CS630 Representing and Accessing Digital Information

Information Retrieval: Indexing

Thorsten Joachims
Cornell University

Based on slides from Jamie Callan

Information Retrieval

- **Basics**
- **Data Structures and Access**
- **Indexing and Preprocessing**
- **Retrieval Models**

Why Index?

- **An index associates a document with one or more *keys***
  - Present a key, get back the document
- **What keys should be used for a document?**
  - Title, author, id, creation date, …
  - Controlled vocabulary terms
  - Terms from the document ("full-text indexing")
- **Some types of keys support content-based retrieval, others don't**
  - Someone can find an information object without knowing exactly how it is indexed

=> What to choose as indexing terms?

Text Representation

- **Manual indexing**
  - Indexers decide which keywords to assign to document base on *controlled vocabularies*
    - Examples: Libraries, Medline, Yahoo
  - Significant human costs, but no computational costs
- **Automatic indexing**
  - Indexing program assigns words, phrases, or other features
    - Example: words from text of document
      - Remove stopwords
      - Stemming
      - Phrases
    - Example: automatic text classification according to controlled vocabulary
  - Computational cost, but no human cost

Example Document

How aspartame prevents the toxicity of ochratoxin A.

Creppy EE, Baudrimont I, Anne-Marie

Toxicology Department, University of Bordeaux, France.

The ubiquitous mycotoxin ochratoxin A (OTA) is found as a frequent contaminant of a large variety of food and feed and beverage such as beer, coffee and win. It is produced as a secondary metabolite of moulds from Aspergillus and Penicillium genera. Ochratoxin A has been shown experimentally to inhibit protein synthesis by competition with phenylalanine its structural analogue and also to enhance oxygen reactive radicals production. The combination of these basic mechanisms with the unusual long plasma half-life time (35 days in non-human primates and in humans), the metabolisation of OTA into still active derivatives and glutathione conjugate both potentially reactive with cellular macromolecules including DNA could explain the multiple toxic effects, cytotoxicity, teratogenicity, genotoxicity, mutagenicity and carcinogenicity. A relation was first recognised between exposure to OTA in the Balkan geographical

Controlled Vocabularies

Example: Medical Subject Headings (MeSH)

1. Anatomy	**Bacterial Infections**
2. Organisms	**Virus Diseases**
3. Diseases	**Parasitic Diseases**
4. Chemicals and Drugs	**Neoplasms**
5. Analytic, Diagnostic, Therapeutic Tech Equip	**Musculoskeletal Diseases**
6. Psychiatry & Psychology	**Digestive System Diseases**
…	**Stomatognathic Diseases**
14. Health Care	**…**
15. Geographic Locations	**Pathological Conditions**

Controlled Vocabulary Indexing: Example

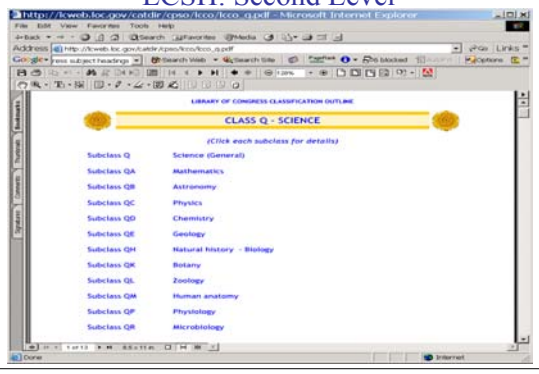
UI	98433165
AU	Creppy EE
AU	Baudrimont I
AU	Anne-Marie
TI	How Aspartame Prevents the Toxicity of Ochratoxin A.
LA	Eng
MH	Animal
MH	Aspartame/*pharmacology
MH	Blood Proteins/metabolism
MH	Cercopithecus aethiops
MH	DNA/drug effects
MH	Human
MH	Mycotoxins/*toxicity

Controlled Vocabulary Indexing

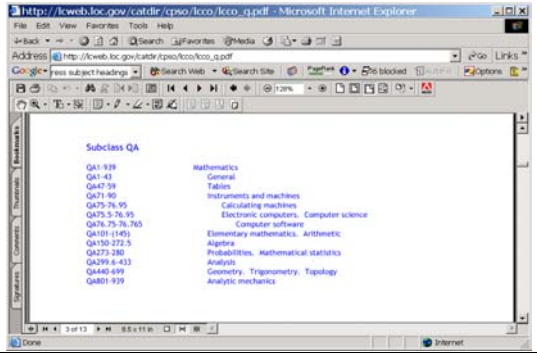
- **There are many controlled vocabularies. None is "best".**
  - Library of Congress Subject Headings (LCSH)
  - Medical Subject Headings (MeSH)
  - …
- **Tradeoffs:** coverage vs. detail
  - Example: LCSH is broad, MeSH is detailed
- **Advantage:** Solves the vocabulary mismatch problem
- **Advantage:** Makes the ontology of a domain explicit
  - Nice for browsing
- **Disadvantage**: Difficult and expensive to create, to use, and to maintain

LCSH: Top Level

LCSH: Second Level



LCSH: Third Level



Full-Text Indexing

Term	TF	Term	TF	Term	TF	Term	TF
the	31	by	6	peptide	4	such	3
of	26	effect	6	several	4	toxic	3
and	22	are	5	toxin	4	vitro	3
in	21	aspartame	5	also	3	when	3
a	15	exposure	5	countries	3	added	2
to	11	human	5	given	3	africa	2
as	9	with	5	it	3	balkan	2
ota	9	animals	4	preventative	3	be	2
for	8	include	4	rate	3	been	2
is	8	ochratoxin	4	shown	3	compound	2

Full-Text Representation Overview

- **Scan for tokens as indexing terms**
- **Parse documents to recognize structure**
- **Stopword removal**
- **Word stemming**
- **Phrase recognition**
- **Concept / feature recognition**

Tokenization

- **Design decisions**
  - Numbers
  - Hyphenation
  - Capitalization
  - Punctuation
  - Special characters
- **Languages such as Chinese and Japanese need segmentation**
- **Record positional information for proximity operators**

Stopword Removal

- **Stopwords: words that are discarded from a document representation**
  - Function words: a, an, and, as, for, in, of, the, to, …
  - About 400 words in English.
  - Other frequent words: "Lotus" in a Lotus Support db
- **Removing stopwords makes some queries difficult to satisfy**
  - Few queries affected, so little effect on *experimental* results
  - But, very **annoying to people**

Full-Text Indexing without Stopwords

Term	TF	Term	TF	Term	TF	Term	TF
ota	9	toxin	4	compound	2	medium	2
effect	6	countries	3	culture	2	mould	2
aspartame	5	given	3	days	2	northern	2
exposure	5	preventative	3	dna	2	phenylalanine	2
humans	5	rate	3	endemic	2	prevent	2
animals	4	toxic	3	food	2	protein	2
include	4	vitro	3	genotoxicity	2	reactive	2
ochratoxin	4	added	2	incidence	2	synthesis	2
peptide	4	africa	2	induce	2	vivo	2
several	4	balkan	2	large	2	weeks	2

Words vs. Phrases vs. Concepts

- **Indexing Term:** General name for any indexing feature
- **Words**
  - word stems
  - N-grams
- **Phrases**
  - Part-of-speech
  - Statistical recognition
  - Examples: "information retrieval", "home run"
- **Concept**
  - Example: "about medicine", "is billing statement"
  - Manual or automatic recognition rules

Stemming

- **Group morphological variants**
  - Plural: "streets" ⇔ "street"
  - Adverbs: "fully" ⇔ "full"
  - Other inflected word forms: "goes" ⇔ "go"
  - Grouping process is called "conflation"
- **More accurate than string matching**
- **Current stemming algorithms make mistakes**
  - Conflating terms manually is difficult, time-consuming
  - Automatic conflation using rules
    - Porter Stemmer
  - Porter stemming example: "police", "policy" => "polic"

Porter Stemming Algorithm

- **Algorithm is based on a set of condition/action rules**
  - `old_suffix->new_suffix`
- **Rules are divided into steps and are examined in sequence**
  - Step 1a: `sses->ss, ies->i, s->NULL`
    - caresses -> caress, ponies -> poni, cats -> cat
  - Step 1b: `if m>0, eed -> ee`
    - Agreed -> agree
- **Many implementations available**
  - http://www.tartarus.org/~martin/PorterStemmer/
- **Good performance on average**

Porter Stemming Example and Problems

- **Original Text**
  Document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals.
- **After Porter stemming and stopwords removal**
  market strateg carr compan agricultur chemic report predict market share chemic report market statist agrochem
- **Problems**
  - Sometimes too aggressive in conflation
    - e.g., policy/police, execute/executive, university/universe
  - Sometimes miss good conflations
    - e.g., European/Europe, matrices/matrix, machine/machinery

Phrases

- **Why use phrases?**
  - "Home run" more precise than "home AND run", "home, run"
  - Sometimes difficult to incorporate in retrieval model
- **Pre-coordinate phrase recognition methods**
  - recognize when document is parsed & indexed
  - Statistically, part-of-speech
  - Recognition costs incurred just once, but not flexible
- **Post-coordinate phrase recognition methods**
  - Recognize when query is evaluated
  - Query operators
  - Recognition costs incurred repeatedly, but very flexible

Phrases: Statistical Recognition

- **Consider all word bigrams**
  - Example: "hit a", "a home", "home run", "run yesterday", …
- **Select by corpus term frequency (*ctf*) or document frequency (*df*)**
  - Remove stopwords
  - Example: "home run" (54), "run yesterday" (1)
- **Reasonably accurate, but makes mistakes**
  - If a pattern occurs often, it is probably a phrase
  - Counter-example: "announced yesterday"
- **Very fast**

Phrases: Part-of-Speech Tagging

- **Assign part of speech tags**
  - Usually with a probabilistic or rule-based part of speech tagger
  - Example: "…hit/v a/art home/n run/n …."
- **Match phrases by POS patterns**
  - Example: N+, AN+
- **More accurate (maybe)**
  - N+: "home run"
  - AN+: "white house"
  - AN+: "big home run" (is this a good phrase?)
- **Reasonably fast, but slower than statistical recognition**

Part-of-Speech Phrases TREC Example

- 65,824 United States
- 61,327 Article Type
- 33,864 Los Angeles
- 18,062 Hong Kong
- 17,788 North Korea
- 17,308 New York
- 15,513 San Diego
- 15,009 Orange County
- 12,869 prime minister
- 12,067 Soviet Union
- 10,811 Russian Federation
- 9,912 United Nations
- 8,127 Southern California
- 7,640 South Korea
- 7,620 end recording
- 7,524 European Union
- 7,086 news conference
- 6,792 City Council
- 6,348 Middle East
- 6,157 peace process
- 5,955 human rights
- 5,837 White House
- 5,778 long time
- 5,776 Armed Forces
- 5,636 Santa Ana
- 5,619 Foreign Ministry
- 5,527 Bosnia-Herzegovina
- 5,458 words indistinct
- 5,452 international community
- 5,443 vice president
- 5,247 Security Council
- 5,098 North Korean

Thesaurus

- **Resolve synonyms**
  - Query for “computer science”, document uses “informatics”
  - Standardized terms => controlled vocabulary
  - Index documents by synset
- **Resolve other relationship**
  - More general, more special terms
  - Broadening/narrowing the search
- **Query expansion**
  - Interactive by user
  - Automatic addition of terms
- **Wordnet**
  - http://www.cogsci.princeton.edu/~wn/

Full-Text Indexing Summary

- **Task: convert the document into a set of indexing terms**
- **Issues and Design Decisions:**
  - **Tokens:** AT&T, drive-in, 527-4701, $1,110,427, …
  - **Stopwords:** Why remove stopwords? How are stopwords defined?
  - **Stemming:** Why stem? How do stemming algorithms work?
  - **Phrases:** Why index phrases? How are phrases recognized?
  - **NLP:** What role can/should it play?
  - **Features:** Why index features? How are features recognized?
  - **Structure:** Why index structure? How is it stored and used?
  - **Efficiency:** Memory, disk space, *speed.*