Information Retrieval

Overview

	IE (Data base)	IR (set of result docs)
data/documents	structured	unstructured
attribute semantics	defined	ambiguous
queries	well defined	Can be free text/arbitrary
retrieval	exact	imprecise

Procedure IR:

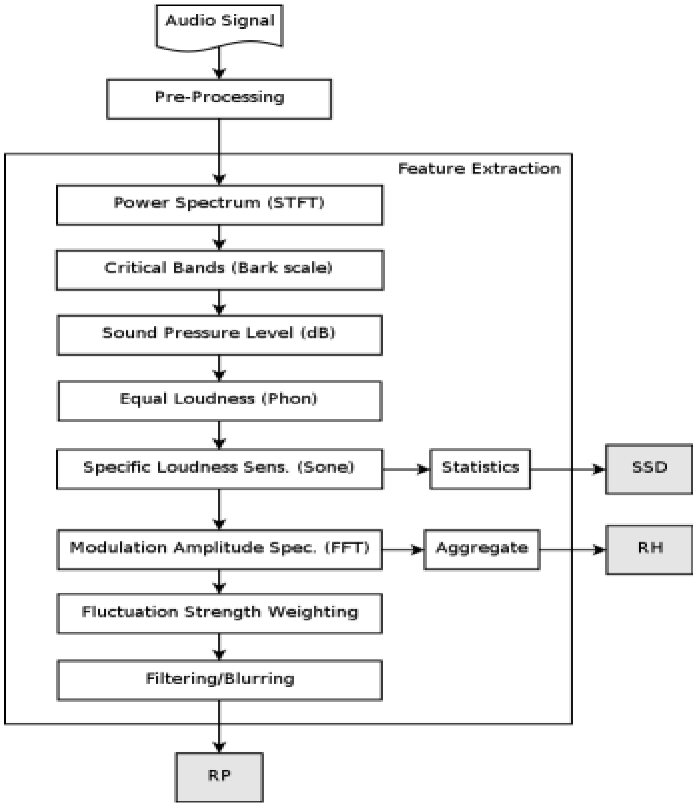
- Pre-Processing
    - Collection Cleansing
        - stop-words
        - formatting information
        - empty/ill-formated docs
    - identify relevant doc parts for indexing (text, image, meta data, ...)
    - stemming (e.g. Porter Stemmer)
    - stop-word removal (manually created stop-word list)
- Indexing (Bag of words)
    - select type of terms used (feature set selection)
        - n-grams
        - words (word stems)
        - word co-occurences (word n-grams) → detect phrases
        - concepts (Date, Person, Company, ...) → NLP
    - and weight the terms → term weighting (feature extraction)
        - df (document frequence) of term to select terms
        - tf (term frequence) inside a document
        - tfidf (tf inverse df) = tf/df oder tfidf = tf/(ln(N/df))
        -
        - Zipf's law (relates term frequency to rank)
        - Heap's law (Predict number of distinct terms)
        -
- Retrieval

- find documents satisfying query
- Retrieval Models
  - Boolean model (Exact-Match)
    - canonicalization until result is satisfactory
    - detailed knowledge of doc domain needed
  - Vector Space Model (Best-Match)
    - normalized high-dimensional feature space (indexing)
    - query = vector
    - result is docs that are closed to high-dimensional query vector
      - similarity by L1, L2, minkovsky, or cosine similarity (distance measure)
  - Probabilistic Model
    - 2-class classification (relevant or not)
    - e.g. Bayes statistics
- Relevance Feedback
  - iterative interactive retrieval to refine query (e.g. add terms)
    - manual refinement (blind – user define relevant result docs)
    - semi-automatic refinement
    - automatic refinement (see ATC) → e.g. rocchio feedback: add terms from relevant and substract terms form irrelevant docs
- ATC – Automatic Text Classification
  - Machine Learning
    - knn-classifiers
    - decision trees
    - rocchio
    - naive bayes
    - support vector machines
    - clustering (SOM, etc.)
  - feature space dimensionality reduction
    - feature selection vs. Extraction
    - local vs. Global
- Evaluation
  - Contingency table
  - Measures
    - precesion
    - recall
    - accuracy/error

        - F-measure

Music IR

- What is Music?
    - Sound
        - Nyquist sampling theorem
        - lossless/lossy sound formats
        - PCM (Pulse Code Modulation) → digital representation of analog signal
        - MIDI (Musical Instrument Digital Interface)
        - Scores/Sheet-Music (by hand → many styles, Printed, MusicXML, (e.g. Lily Pond), ...)
    - Text
    - Community Data
    - Images/Videos
- Web Music Retrieval
    - music search engines
    - centralized/de-centralized/hybrid P2P
- Audio features
    - MPEG7-Standard Features
    - Marsyas System
    - Rhythm Patterns/Rhythm Histograms/Statistical Spectrum Descriptor


Audio Signal
Pre-Processing
Feature Extraction
Power Spectrum (STFT)
Critical Bands (Bark scale)
Sound Pressure Level (dB)
Equal Loudness (Phon)
Specific Loudness Sens. (Sone)
Statistics
SSD
Modulation Amplitude Spec. (FFT)
Aggregate
RH
Fluctuation Strength Weighting
Filtering/Blurring
RP

- Evaluation/Benchmarking of Music classification (clustering/similarity based) and Music IR
  - E.g. MIREX (besides many others)
- Application Example
  - SOM Player using clustering
  - online (client-based) audio feature extraction
  - audio segmentation - lead-in, verse, chorus, etc. (→ structure e.g.: ABCBDBAC‘A) & e.g. k-means clustering
  - chord detection
  - instrument separation using template matching → artificial music

Information Extraction (IE)

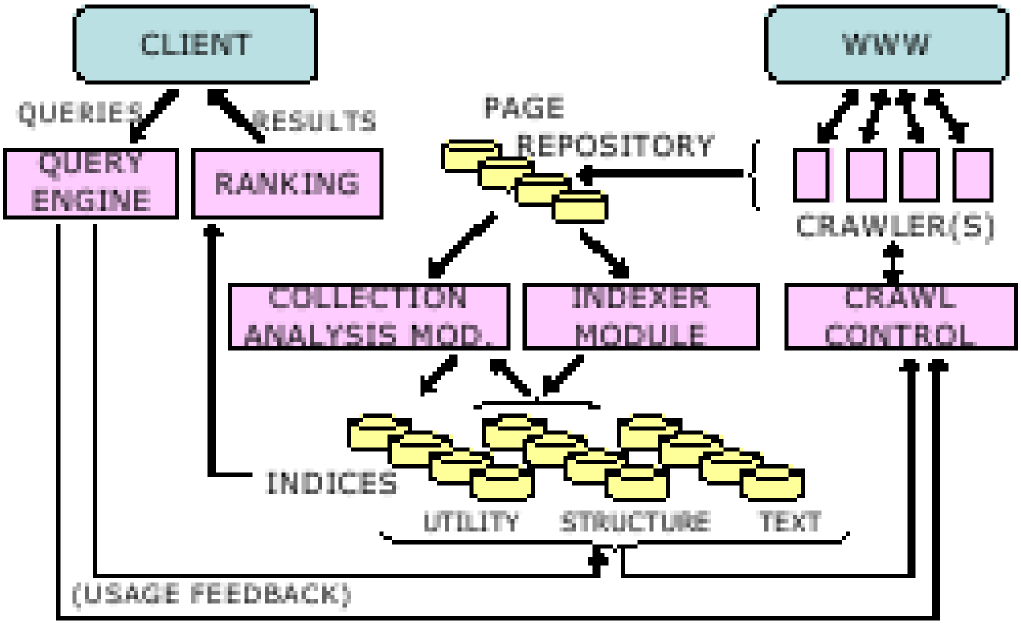
- Application Areas
  - Database population → Filling slots in a database from sub-segments of text
  - Ontology population/evolution
  - Automatic summarization - abstract/extract main ideas in less space
    - Input
      - singe- VS multi-document summarization
      - mono- VS multi- VS cross-lingual
      - text VS multimedia
    - Purpose
      - informative VS indicative(user-interaction)
      - generic VS user-orientated(based on user query)
      - domain-specific VS general (portable to all domains)
    - Output – Quality is crucial
      - Extract(sentences, paragraphs, ...)
        - machine learning/language processing (e.g. tree generation)
        - Edmundsonian Paradigm (ranking of sentences) → can include post-processing to eliminate incoherences (anaphora, semantic gaps, etc..)
      - Abstract (scripts & simple generation VS MUC-like concepts & NLG)
        - prior knowledge (semantic structure known → ontology)
        - clustering
    - Evaluation
      - intrinistic → summary's purpose/task is neglected
        - sentence integrity (anaphor without reference)
        - readability
        - fidelity

      - human „gold“ summary → precision & recall & etc.
    - extrinistic → account for purpose/task
      - relevance (compare to topic(s) assigned by human judges)
      - comprehensibility (judges are asked questions on studied text)
- Question answering (natural language answering)
  - Dimensions
    - nature of information (DB ↔ free text)
    - nature of questions (facts ↔ opinions)
    - nature of answers (extracted ↔ [NLP-]generated)
    - nature of technique (linguistically correct/sophisticated ↔ linguistically uninformed (e.g. n-grams)
  - Components
    - Analyze question
    - Gather Information
    - Distill Answer
    - Sanity check
    - Present Answer
- Web Search (mostly ranked list, no precise answer)
  - Components
    - Crawler (Robots, Spiders)
      - Page selection based on
        - Factors
          - Coverage
          - Quality (indexing good pages)
          - Efficiency (no duplicates)
          - Etiquette (minimize overloaded server loads)
          - Freshness/Age (page life-time)
            - Batch Mode Crawler (periodic update)
            - Steady Crawler (incremental update)
              - keep local collection fresh
              - continuously improve collection's value
            - determined with fixed/variable frequency (based on page's rate of change)
            - in-place update VS shadowing
        - Mathematical Model
          - Importance Metrics

            - Interest-Driven
            - Popularity-Driven
            - Location-Driven
        - Quality Metrics
            - performance of a crawler is described
        - Ordering Metrics
- Page repository (scalable storage system – local copy of the web)
    - Interface to
        - Crawler
        - Indexer Module
        - Query Engine
    - Storage Manager
        - distribute Pages over available storage nodes
            - uniform distribution policy
            - hash distribution policy
            - log-structured policy (B*-tree index)
            -
        - handles updates
        - handles different access modes (stream, random access, ...)
- Indexer (uses statistician during merging OR during flushing)
    - Indices Types
        - Utility Index (Search engine specific info → speed up!) - e.g. rank, site index, ...
        - structure/link index (graph-modelled → VERY large graph representing the links)
        - text index=inverted file (identify and select pages) - e.g. rate of change, anchors, headings, ...
            - → is a mapping of content to its location!!!
                - local
                - global
    - Parallel Processing
    -
- Ranking
    - Extended Boolean Models
        - tf, headings, titles, keywords,...
        - idf, word count,...

- Content-based (Vector or Probabilistic Model)
- human annotation
- Factors
  - ad-hoc factors (porn filter...)
  - popularity
  - text anchor
  - LINK ANALYSIS (find pages with high authority (HUBS) → Assumption: good pages link to good pages)
    - e.g. Hypertext Induced Topic Selection(HITS) Algorithm:
      - generate query-independent sub- graph of THE web graph
      - recursively calculate hubs and authorities –> refine graph
  - Bibliometric law (often cited articles hae high scientific value)
- Problems
  - Rank sink
  - Rank leak

- Architecture



- 3rd Generation Web Search Engines
  - semantic analysis (what does it mean?)
  - determine user context rather than analyze query
    - user location
    - previous queries
    - user profile
    - spell checking

        - query suggestion
- General Procedure
    - segmentation (select relevant words/terms/phrases)
    - classification (noun, verb, concept, ...)
    - clustering (group classified data by co-referencing detection – e.g. group = „Gerhardt works Apple“)
    - association (fill groups into DB)
- Architecture
    - Tokenization (for text sectioning and filtering → see indexing)
    - Lexicon and Morphology (for e.g. maximum entropy POS Tagging) → Named-Entity Recognition (e.g. „Apple Computer Inc.“, „Sepp Maier“)
        - Combining Morphemes:
            - free morphemes → stand alone words
            - bound morphemes → 'tion' ind 'creation
            - inflectional morphemes → big, big-'ger', big-'gest'
            - derivational morphemes → verb + 'ment' == noun
        - choice of Morphology or Lexicon
            - language dependent
            - domain-specific (medicine, chemistry, literature, ...)
    - Parsing
        - find grammatical structure (also phrases, etc.)
        - and for Co-reference resolution:
            - name-aliases
            - pronoun-antecedents
            - using definite description like ontology
    - Domain-Specific analysis (to merge partial results using detected co-referencing)
        - using templates consisting of slots (ontologies
        - Approaches
            - atomic approach
                - intelligent guessing → high recall – low precision
                - precision improved by filering
            - molecular approach (more popular)
                - small amount of highly reliable rules try to match all arguments to pattern/template/ontology → high precision – low recall
                - iteratively generalizes rules to cover other patterns/templates → Over-generation possible with leads to lower precision and higher recall
- Wrappers

  - rigorous (unified format, complete info)
  - semi-rigorous (unified format, incomplete info)
  - semi-relaxed (non-unified format, complete info)
  - relaxed (non-unified format, incomplete info)
- Knowledge Eningeering
  - pre-processing (segmentation, filtering, see indexing)
  - Analysis (parsing, semantic interpretation – e.g. through co-references)
  - post-processing (chose templates/ontologies and categories to map)
  - Example models:
    - FASTUS (Finite State Automation Text Understanding System)
    - GENLTOOLSET
    - PLUM (Probabilistic Language Understanding Model)
    - PROTEUS
- Machine Learning
  - Supervised learning
    - Autoslog
      - extract dictionary of concepts
      - single slot extraction
    - PALKA
      - new rules are generalized
      - Existing rules are specialized
    - WHISK
      - REGEXP in top-down induction
    - RAPIER (Robust Automated Production of Extraction Rules)
      - Input: already filled templates
      - Output: Pattern-Matching rules
      - bottom-up
    - GATE (General Architecture for Text Engineering)
    - WIEN (Wrapper Induction Environment)
    - bottom-up
    - Lixto
      - Supervised Wrapper Generation Program
  - Semi-supervised learning
    - Bootstrapping
  - Unsupervised learning