Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
Άνοιξη 2009

HY463 - Συστήματα Ανάκτησης Πληροφοριών
Information Retrieval (IR) Systems

Γλώσσες Επερώτησης για Ανάκτηση Πληροφοριών

Κεφάλαιο 4

Γλώσσες Επερώτησης για Ανάκτηση Πληροφοριών

- Επερωτήσεις λέξεων (Keyword-based Queries)
  - Μονολεκτικές επερωτήσεις (Single-word Queries)
  - Επερωτήσεις φυσικής γλώσσας (Natural Language Queries)
  - Boolean Επερωτήσεις (Boolean Queries)
  - Επερωτήσεις Συμφραζομένων (Context Queries)
    - Φραστικές Επερωτήσεις (Phrasal Queries)
    - Επερωτήσεις Εγγύτητας (Proximity Queries)
- Ταίριασμα Προτύπου (Pattern Matching)
  - Απλό (Simple)
  - Ανεκτικές σε ορθογραφικά λάθη (Allowing errors)
    - Levenstein distance, LCS longest common subsequence
  - Κανονικές Εκφράσεις (Regular expressions)
- Δομικές Επερωτήσεις (Structural Queries)
  - *(θα καλυφθούν σε επόμενο μάθημα)*
- Πρωτόκολλα επερώτησης (Query Protocols)

Γλώσσες Επερώτησης για Ανάκτηση Πληροφοριών
Εισαγωγή

- Ο τύπος των επερωτήσεων που επιτρέπονται σε ένα σύστημα εξαρτάται σε ένα βαθμό και από το Μοντέλο Ανάκτησης που χρησιμοποιεί το σύστημα
  - Boolean model => boolean queries
  - Extended Boolean model => boolean queries (…)
  - Vector Space model => natural language queries (free text)
  - Probabilistic model => natural language queries
  - ...
- Retrieval unit: basic unit that can be returnes as the answerto a query
- Protocol: not for end users
- Εδώ θα δούμε τύπους επερωτήσεων χρήσιμους για την ανάκτηση πληροφοριών.
  - Αργότερα θα δούμε τις δομές δεδομένων και αλγόριθμους για την αποτίμησή τους.

Επερωτήσεις φυσικής γλώσσας
(“Natural Language” Queries)

- Keyword queries

Popular:

- Intuitive
- Simple to express
- Fast ranking

Boolean Queries

- Keywords combined with Boolean operators:

Atoms combined with Boolean operators

- OR: (e_1 OR e_2)
- AND: (e_1 AND e_2)
- BUT: (e_1 BUT e_2) Satisfy e_1 but **not** e_2

Compositional -> query syntax tree

- Negation only allowed using BUT to allow efficient use of inverted index by filtering another efficiently retrievable set.
- Naïve users have trouble with Boolean logic.
- Also, sort by some criteria and highlight the occurrences of the query words

Επερωτήσεις φυσικής γλώσσας
("Natural Language" Queries)

- Full text queries as arbitrary strings.
- Typically just treated as a **bag-of-words** for a vector-space model.
- Typically processed using standard vector-space retrieval methods.
- A whole document may be considered as a query

Context-Queries

- Ability to search words in a given *context*, that is, near other words
- Types of Context Queries
  - **Phrasal Queries**
  - **Proximity Queries**

Context-Queries

Phrasal Queries

- Retrieve documents with a specific phrase (**ordered** list of contiguous words)
  - "information theory"
  - "to be or not to be"
- May allow intervening stop words and/or stemming.
  - For example, "**buy camera**" matches:
    - "buy a camera",
    - "buy a camera", (two spaces)
    - "buying the cameras" etc.

Proximity Queries (Επερωτήσεις Εγγύτητας)

- List of words with specific maximal distance constraints between words.
- For example:
  - **"dogs" and "race" within 4 words**
- will match
  - "…dogs will begin the race…"
- May also perform stemming and/or not count stop words.
- The order may or may not be important

Pattern Matching

- Allow queries that match strings rather than word tokens.
- Requires more sophisticated data structures and algorithms than inverted indices to retrieve efficiently.

Some types of simple patterns:

- **Prefixes**: Pattern that matches start of word.
  - "anti" matches "antiquity", "antibody", etc.
- **Suffixes**: Pattern that matches end of word:
  - "ix" matches "fix", "matrix", etc.
- **Substrings**: Pattern that matches arbitrary subsequence of characters.
  - "rapt" matches "enrapture", "velociraptor" etc.
- **Ranges**: Pair of strings that matches any word lexicographically (alphabetically) between them.
  - "tin" to "tix" matches "tip", "tire", "title", etc.

More Complex Patterns: Allowing Errors

- What if query or document contains typos or misspellings?
- Judge similarity of words (or arbitrary strings) using:
  - **Edit distance (Levenstein distance)**
  - **Longest Common Subsequence (LCS)**
- Allow proximity search with bound on string similarity.

Edit (Levenstein) Distance

- Minimum number of character *deletions*, *additions,* or *replacements* needed to make two strings equivalent.
  - "misspell" to "mispell" is distance 1
  - "misspell" to "mistell" is distance 2
  - "misspell" to "misspelling" is distance 3
- Can be computed efficiently using *dynamic programming*
  - O(*mn*) time where *m* and *n* are the lengths of the two strings being compared.

Longest Common Subsequence (LCS)

- Length of the longest subsequence of characters shared by two strings.
- A *subsequence* of a string is obtained by deleting zero or more characters.
- Examples:
  - "misspell" to "mispell" is 7
  - "misspelled" to "misinterpretted" is 7
    "mis…p…e…ed"

More complex patterns: Regular Expressions

- Language for composing complex patterns from simpler ones.
  - An individual character is a regex.
  - Union: If e_1 and e_2 are regexes, then $(e_1 \mid e_2)$ is a regex that matches whatever either e_1 or e_2 matches.
  - Concatenation: If e_1 and e_2 are regexes, then $e_1 e_2$ is a regex that matches a string that consists of a substring that matches e_1 immediately followed by a substring that matches e_2
  - Repetition (Kleene closure): If e_1 is a regex, then e_1^* is a regex that matches a sequence of zero or more strings that match e_1

Regular Expression Examples

- **(u|e)nabl(e|ing)** matches
  - unable
  - unabling
  - enable
  - enabling
- **(un|en)***able matches
  - able
  - unable
  - unenable
  - enununenable

Enhanced Regex's (Perl)

- Special terms for common sets of characters, such as alphabetic or numeric or general "wildcard".
- Special repetition operator (+) for 1 or more occurrences.
- Special optional operator (?) for 0 or 1 occurrences.
- Special repetition operator for specific range of number of occurrences: {min,max}.
  - A{1,5} One to five A's.
  - A{5,} Five or more A's
  - A{5} Exactly five A's

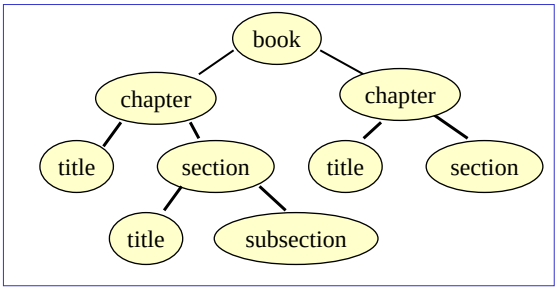
Perl Regex's

- Character classes:
  - \w (word char) Any alpha-numeric (not: \W)
  - \d (digit char) Any digit (not: \D)
  - \s (space char) Any whitespace (not: \S)
  - . (wildcard) Anything
- Anchor points:
  - \b (boundary) Word boundary
  - ^ Beginning of string
  - $ End of string
- Examples
  - U.S. phone number with optional area code:
    - /\b(\(\d{3}\)\s?)?\d{3}-\d{4}\b/
  - Email address:
    - /\b\S+@\S+(\.com|\.edu|\.gov|\.org|\.net)\b/

    Note: Packages available to support Perl regex's in Java

Δομικές Επερωτήσεις (Structural Queries)

- Εδώ τα έγγραφα έχουν **δομή** που μπορεί να αξιοποιηθεί κατά την ανάκτηση
- Η δομή μπορεί να είναι:
  - Ένα προκαθορισμένο σύνολο πεδίων
    - title, author, abstract, etc.
  - Δομή Hypertext
  - Μια ιεραρχική δομή
    - Book, Chapter, Section, etc.



Θα τις μελετήσουμε αναλυτικά σε μια άλλη διάλεξη

Query Protocols

- They are not intended for final users.
- They are query languages that are used automatically by software applications to query text databases. Some of them are proposed as standard for querying CD-ROMs or as intermediate languages to query library systems
- Query Protocols
  - Z39.50
    - 1995 standard ANSI, NISO
    - bibliographical information
    - **SRW (Search and Retrieve Web Service): Extension of Z39.50 using Web Technologies. Queries in CQL**
  - WAIS (Wide Area Information Service)
    - used before the Web
  - Dienst Protocol
  - For CD-ROMS
    - CCL (Common Command Language)
      - 19 commands. Based on Z39.50
    - CD-RDx (Compact Disk Read only Data Exchange)
    - SFQL (Structured Full-text Query Language)

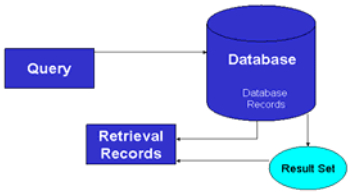
SFQL

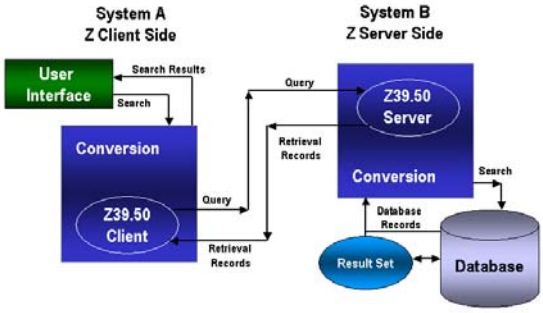
- **SFQL** (Structured Full-text Query Language)
  - Relational database query language SQL enhanced with “full text” search.
  - Παράδειγμα:

```
select abstract
from journal.papers
where   author contains “Teller” and
        title contains “nuclear fusion” and
        date < 1/1/1950
```

- Supports Boolean operators, thesaurus, proximity operations, wild cards, repetitions.
  - It is old, but just indicatively let’s have a look

Z39.50





CQL (Common Query Language)

- **A formal language for representing queries to information retrieval systems – Now: CQL-> Context Query Language**
- **Human-readable**

http://www.loc.gov/standards/sru/specs/cql.html

CQL (Common Query Language)

Search clause

- Always includes a term
  - simple terms consist of one or more words
- May include index name (i.e. field name)
  - To limit search to a particular field/element
  - Index name includes base name and may include prefix
    - title, subject
    - dc.title, dc.subject
  - Several index sets have been defined (called Context Sets in SRW)
    - dc
    - bath
    - srw
  - Context set defines the available indexes for a particular application

CQL

- Relation
  - <, >, <=, >=, =, <>
  - exact used for string matching
  - all when term is list of words to indicate all words must be found
  - any when term is list of words to indicate any words must be found
- Boolean operators: and, or, not
- Proximity (prox operator)
  - relation (<, >, <=, >=, =, <>)
  - distance (integer)
  - unit (word, sentence, paragraph, element)
  - ordering (ordered or unordered)
- Masking rules and special characters
  - single asterisk (*) to mask zero or more characters
  - single question mark (?) to mask a single character
  - carat/hat (^) to indicate anchoring, left or right

CQL Examples

- **Simple queries:**
  - dinosaur
  - "the complete dinosaur"
- **Boolean**
  - dinosaur and bird or dinobird
  - "feathered dinosaur" and (yixian or jehol)
- **Proximity**
  - foo prox bar
  - foo prox/>/4/word/ordered bar
- **Indexes**
  - title = dinosaur
  - bath.title="the complete dinosaur"
  - srw.serverChoice=dinosaur
- **Relations**
  - year > 1998
  - title all "complete dinosaur"
  - title any "dinosaur bird reptile"
  - title exact "the complete dinosaur"