An Overview of Information Retrieval

Nov. 10, 2009

Maryam Karimzadehgan

mkarimz2@illinois.edu

Department of Computer Science

University of Illinois, Urbana-Champaign

Outline

- **Limitations of Database systems (Motivation for IR systems)**
- **Information Retrieval**
  - **Indexing**
  - **Similarity Measures**
  - **Evaluation**
  - **Other IR applications**
- **Web Search**
  - **PageRank Algorithm**
- **News Recommender system on Facebook**

A (Simple) Database Example

Student Table

Student ID	Last Name	First Name	Department ID	email
1	Maryam	Karimzadehg	CS	mkarimz2@uiuc.edu
2	Peters	jordan	EE	kj@uiuc.edu
3	Smith	Chris	CE	sc@uiuc.edu
4	Smith	John	CLIS	Sj@uiuc.edu

Department Table

Department ID	Department
EE	Electrical Engineering
CE	Computer Engineering
CLIS	Information Studies

Course Table

Course ID	Course Name
lbsc690	Information Technology
ee750	Communication
ce098	Computer Architecture

Enrollment Table

Student ID	Course ID	Grade
1	lbsc690	90
1	ee750	95
2	lbsc690	95
2	hist405	80
3	hist405	90
4	lbsc690	98

Databases vs. IR

- **Format of data:**
  - **DB: Structured data. Clear semantics based on a formal model.**
  - **IR: Mostly unstructured. Free text.**
- **Queries:**
  - **DB: Formal (like SQL)**
  - **IR: often expressed in natural language (keywords search)**
- **Result:**
  - **DB: exact result**
  - **IR: Sometimes relevant, often not**

Short vs. Long Term Info Need

- **Short-term information need (Ad hoc retrieval)**
  - "Temporary need"
  - Information source is relatively static
  - User "pulls" information
  - Application example: library search, Web search
- **Long-term information need (Filtering)**
  - "Stable need", e.g., new data mining algorithms
  - Information source is dynamic
  - System "pushes" information to user
  - Applications: news filter

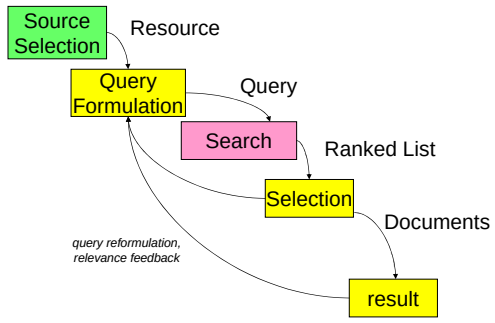
What is Information Retrieval?

- Goal: Find the documents most relevant to a certain query (information need)
- Dealing with notions of:
  - Collection of documents
  - Query (User's information need)
  - Notion of Relevancy

What Types of Information?

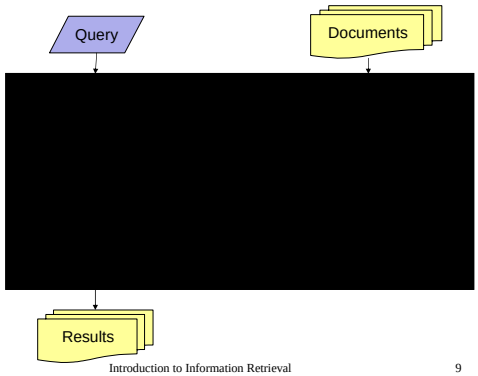
- Text (Documents)
- XML and structured documents
- Images
- Audio (sound effects, songs, etc.)
- Video
- Source code
- Applications/Web services

The Information Retrieval Cycle



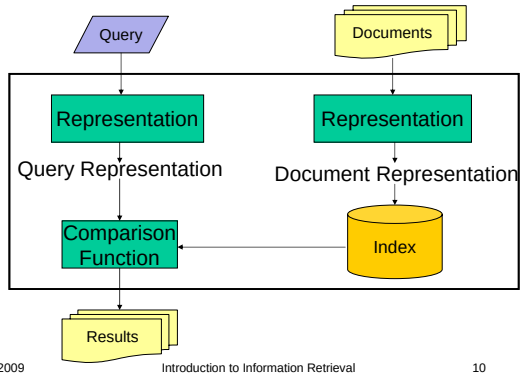
Slide is from Jimmy Lin's tutorial

The IR Black Box



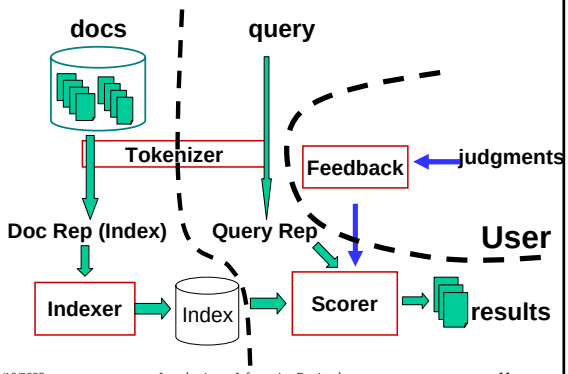
Slide is from Jimmy Lin's tutorial

Inside The IR Black Box



Slide is from Jimmy Lin's tutorial

Typical IR System Architecture

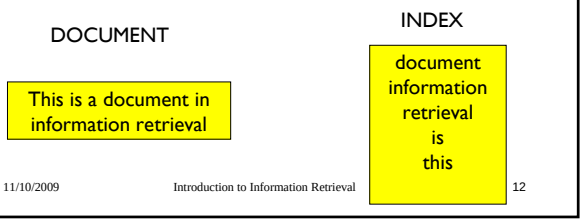


Slide is from ChengXiang Zhai's CS410

1) Indexing

- Making it easier to match a query with a document
- Query and document should be represented using the same units/terms

Bag of Word Representation



What is a good indexing term?

- **Specific (phrases) or general (single word)?**
- **Words with middle frequency are most useful**
  - **Not too specific (low utility, but still useful!)**
  - **Not too general (lack of discrimination, stop words)**
  - **Stop word removal is common, but rare words are kept in modern search engines**
    - **Stop words are words such as:**
      - *a, about, above, according, across, after, afterwards, again, against, albeit, all, almost, alone, already, also, although, always, among, as, at*

Inverted Index

Doc 1

This is a sample document with one sample sentence

Doc 2

This is another sample document

Dictionary

Term	# docs	Total freq
This	2	2
is	2	2
sample	2	3
another	1	1
…	…	…

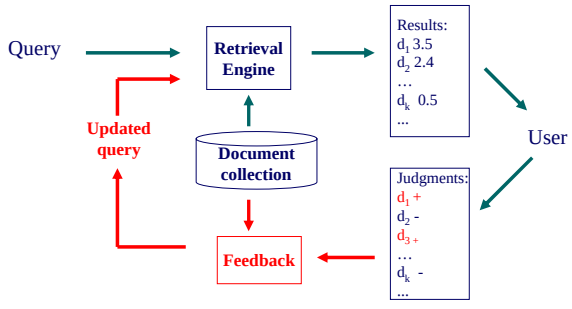
Postings

Doc id	Freq
1	1
2	1
1	1
2	1
1	2
2	1
2	1
…	…
…	…

2) Tokenization/Stemming

- **Stemming: Mapping all inflectional forms of words to the same root form, e.g.**
  - **computer -> compute**
  - **computation -> compute**
  - **computing -> compute**
- **Porter's Stemmer is popular for English**

3) Relevance Feedback



Slide is from ChengXiang Zhai's CS410

4) Scorer/Similarity Methods

1) Boolean model
2) Vector-space model
3) Probabilistic model
4) Language model

Boolean Model

- Each index term is either present or absent
- Documents are either Relevant or Not Relevant(no ranking)
- Advantages
  - Simple
- Disadvantages
  - No notion of ranking (exact matching only)
  - All index terms have equal weight

Vector Space Model

- Query and documents are represented as vectors of index terms
- Similarity calculated using COSINE similarity between two vectors
  - Ranked according to similarity

TF-IDF in Vector Space model

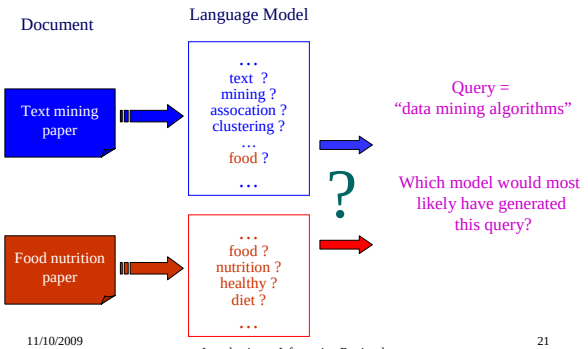
$$tfidf_{i,k} = f_{i,k} * \log\left(\frac{N}{df_i}\right)$$

TF part

IDF part

IDF: A term is more discriminative if it occurs only in fewer documents

Language Models for Retrieval



Slide is from ChengXiang Zhai's CS410

Retrieval as Language Model Estimation

- **Document ranking based on *query likelihood***

$$\log p(q \mid d) = \sum_i \log p(w_i \mid d)$$

where, $q = w_1 w_2 \ldots w_n$

Document language model

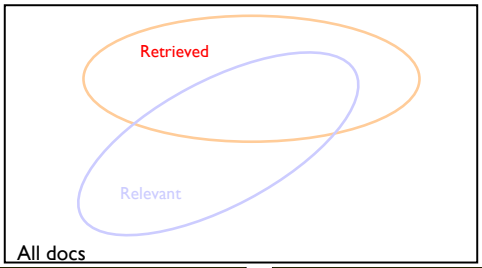
- Retrieval problem ≈ Estimation of $p(w_i|d)$
  - Smoothing is an important issue, and distinguishes different approaches

Slide is from ChengXiang Zhai's CS410

Information Retrieval Evaluation

- **Coverage of information**
- **Form of presentation**
- **Effort required/ease of Use**
- **Time and space efficiency**
- **Recall**
  - **proportion of relevant material actually retrieved**
- **Precision**
  - **proportion of retrieved material actually relevant**

Precision vs. Recall



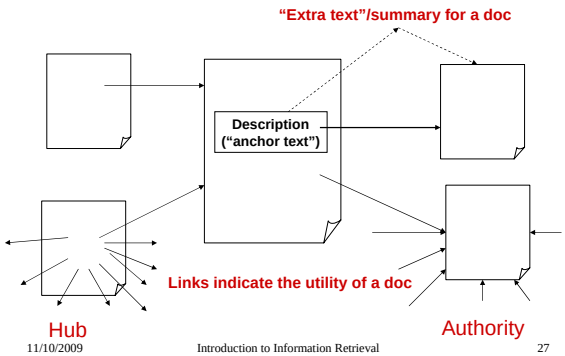
$$\text{Recall} = \frac{|\text{RelRetrieved}|}{|\text{Rel in Collection}|} \qquad \text{Precision} = \frac{|\text{RelRetrieved}|}{|\text{Retrieved}|}$$

Web Search – Google PageRank Algorithm

Characteristics of Web Information

- "Infinite" size
  - Static HTML pages
  - Dynamically generated HTML pages (DB)
- Semi-structured
  - Structured = HTML tags, hyperlinks, etc
  - Unstructured = Text
- Different format (pdf, word, ps, …)
- Multi-media (Textual, audio, images, …)
- High variances in quality (Many junks)

Exploiting Inter-Document Links



Slide is from ChengXiang Zhai's CS410

PageRank: Capturing Page "Popularity"

[Page & Brin 98]

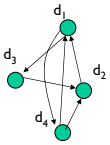
- Intuitions
  - Links are like citations in literature
  - A page that is cited often can be expected to be more useful in general
  - Consider "indirect citations" (being cited by a highly cited paper counts a lot…)
  - Smoothing of citations (every page is assumed to have a non-zero citation count)

The PageRank Algorithm (Page et al. 98)

Random surfing model: At any page,
With prob. α, randomly jumping to a page
With prob. (1-α), randomly picking a link to follow.

M_{ij} = probability of going from d_i to d_j

"Transition matrix"



$$M = \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{bmatrix}$$

N= # pages

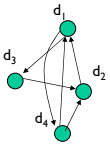
p(di): PageRank score (average probability of visiting page di);

l_{ij} = 1/N

Initial value p(d)=1/N, iterate until converge

Slide is from ChengXiang Zhai's CS410

PageRank: Example



$$p(d_j) = \sum_{i=1}^{N} [\tfrac{1}{N}\alpha + (1-\alpha)M_{ij}]p(d_i)$$

$$\vec{p} = (\alpha I + (1-\alpha)M)^T \vec{p}$$

$$A = (1-0.2)M + 0.2I = 0.8 \times \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{bmatrix} + 0.2 \times \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

$$\begin{bmatrix} p^{n+1}(d_1) \\ p^{n+1}(d_2) \\ p^{n+1}(d_3) \\ p^{n+1}(d_4) \end{bmatrix} = A^T \begin{bmatrix} p^n(d_1) \\ p^n(d_2) \\ p^n(d_3) \\ p^n(d_4) \end{bmatrix} = \begin{bmatrix} 0.05 & 0.85 & 0.05 & 0.45 \\ 0.05 & 0.05 & 0.85 & 0.45 \\ 0.45 & 0.05 & 0.05 & 0.05 \\ 0.45 & 0.05 & 0.05 & 0.05 \end{bmatrix} \times \begin{bmatrix} p^n(d_1) \\ p^n(d_2) \\ p^n(d_3) \\ p^n(d_4) \end{bmatrix}$$

Initial value p(d)=1/N, iterate until converge

Beyond Just Search – Information Retrieval Applications

Examples of Text Management Applications

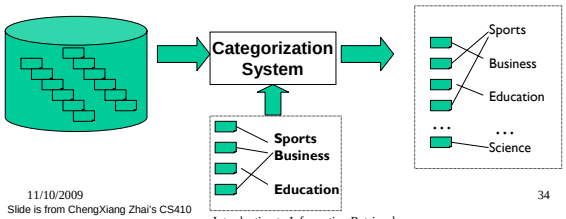
- **Search**
  - **Web search engines (Google, Yahoo, …)**
  - **Library systems**
  - …
- **Filtering**
  - **News filter**
  - **Spam email filter**
  - **Literature/movie recommender**
- **Categorization**
  - **Automatically sorting emails**
  - …
- **Mining/Extraction**
  - **Discovering major complaints from email in customer service**
  - **Business intelligence**
  - **Bioinformatics**
  - …

Sample Applications

- 1) Text Categorization
- 2) Document/Term Clustering
- 3) Text Summarization
- 4) Filtering

1) Text Categorization

- Pre-given categories and labeled document examples (Categories may form hierarchy)
- Classify new documents
- A standard supervised learning problem



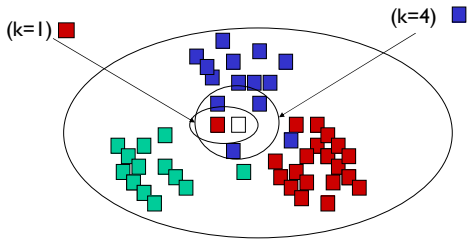
Slide is from ChengXiang Zhai's CS410

K-Nearest Neighbor Classifier

- Keep all training examples
- Find k examples that are most similar to the new document ("neighbor" documents)
- Assign the category that is most common in these neighbor documents (neighbors vote for the category)
- Can be improved by considering the distance of a neighbor (A closer neighbor has more influence)
- Technical elements ("retrieval techniques")
  - **Document representation**
  - **Document distance measure**

Slide is from ChengXiang Zhai's CS410

Example of K-NN Classifier



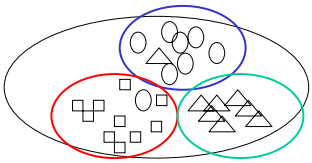
Slide is from ChengXiang Zhai's CS410

Examples of Text Categorization

- **News article classification**
- **Meta-data annotation**
- **Automatic Email sorting**
- **Web page classification**

2) The Clustering Problem

- Group similar objects together
- Object can be document, term, passages
- Example



Slide is from ChengXiang Zhai's CS410

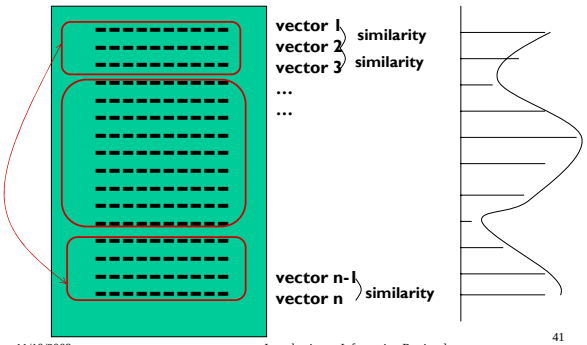
Similarity-based Clustering

- Define a similarity function to measure similarity between two objects
- Gradually group similar objects together in a bottom-up fashion
- Stop when some stopping criterion is met

Examples of Doc/Term Clustering

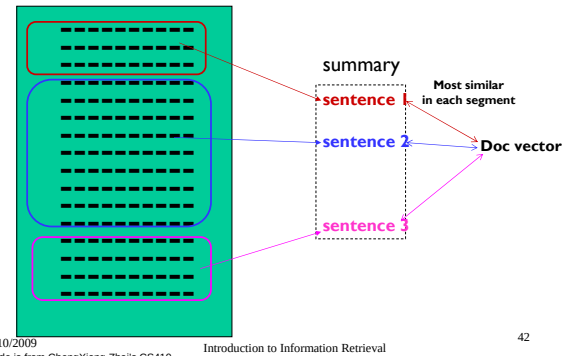
- **Clustering of retrieval results**
- **Clustering of documents in the whole collection**
- **Term clustering to define "concept" or "theme"**

3) Summarization - Simple Discourse Analysis



Slide is from ChengXiang Zhai's CS410

A Simple Summarization Method



Slide is from ChengXiang Zhai's CS410

Examples of Summarization

- News summary
- Summarize retrieval results
  - Single doc summary
  - Multi-doc summary

4) Filtering

- Content-based filtering (adaptive filtering)
- Collaborative filtering (recommender systems)

Examples of Information Filtering

- News filtering
- Email filtering
- Movie/book recommenders such as Amazon.com
- Literature recommenders

Content-based Filtering vs. Collaborative Filtering

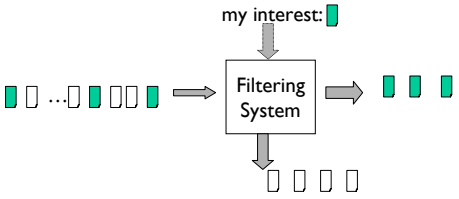
- Basic filtering question: Will user *U* like item *X*?
- Two different ways of answering it
  - Look at what U likes => characterize X => content-based filtering
  - Look at who likes X => characterize U => collaborative filtering
- Can be combined

Collaborative filtering is also called "Recommender Systems"

Slide is from ChengXiang Zhai's CS410

Adaptive Information Filtering

- Stable & long term interest
- System must make a delivery decision immediately as a document "arrives"



Slide is from ChengXiang Zhai's CS410

Collaborative Filtering

- Making filtering decisions for an individual user based on the judgments of other users
- Inferring individual's interest/preferences from that of other similar users
- General idea
  - Given a user *u*, find similar users $\{u_1, \ldots, u_m\}$
  - Predict *u*'s preferences based on the preferences of $u_1, \ldots, u_m$

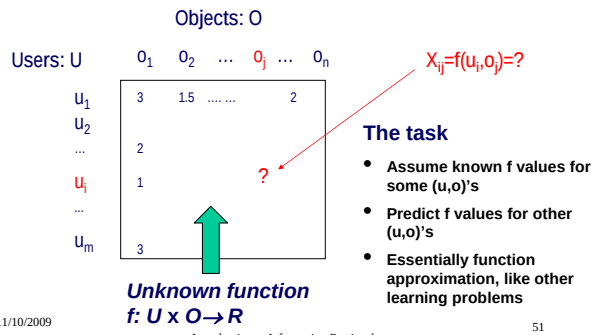
Collaborative Filtering: Assumptions

- Users with a common interest will have similar preferences
- Users with similar preferences probably share the same interest
- Examples
  - "interest is IR" => "favor SIGIR papers"
  - "favor SIGIR papers" => "interest is IR"
- Sufficiently large number of user preferences are available

Collaborative Filtering: Intuitions

- User similarity (user X and Y)
  - If X liked the movie, Y will like the movie
- Item similarity
  - Since 90% of those who liked Star Wars also liked Independence Day, and, you liked Star Wars
  - You may also like Independence Day

A Formal Framework for Rating



Unknown function
f: U x O→R

The task

- Assume known f values for some (u,o)'s
- Predict f values for other (u,o)'s
- Essentially function approximation, like other learning problems

Slide is from ChengXiang Zhai's CS410

News Recommendation on Facebook

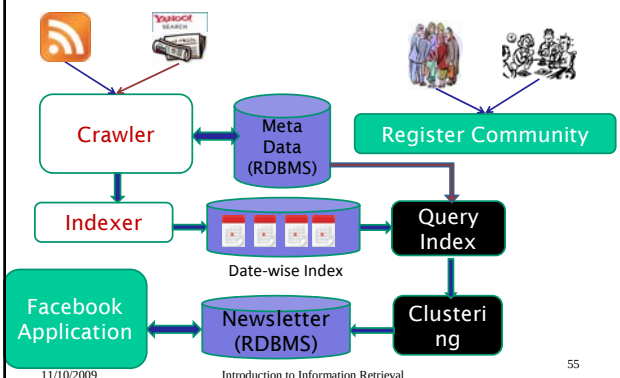
http://sifaka.cs.uiuc.edu/ir/proj/rec/

Motivation

Facebook as a medium for recommendations

- Provides a great platform with in built social networks.
- More than 120 million users log on to Facebook at least once each day.
- More than 95% of the users have used at least one application built on the Facebook Platform.
- Possible to make applications that deeply integrate into a user's Facebook experience.
  - FBML (Facebook Markup language)
  - FBJS (Facebook Javascript)
  - FQL (Facebook Query Language)
  - Facebook API

System Architecture



Application Main page

Collaborative User Feedback

- Three kinds of user feedback captured
  - Clickthroughs
  - Explicit Ratings
  - Inter-person recommendations
- They are linearly combined as follows:

$$F_{ij} = \sum_{k=1}^{3} \lambda_k * fk_{ij}$$

Where F_{ij} is aggregating all kinds of feedback for article a_j from user u_i

Demo

- News Recommender on Facebook

Application Information

- For more information about the application:
  - http://sifaka.cs.uiuc.edu/ir/proj/rec/
- http://apps.facebook.com/news_letters/

- We are looking for motivated students to work on this application.
- Requirements:
  - DataBase Knowledge
  - PHP
  - Perl
- Contact me if you are interested:
  - mkarimz2@illinois.edu

Thanks