages (2/7)

- **Sentence-level** (within sentence)
  - Take words as symbols
  - *k*-order Markovian model was used to generate text (also called *n*-gram language models)
    - E.g., text generated by 5-order model using the distribution of words in the Bible might make sense
  - More complex models
    - Finite-state models (regular languages)
    - Grammar models (context-free and other languages)

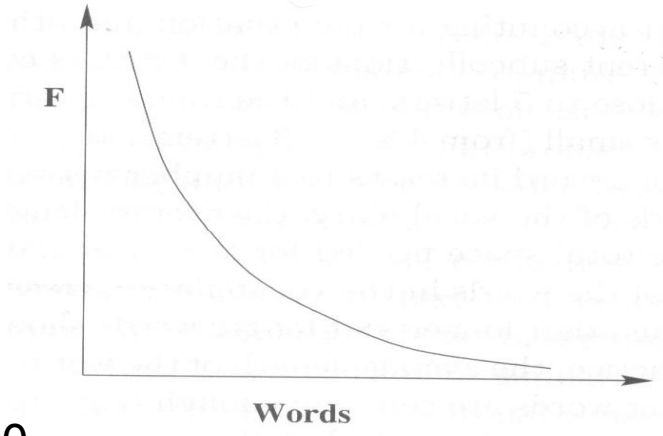
- Trigram approximation to Shakespeare
  - (a) Sweet prince, Falstaff shall die. Harry of Monmouth's grave.
  - (b) This shall forbid it should be branded, if renown made it empty.
  - (c) What is't that cried?
  - (d) Indeed the duke; and had a very good friend.
  - (e) Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.
  - (f) The sweet! How many then shall posthumus end his miseries.

- Quadrigram (fourgram) approximation to Shakespeare
  - (a) King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;
  - (b) Will you not tell me who I am?
  - (c) It cannot be but so.
  - (d) Indeed the short and the long. Marry, 'tis a noble Lepidus
  - (e) They say all lovers swear more performance than they are wont to keep obliged faith unforfeited!
  - (f) Enter Leonato's brother Antonio, and the rest, but seek the weary beds of people sick.

Text: Modeling Natural Languages (3/7)

- **Issue 2**: How the different words are distributed inside each documents
  - **Zipf's law** : an approximate model
    - Attempt to capture the distribution of the frequencies (number of occurrences) of the words
    - The frequency of the *i*-th most frequent word is $1 / i^{\theta}$ times that of the most frequent word
    - E.g., 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

$$H_V(\theta) = \frac{1}{1^{\theta}} + \frac{1}{2^{\theta}} + ... + \frac{1}{V^{\theta}} = \sum_{j=1}^{V} \frac{1}{j^{\theta}}$$



θ: depends on the text, between 1.5 and 2.0

Text: Modeling Natural Languages (4/7)

– A few hundred words take up 50% of the text !

- Words that are too frequent (known as `stopwords`) can be discarded
- Stopwords often doe not carry meaning in natural language and can be ignored
  – E.g., “a,” “the,” “by,” etc.

Text: Modeling Natural Languages (5/7)

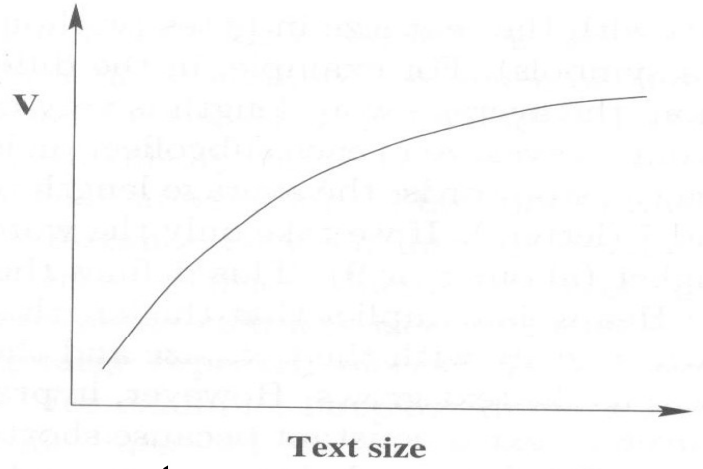
- **Issue 3**: the distribution of words in the documents of a collection
  - The fraction of documents containing a word *k* time is modeled as a negative binominal distribution

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

    - *p* and α are parameters that depend on the word and the document collection
      - E.g., *p*=9.2 and α=0.42 for the word "said" in the Brown Corpus

Text: Modeling Natural Languages (6/7)

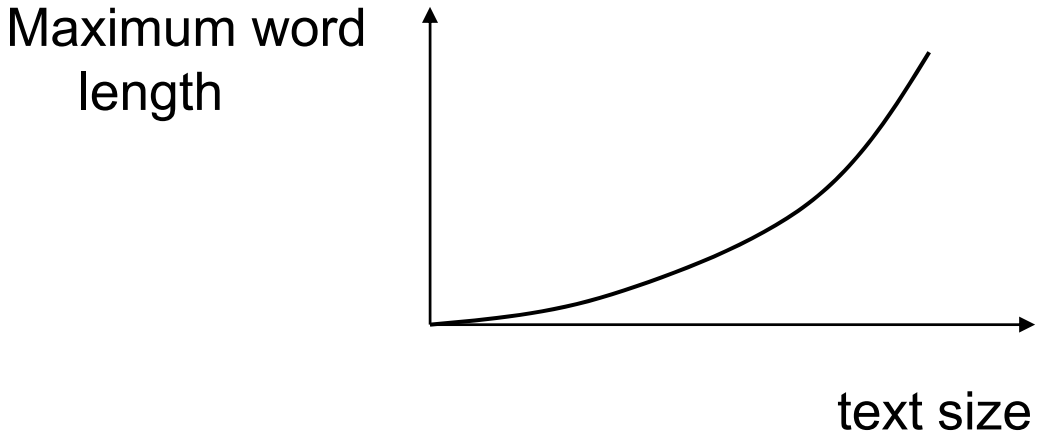
- **Issue 4**: the number of distinct words in a document (also called "**document vocabulary**")
  - **Heaps' Law**
    - Predict the growth of the vocabulary size in natural language text
    - The vocabulary of a text of size n words is of size $V=KN^{\beta}=O(N^{\beta})$
      - K :10~100
      - β: a positive number less than 1



    - Also applicable to collections of documents
      - The vocabulary of a text grows sublinearly with the text size (typos and spelling errors matter!)

Text: Modeling Natural Languages (7/7)

- **Issue 5**: the average length of words
  - **Heaps' Law**
    - Imply that the length of words of the vocabulary increases logarithmically with the text size
    - Longer and longer words should appear as the text grows
    - However, in practice, the average length of the words in the overall text is constant because shorter words (stopwords) are common enough



Text: Similarity Models (1/2)

- The syntactic similarity between strings or documents is measured by a distance function
  - Should be symmetric $distance(a,b) = distance(b,a)$
  - Should satisfy the triangle inequality

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

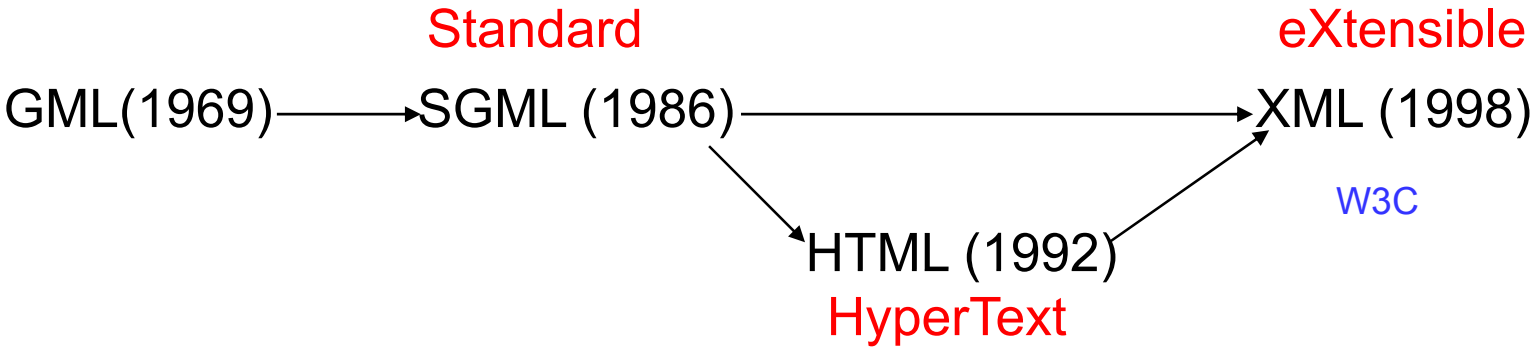
- Variant distance functions
  - Hamming distance
    - The number of positions that have different characters between two strings of the same length

Text: Similarity Models (2/2)

- Variant distance functions
  - Edit (or Levenshtein) distance
    - The minimum number of character **insertions**, **deletions**, and **substitutions** needed to perform to make any two strings equal (different weights/penalties can be applied for different operations)
    - E.g., 'color' and 'colour', 'survey' and 'surgery'
  - Longest Common Subsequence (LCS)
    - The only allowed operation is deletion of characters
    - Measure the remaining longest common subsequence of both string
    - E.g., 'survey' and 'surgery' → 'surey'
- The above similarity measures can be extended to documents
  - Lines in documents are considered as single symbols

Markup Languages

- The extra textual language used to describe formatting actions, structure information, text semantics, attributes, etc (Layout of documents)
  - Use marks (or called '**tags**') to surround the marked text
- The standard meta-language for markup is SGML (Standard Generalized Markup Languages)



SGML (1/2)

- Document Type Declaration (DTD) in SGML
  - Grammar or schema for defining the tags and structure of a particular document type
  - Allows defining structure of a document **element** using a regular expression
  - Expression defining an **element** can be recursive, allowing the expressive power of a context-free grammar

- A SGML document is defined by
  - DTD (a description of the document structure)
  - The text itself marked with initial and ending tags for describing the structure

SGML (2/2)

- Information about document's semantics, application conventions, etc., can be expressed informally as comments
  - DTD does not defined the semantics (meaning, presentation, and behavior), intended use of the tag
  - More complete information is usually present in separation documentation

- SGML does not specify how a doc should look
  - Separate content from format
  - Output specification can be added to SGML documents
    - E.g., Document Style Semantic Specification Language (DSSL) ,..

```
<!--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 >

<!--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 Márquez </Cc>
  </prolog>
  <contents>
       <par>
       As promised in my previous letter, I am sending two digital
       pictures to show you my house and the splendid view of the
       Pacific Ocean from my bedroom (photo <ref idref=F2>).
       </par>
       <image id=F1> "photo1.gif" </image>
       <image id=F2> "photo2.jpg" </image>
       <par>
       Regards from the South, Pablo.
  </contents>
</e-mail>
```

Document Type Declaration (DTD)

A document using DTD

optional (omission of) ending tag

Figure 6.3 DTD for structuring electronic mails and an example of its use.

HTML

- HTML: Hypertext Markup Language
  - An instance of SGML, created in 1992
  - Version 4.0 announced in 1997
- May include code such as Javascript in Dynamic HTML (DHTML)
- Separates layout somewhat by using style sheets (Cascade Style Sheets, CSS)
- HTML primarily defines layout and formatting

Visual effects for improving the aesthetics of HTML pages

XML (1/2)

- XML: eXtensible Markup Language
  - A simplified subset of SGML
- Simplification of original SGML for the Web promoted by WWW Consortium (W3C)
- Fully separates semantic information and layout
  - Allow a human-readable semantic makeup
- XML impose rigid syntax on the markup
  - Case sensitive
  - Data validation capabilities

XML (2/2)

- Allow users to define new tags, define more complex structures
- The using of DTD is optional
- Recent uses of XML include
  - Mathematical Markup Language (MathML)
  - Synchronized Multimedia Interchange Language (SMIL)
  - Resource Description Format (RDF)
  - VoiceXML
    - For speech-enabled Web pages
    - Compete with Microsoft SALT (Speech Application Language Tags)

No DTD included

```
<?XML VERSION="1.0" RMD="NONE" ?>
<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 Márquez </Cc>
  </prolog>
  <contents>
      <par>
      As promised in my previous letter, I am sending two digital
      pictures to show you my house and the splendid view of the
      Pacific Ocean from my bedroom (photo <ref idref="F2"/>).
      </par>
      <image id="F1" ref="photo1.gif" />
      <image id="F2"> ref="photo2.jpg" />
      <par>
      Regards from the South, Pablo.
      </par>
  </contents>
</e-mail>
```

For elements without textual content

Figure 6.5 An XML document without a DTD analogous to the previous SGML example.

Multimedia (1/3)

- Most common types of media in multimedia applications
  - Text
  - Sound (Speech/Music)
  - Images
  - Video
- These types of media is quite different in
  - Volumes
  - Formats
  - Processing requirements
  - Presentation styles (spatial and temporal attributes)

Multimedia (2/3)

- Formats
  - Image
    - Bit-mapped (or pixel-based) display
      - XBM, BMP, PCX
      - Simple but consume too much space (redundancy)
    - Compressed Images
      - Compuserve's Graphic Interchange Format (GIF)
      - Lossy Compressed Images
        - » Joint Photographic Experts Group (JPEG)
    - Exchange documents between different applications and platforms
      - Tagged Image File Format (TIFF)
      - True Version Targa Image File (TGA)

Multimedia (1/3)

- Formats
  - Audio
    - AU, MIDI, WAVE
    - RealAudio, CD formats
  - Video
    - MPEG (Moving Pictures Experts Group), AVI, FLI, QuickTime (by Apple)

Textural Images (1/2)

- Textural Images: images of documents that contain mainly typed or typeset text
  - Obtained by scanning the documents, usually for archiving purposes
  - Can be used for retrieval purposes and data compression
- Retrieval of Textural Images
  - **Alternative 1**
    - At creation time, a set of keywords (called metadata) is associated with each textual image
    - Conventional text retrieval techniques can be applied to keywords

Textural Images (2/2)

- Retrieval of Textural Images (cont.)
  - **Alternative 2**
    - Use OCR to extract the text of the image
    - The resultant ASCII text can be used to extract keywords
    - Quality depends on the OCR process
  - **Alternative 3**
    - Symbols extracted from the images are used as basic units to combine image retrieval techniques with sequence retrieval techniques
      - E.g., approximately matching of symbol strings between the query and extracted symbols
    - A promising but difficult issue

Index Term Selection and Text Operations

- Index Term Selection
  - Noun words (or group of noun words) are more representative of the semantics of a doc content
  - Preprocess the text of docs in collection in order to select the meaningful/representative index terms
    - Control the size of the vocabulary

E.g., "~~the~~ house ~~of~~ ~~the~~ lord"

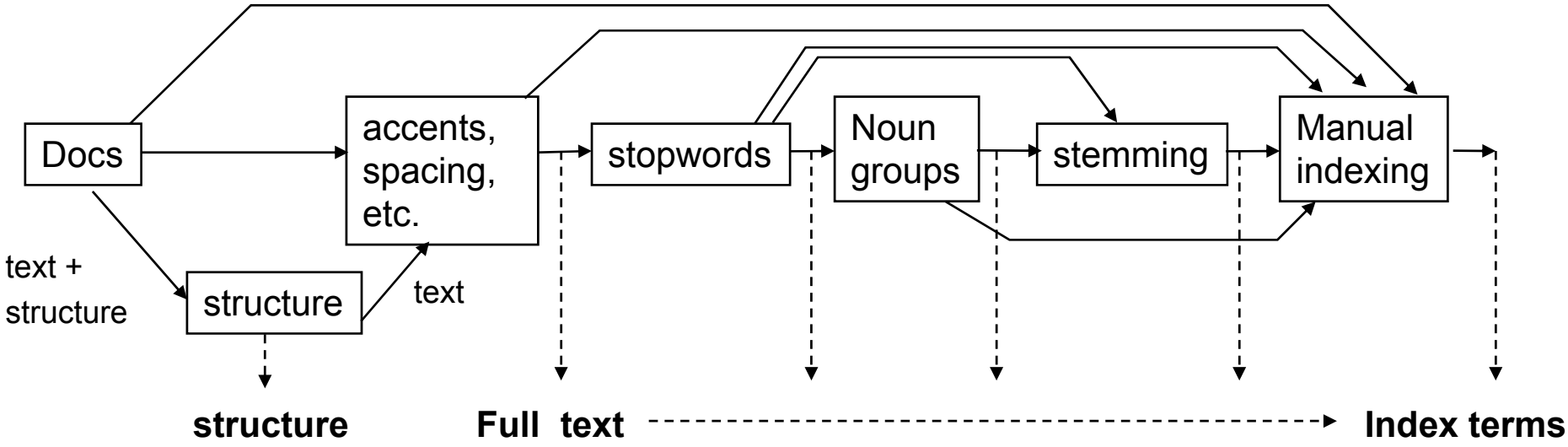
- Text Operations
  - During the preprocessing phrase, a few useful text operations can be performed
    - Lexical analysis
    - Eliminate of stop words
    - Stemming

    control the size of vocabulary (reduce the size of distinct index terms)
    side effect ?

    - Thesaurus construction/text clustering ⇨ improve performance but waste time
    - Text compression ⇨ controversial for its benefits
    - Encryption

Index Term Selection and Text Operations

- Logic view of a doc in text preprocessing



- Goals of Text Operations
  - Improve the quality of answer set (recall-precision figures)
  - Reduce the space and search time

Document Preprocessing

- Lexical analysis of the text
- Elimination of stopwords
- Stemming the remaining words
- Selecting of indexing terms
- Construction term categorization structures
  - Thesauri
  - Word/Doc Clustering

Lexical Analysis of the Text

- Lexical Analysis
  - Convert a stream of characters (the text of document) into stream words or tokens
  - The major objectives is to identify the words in the text

- Four particular cases should be considered with care
  - Digits
  - Hyphens
  - Punctuation marks
  - The case of letters

Lexical Analysis of the Text

- **Numbers/Digits**
  - Most numbers are usually not good index terms
  - Without a surrounding context, they are inherently vague
  - The preliminary approach is to remove all words containing sequences of digits unless specified otherwise
  - The advanced approach is to perform date and number normalization to unify format

- **Hyphens** anti-virus, anti-war,…
  - Breaking up hyphenated words seems to be useful
  - But, some words include hyphens as an integrated part
  - Adopt a general rule to process hyphens and specify the possible exceptions

state-of-the-art → state of the art

B-49 → B 49

Lexical Analysis of the Text

- **Punctuation marks**
  - Removed entirely in the process of lexical analysis
  - But, some are an integrated part of the word 510B.C.
- **The case of letters**
  - Not important for the identification of index terms
  - Converted all the text to either to either lower or upper cases
  - But, parts of semantics will be lost due to case conversion

John ⇨ john

The side effect of lexical analysis

User find it difficult to understand what the indexing strategy is doing at doc retrieval time.

Elimination of Stopwords

- **Stopwords**
  - Word which are too frequent among the docs in the collection are not good discriminators
  - A word occurring in 80% of the docs in the collection is useless for purposes of retrieval
    - E.g, articles, prepositions, conjunctions, …
  - Filtering out stopwords achieves a compression of 40% size of the indexing structure
  - **The extreme approach**: some verbs, adverbs, and adjectives could be treated as stopwords

- **The stopword list**
  - Usually contains hundreds of words

If queries are:

state of the art, to be or not to be, ….

Stemming

- Stem (詞幹)
  - The portion of a word which is left after the removal of affixes (prefixes and suffixes)
  - E.g., *V*(*connect*)={*connected, connecting, connection, connections, …* }

- Stemming
  - The substitution of the words with their respective stems
  - Methods
    - Affix removal
    - Table lookup
    - Successor variety (determining the morpheme boundary)
    - *N*-gram stemming based on letters' bigram and trigram information

Stemming: Affix Removal

- Use a suffix list for suffix stripping
  - E.g., **The Porter algorithm**
  - Apply a series of rules to the suffixes of words
    - Convert **plural forms** into **singular forms**
      - Words end in "*sses*"

        $sses \rightarrow ss$ stresses → stress
      - Words end in "*ies*" but not "*eies*" or "*aies*"

        $ies \rightarrow y$
      - Words end in "*es*" but not "*aes*", "*ees*" or "*oes*"

        $es \rightarrow e$
      - Word end in "*s*" but not "*us*" or "*ss*"

        $s \rightarrow \phi$

Stemming: Table Lookup

- Store a table of all index terms and their stems

Term	Stem
engineering	engineer
engineered	engineer
engineer	engineer

  - Problems
    - Many terms found in databases would not be represented
    - Storage overhead for such a table

Stemming: Successor Variety

- Based on work in structural linguistics
  - Determine word and morpheme boundaries based on distribution of phonemes in a large body of utterances
  - The successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached
    - At this point, the successor will sharply increase
    - Such information can be used to identify stems

Prefix	Successor Variety	Stem
R	3	E, I,O
RE	2	A, D
REA	1	D
READ	3	A, I, S
READA	1	B
READAB	1	L
READABL	1	E
READABLE	1	BLANK

Stemming: N-gram Stemmer

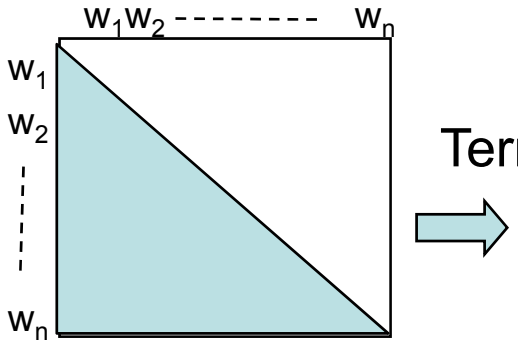
- Association measures are calculated between pairs of terms based on shared unique diagrams
  - diagram: or called the bigram, is a pair of consecutive letters
  - E.g.

statistics → st ta at ti is st ti ic cs
unique diagrams= at cs ic is st ta ti (7 unique ones)
statistical → st ta at ti is st ti ic ca al
unique diagrams= al at ca ic is st ta ti (8 unique ones)

6 diagrams shared

  - Using Dice's coefficient

$$S = \frac{2C}{A+B} = \frac{2x6}{7+8} = 0.80$$



Term Clustering

Building a similarity matrix

Index Term Selection

- Full text representation of the text
  - All words in the text are index terms
- Alternative: an abstract view of documents
  - Not all words are used as index terms
  - A set of index terms (keywords) are selected
    - Manually by specialists
    - Automatically by computer programs

- Automatic Term Selection
  - **Noun words**: carry most of the semantics
  - **Compound words**: combine two or three nouns in a single component
  - **Word groups**: a set of noun words having a predefined distance in the text

Thesauri

- Definition of the thesaurus
  - A treasury of words consisting of
    - A precompiled list important words in a given domain of knowledge
    - A set of related words for each word in the list, derived from a synonymity (同義) relationship
  - Originally, a thesaurus is a classification scheme composed of words and phrases whose organization aims at facilitating the expression of ideas in written text
  - More complex constituents (phrases) and structures (hierarchies) can be used
    - E.g., the Roget's thesaurus

cowardly *adjective* (膽怯的)
Ignobly lacking in courage: *cowardly turncoats*
Syns: chicken (slang), chicken-hearted, craven, dastardly, faint-hearted, gutless, lily-livered, pusillanimous, unmanly, yellow (slang), yellow-bellied (slang)

Thesauri: Term Relationships

- Relative Terms (RT)
  - Synonyms and near-synonyms
    - Thesauri are most composed of them
  - Co-occurring terms
    - Relationships induced by patterns of within docs

Depend on specific context

- Broader Relative Terms (BT)
  - Like hypernyms (上義詞)
  - A word with a more general sense, e.g., animal is a hypernym of cat
- Narrower Relative Terms (NT)
  - Like hyponyms (下義詞)
  - A word with more specialized meaning, e.g., mare is a hyponym of horse

form a hierarchical structure

automatically or by specialists

Thesauri: Term Relationships

- Example 1:

Figure 1 shows an example of a poset representing geographic locations and sub-locations using a tree structure to show the partial ordering relation.

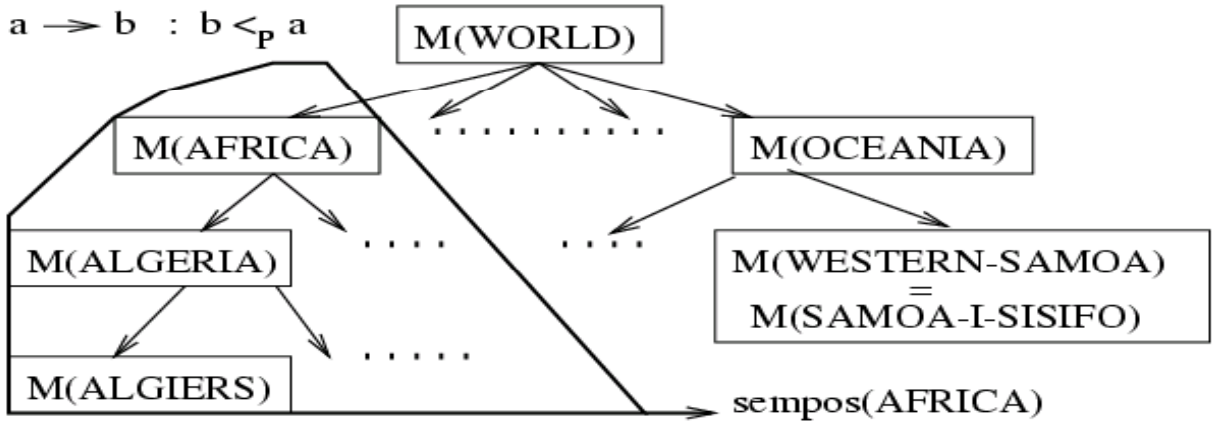


Figure 1: Example of Geographic Semantic Poset

- Example 2: Yahoo presents the user with a term classification hierarchy that can be used to reduce the space to be searched

Thesauri: Purposes

Forskett, 1997

- Provide a standard vocabulary (system for references) for indexing and searching
- Assist users with locating terms for proper query formulation
- Provide classified hierarchies that allow the broadening and narrowing of the current query request according to the needs of the user

Thesauri: Use in IR

- Help with the query formulation process
  - The initial query terms may be erroneous or improper
  - Reformulate the query by further including related terms to it
  - Use a **thesaurus** for assisting the user with the search for related terms

- **Problems**
  - **Local context** (the retrieved doc collection) vs. **global context** (the whole doc collection)
    - Determine thesaurus-like relationships (for local context) at query time
      - Time consuming

Thesauri: Use for the Web

- It is not clear how useful a thesaurus in the context of the Web
  - However, Yahoo! presents the user with a term classification hierarchy for (query) term suggestion
- It is still to early to reach a consensus on the advantages of a thesaurus for the Web
  - Many search engines simply use all the words in the documents as index terms
  - That is, there is no notion of using the concepts of a controlled vocabulary for indexing and search purposes
  - Whether thesaurus-based techniques will flourish in the context of Web remains to be seen