INFSCI 2140
Information Storage and Retrieval
Lecture 6: Taking User into Account

Peter Brusilovsky
http://www2.sis.pitt.edu/~peterb/2140-051/

Ad-hoc IR in text-oriented DS

- The context (L1)
- Querying and matching (L2,L3)
- How to evaluate results (L4)
- How it all works internally (L5,L7)
- Better search and presentation taking user unto account: RF, QE, UM (L6)
- Better organization and visualization of search results (L10)

Overview

- Query Expansion and Relevance Feedback
- User modeling and adaptive information access
  - User profiles
  - Adaptive filtering
  - Adaptive search
  - Adaptive presentation in IR and information systems

Relevance Feedback and Query Expansion - 3 points of view

- Pragmatic view
  - Modern Information Retrieval, Baesa-Yates
- Designer's view
  - Bob Korfhage
- Interaction view
  - Nick Belkin

QE: Pragmatic point of view

- Query Expansion is a general technique of improving query to achieve better result (precision or recall)
- The idea is to “steer” the query closer to the vector subspace of the relevant documents
- How to steer? Vector operations:
  - Project (remove), Add, Re-weight

Query Expansion

- The idea: add some extra “good” terms to a query in a hope that it will bring more results or better precision
- Possible sources
  - Automatic: Properties of the document space and term distribution
    - Local analysis (current search)
    - Global analysis (whole space)
  - User-based: Relevance feedback

Automatic Query Expansion

- Local analysis: documents and term distribution in the current search
  - Local Clustering
  - Local Context Analysis
- Global analysis: document and term distribution in the whole space
  - Using Similarity Thesaurus
  - Using Statistical Thesaurus

QE with local clustering

- Idea: add terms that are similar to good terms in the context of good documents
- Step 1: Cluster all terms using similarity metrics based of co-occurrence in documents
- Step 2: For each term in a query add M nearest neighbors in the cluster to a query

QE with local context analysis

- Idea - add concepts “similar” to the whole query
- Step 1: Get N top ranked passages using original query
  - Document is divided into small chunks
- Step 2: Calculate similarity between each concept from the passage and the whole query using a version of TF*IDF
- Step 3: Add top concepts to the query

QE with global analysis

- Similar ideas based on global analysis of terms in document collection
- Use global similarity thesaurus (terms clustered as documents with inverse indexing)
- Represent a query in the space of concepts and find terms that are most close in this space to the whole query

Korfhage's view

- QE is a manipulation with query to improve search results
- Main source of information - user
- What can be changed
  - query, document, algorithms
- What kind of manipulation
  - re-weighting, adding/removing, altering
- User profiles and genetic algorithms

Belkin's view

- Information retrieval is an interaction between a human and information [system]
- Query is simply the first step in a dialogue - a part of user model that the system can build
- More interaction is required to update models of dialog partners

Relevance Feedback

- A IR system can learn something about the user preferences using the *relevance feedback*
- The user indicates the relevance of a set of documents and the system uses this feedback to modify its retrieval behaviors
- Then a new set of documents is presented and the retrieval process starts again

What kind of feedback?

- Positive feedback
  - Mark relevant documents
- Negative feedback
  - Mark irrelevant documents
- Mixed feedback
  - Positive and negative
  - Rating on some scale (cold/hot/lukewarm)

Relevance Feedback: The Idea

- A system can use positive relevance judgment trying to obtain more documents similar to those judged relevant
- A system can use negative relevance judgments trying to avoid documents similar to the one that were rejected

Relevance Feedback: Where?

- Where can we apply the information provided by the user?
- Query
- Profile
- Document representation
- Retrieval algorithm

Modifying the query

- This is what we can call user-based query expansion
- It is the simplest way
- It has no lasting impact on the system (that is a mixed blessing)
- Explored by Salton and Rocchio
  - Rocchio Algorithm

Modifying the user profile

- Profile - a long term representation of user inderests
  - We will learn details later
- These modifications last
- User profile and query often have the same or nearly the same representation
  - it is possible to use the same techniques
- The modifications should not be made on the basis of a single query

Modifying the document representation

- Modifications that last and can effect the behavior of the system for all the users
- It can be accepted if the community of users is a closed community of experts
- Methods are similar to query modification
  - Some variants of Rocchio algorithm can be used

Modifying the search algorithm

- It is something to do very carefully
- It is possible to change
  - Algorithms parameters (easy to undo)
  - The algorithm itself (this modifies deeply the behavior of the system

Rocchio Algorithm (IR)

$$Q' = Q + \frac{1}{n_i}\alpha \sum_{i=1}^{n_1} R_i - \frac{1}{n_2}\beta \sum_{i=1}^{n_2} S_i$$

where

Q is the vector of the initial query

R_i is the vector for relevant document

S_i is the vector for the irrelevant documents

α, β are Rocchio's weights

Relevance feedback: "space" view

- D_r - set of relevant documents $\{d_r\}$
- D_n - set of non-relevant documents $\{d_n\}$
- Rocchio's Formula:

$q_m = \alpha\, q + (\beta / |D_r|) \Sigma\, d_r - (\gamma / |D_n|) \Sigma\, d_n$

- Ide's Regular Formula

$q_m = \alpha\, q + \beta \Sigma\, d_r - \gamma \Sigma\, d_n$

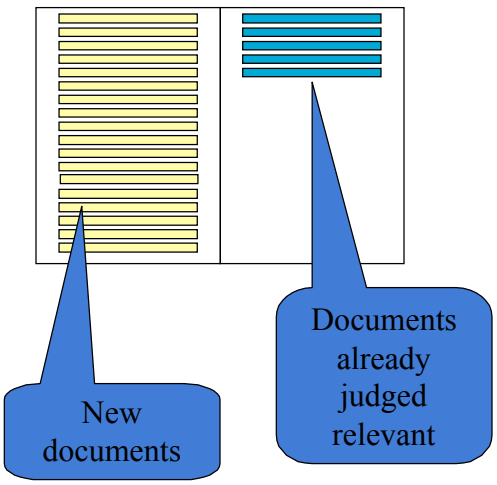
- Ide's Dec Hi Formula

$q_m = \alpha\, q + \beta \Sigma\, d_r - \gamma \Sigma \max_{\text{non-rel}} (d_j)$

Problems - User Side

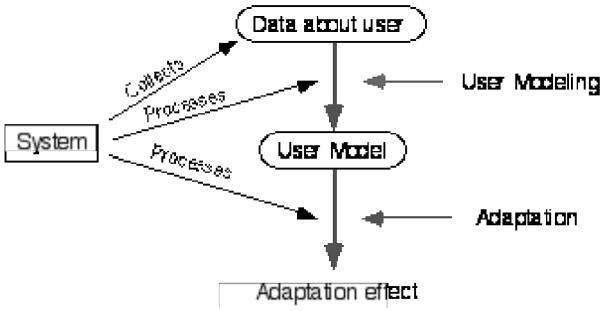
- Rating
  - More information vs. user overload
- Supporting iterative search
  - The user gets tired after 3 or 4 iterations
  - The user prefer to have a sort of "incremental interface" with the new documents highlighted in order to avoid to scan the same documents again and again

Interface for RF Search



- What can help?
- Better interface
- The use of long-term profiles

User models in adaptive systems



Classic loop user modeling - adaptation in adaptive systems

SDI: The origin of profiles

- Selective Dissemination of Information
  - User defines her profile of interests
  - System filters all relevant new sources
  - Profile - more than a query - long term interests - that is where the work on user profiles started
- Used for retrospective and awareness
- Profiles kept updated by the users

Information Filtering

- Common meaning
  - Modern version of SDI also known as Awareness Systems
  - These systems are designed to keep the user informed about an area of interest
  - The user submits his profile as a permanent query that periodically is matched by the system to the new information
- Bob Korfhage's view:
  - "Mining rich ore"

User profile

- Common term for user models in IR/IF
- A user's profile is a collection of information about the user of the system.
- These information is used to get the user to more relevant information
- Views on user profiles
  - Korfhage - another reference point
  - Belkin - starting part of the user model

Simple vs. Extended User Profile

- Simple profile
  - A set of search terms (0-1 vector)
  - A boolean query
  - A set of terms with their weights (vector)
- Extended profile
  - contains information related to the user as a person in order to understand or model the use that a person will make with the information retrieved

Extended profile

- Knowledge: about the system and the subject
- Goals: local and global
- Interests
- Background: profession, language, prospect, capabilities
- Preferences (types of docs, authors, sources…)

Who maintains the profile?

- Profile is provided and maintained by the user/administrator
  - Sometimes the only choice
- The system constructs and updates the profile (automatic personalization)
- Collaborative - user and system
  - User creates, system maintains
  - User can influence and edit

General system types

- Search-oriented Web IR systems
  - Ad-hoc IR
  - Information Filtering
- Browsing-oriented IR systems
  - Hypertext and the Web
- Information Visualization
- Personalized information spaces
  - Bookmarking systems, MyLibrary systems
- Search services
  - Cover several functions around single user model

What Can Be Adapted?

- Adaptive search and filtering
- Adaptive presentation
  - Presenting a page (analyze results)
- Adaptive information visualization
- Adaptive navigation support
  - Presenting search results (analyze results)
  - Presenting links on a page (proceed from)
- Adaptive collection (crawling)

Adaptive Information Filtering

- Goals:
  - Improve the long-term user profile to get better filtering results
- Methods
  - Variations of relevance feedback for improving the profile
  - Machine learning approaches to learn users' "true" long-tem interests

Example: Adaptive News Server

- Adapts to long-term user preferences
- What to consider:
  - Which news items the user browses
  - How many pages in a new item the user read (mobile platform)
- Uses machine learning
- Significantly improved the number of messages read. Startup launched

Adaptive Search

- Goals:
  - Find documents (pages) that are most suitable for the individual user
- Methods:
  - Employ user profiles representing long-term interests (Korfhage)
  - Use heuristics for adaptation to user types and actions

User profile for adaptive search

- The profile is used to give a context to the query, in order to reduce ambiguity.
- For example the background of the user can be helpful to understand what kind of information he is looking for. A query about the theory of groups has a different meaning for a mathematician and a sociologist. Moreover a student in math is interested in the basic concepts, while a an expert is interested in advanced materials

Using user's profile for search

- The user profile can be applied in three ways:
- To modify the query itself (pre-filter)
- To process results of a query (post-filter),
- To change the usual way of retrieval
  - Profile is treated as a *reference point*

Post-filter

- The user profile is used to organize the results of the retrieval process
  - present to the user the most interesting documents
  - Filter out irrelevant documents
- Extended profile can be used effectively
- In this case the use of the profile adds an extra step to processing
- Similar to classic information filtering problem
- Typical way for adaptive Web IR

Pre-filter

- In this case the profile is used to modify the query.
- Imagine that:
  - the documents,
  - the query
  - the user profile

  are represented by the same set of weighted index terms

Pre-filter: Linear Transformation

- The query $q=q_1, q_2, \ldots q_n$
- The profile $p=p_1, p_2, \ldots p_n$

- The query modified by the user profile will be something like that:

$$modified\ q_i=Kp_i+(1-K)q_i \qquad i=1,2,\ldots n$$

Pre-filter: Linear Transformation

$$modified\ q_i=Kp_i+(1-K)q_i$$

- In this case we add the terms of the profile to the query ones, weighted by K

for $K=0$ $modified\ q_i=q_i$ the query is unmodified

for $K=1$ $modified\ q_i=p_i$ the query is substituted by the profile

Piecewise linear transformation

- if the term appears in the query **and** in the profile then the linear transformation is applied
- if the term appears in the query but not in the profile is left unmodified or diminished slightly
- if the term appears in the profile but not in the query it is not introduced, or introduced with a weight lower than in the profile.

Separate reference points

- In this case documents are retrieved if they are “near” the query or the profile.
- In the following discussion we assume that the similarity is measured by distance

  $\|D, Q\|$

  where D is the document and Q is the query

Separate reference points

- We have different way to integrate query and profile as separate reference points:
  - Disjunctive model of query-profile integration
  - Conjunctive model of query-profile integration
  - Ellipsoidal model
  - Cassini oval model

Disjunctive model

- In this case we will take the document if the following condition is satisfied:

$$\min\left(\|D,Q\|,\|D,P\|\right) < d$$

The D document should be “near” the query Q or the profile P

Conjunctive model

- Condition to satisfy

$$\max\left(\|D,Q\|,\|D,P\|\right) < d$$

The D document should be "near" the query Q and the profile P

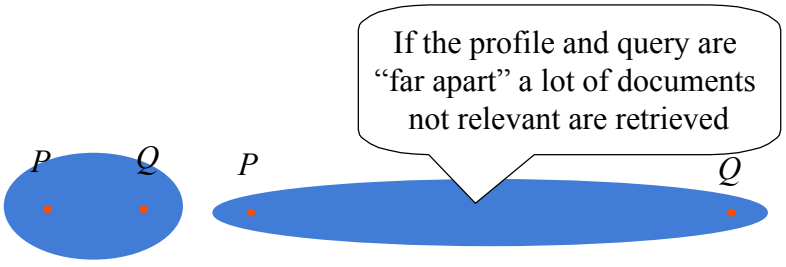
- In this case if profile and query have little, or noting, in common very few documents are retrieved

Ellipsoidal model

- Condition to satisfy

$$\|D,Q\| + \|D,P\| < d$$

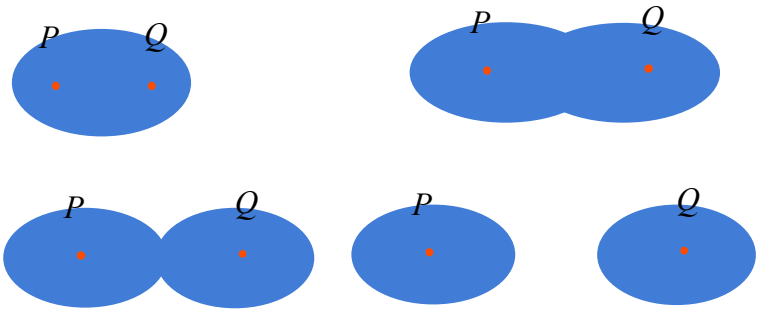
this is the equation of a ellipse.



Cassini model

- Condition to satisfy

$$\|D,Q\| \times \|D,P\| < d$$



Types of systems

- Pre-filter with QE - Mobasher's
- Adaptive search: SmartGuide
- Post-filters: Syskill & Webert, WIFS
- Search-based Recommenders: a combination of IR and IF
- Adaptive Meta-Search engines
  - Adaptive selection and ranking of sources

Example: SmartGuide

- Access to the CIS-like information
- User has a long-term interests profile and current queries
- Information is searched using a combination of both
- Profile is initiated from a stereotype and kept updated
- Increased user satisfaction, decreased navigation overhead

Example: WIFS

- Adaptive post-filter to AltaVista search engine
- Maintains an advanced stereotype-based user model (Humos subsystem)
- User model is updated by watching the user
- The model is used to filter and re-order the links returned by AltaVista

Adaptive Presentation

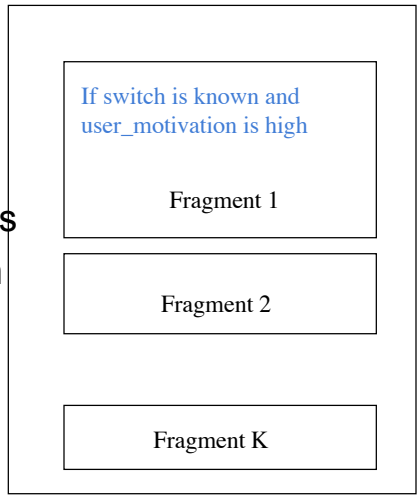
- Provide the different content for users with different knowledge, goals, background
- Select/stress most relevant content for the user
- Remove/fade irrelevant pieces of content
- Show additional relevant material for some categories of users
  - comparisons
  - extra explanations
  - details
- Sort fragments - most relevant first

Adaptive presentation techniques

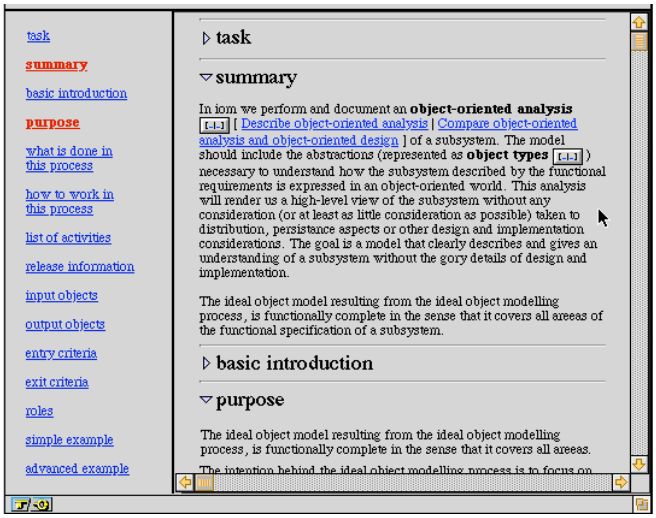
- Conditional text filtering and stretchtext
  - ITEM/IP, PT, AHA!, MetaDoc, KN-AHS, PUSH, ADAPTS
- Frame-based adaptation
  - Hypadapter, EPIAIM, ARIANNA, SETA
- Full natural language generation
  - ILEX, PEBA-II, Ecran Total
- Most of techniques rely on extended profiles

Conditional text filtering

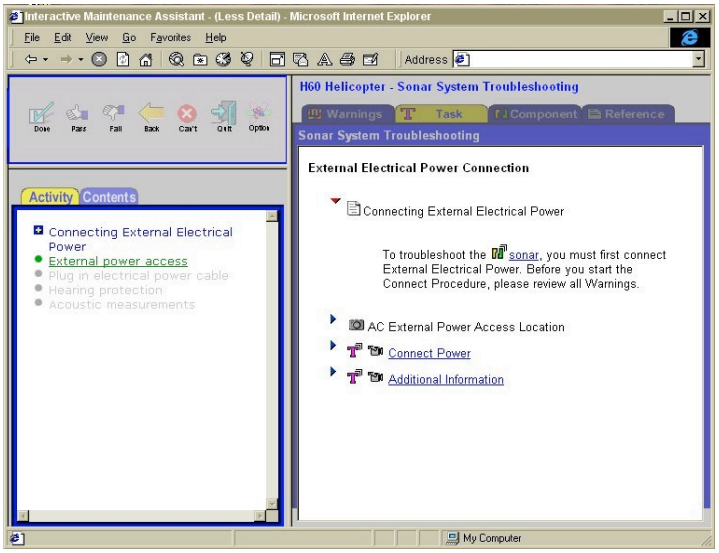
- Similar to UNIX cpp
- Universal technology
  - Altering fragments
  - Extra explanation
  - Extra details
  - Comparisons
- Low level technology
  - Text programming



Example: Stretchtext (PUSH)



Example: Stretchtext (ADAPTS)



Adaptive presentation: evaluation

- MetaDoc: On-line documentation system, adapting to user knowledge on the subject
- Reading comprehension time decreased
- Understanding increased for novices
- No effect for navigation time, number of nodes visited, number of operations