Where we are?

- The context (L1)
- Querying and matching (L2,L3)
- How it all works internally (L4,L10)
- How to evaluate results (L5)
- Better organization and visualization of search results (L7)
- Improving search results: UI, RF, QE (L8); UM (L9)
Overview

- **User modeling**
  - User profiles
- **Adaptive information access**
  - Adaptive filtering
  - Adaptive search
  - Adaptive presentation in IR and information systems
  - Adaptive navigation support in HT and WWW

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 - while looks like a query - is really *more* than a query since it represents long term interests
  - that is where the work on user profiling 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.
- This information is used to get the user to more relevant information
- Views on user profiles in IR community
  - Classic (Korfhage) - a reference point
  - Modern (Belkin) - starting part of the user model

Core vs. Extended User Profile

- Core profile
  - contains information related to the user search goals and interests
- 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
Simple (vector) Core Profiles

- Primitive profile (any model)
  - A set of search terms (0-1 vector)
- For Boolean model of IR
  - A Boolean query
- For vector model of IR (dominated)
  - A set of terms with their weights (vector)
  - An overlay (set of weights) over a simple domain model that is just a list of terms that could be of interest to the users

Advanced Core Profiles

- Domain model is a network of terms
  - A subset of the model (simple overlay)
  - A weighted overlay over DM
- Domain model is a hierarchy of topics
  - typically, each topic is a vector of terms
  - A subset of the model (simple overlay)
  - A weighted overlay over DM
Group Profiles

- A system can maintain a group profile in parallel or instead of user profile
- Could resolve the privacy issue (navigation with group profile)
- Could be use for new group members at the beginning
- Could be used in addition to the user profile to add group “wisdom”

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
  - Does it help or not?

General System Types

- Search-oriented systems
  - Classic search and focused crawling
- Classic Information Filtering
- Recommender systems
  - Content-based, collaborative, hybrid,…
- Browsing-oriented 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 navigation support
  - Presenting search results (analyze results)
  - Presenting links on a page (proceed from)
- Adaptive information visualization
- 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-term 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
Using Profiles for Search: Ideas

- The profile is used to give a context to the query in order to reduce ambiguity.
- The profile of interests will allow to distinguish what the user asking about "Pirates" really wants.
- 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 an expert is interested in advanced materials.

Using Profiles for Search: Ways

- The user profile can be applied in four 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
- Special case: meta-adaptation for meta-search
Examples of Systems

- Pre-filter with QE
  - Koutrika, Mobasher
- Pre-filter with RF
  - SmartGuide
- Post-filters
  - Syskill & Webert, WIFS
- Adaptive Meta-Search engines
  - Adaptive selection and ranking of sources

Pre-Filter: Query Expansion

- User profile is applied to add terms to the query (works with any IR model)
  - Similar terms could be added to resolve indexer-user mismatch
  - Related terms could be added to resolve ambiguity
- Requires an advanced profile
Example: Koutrika & Ioannidis’05

- Advanced relevance network for query expansion
- `java -> java and programming -> java and (programming or development)`

Pre-Filter: Relevance Feedback

- In this case the profile is used to “move” the query vector (vector IR only)
- 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:
\[
modified \ q_i = Kp_i + (1-K)q_i \quad i=1,2,\ldots,n
\]

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.

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

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

Post-Filter: Re-Ranking

- Re-ranking is a typical approach for post-filtering
- Each document is rated according to its relevance (similarity) to the user or group profile
- This rating is fused with the relevance rating returned by the search engine
- The results are ranked by fused rating
  - User model: WIFS, group model: I-Spy
Example: WIFS (Micarelli)

- 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

Post-Filter: Annotations

- The result could be relevant to the user in several aspects. Fusing this relevance with query relevance is error prone and leads to a loss of data
- Results are ranked by the query relevance, but annotated with visual cues reflecting other kinds of relevance
  - User interests - Syskill and Webert, group interests - KnowledgeSea
Example: Syskill and Webert

- First example of annotation
- Post-filter to Lycos
- Hot, cold, lukewarm

Example: Knowledge Sea II
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.

  where \( D \) is the document and \( Q \) is the query

\[ \|D, Q\| \]

Adaptive Presentation: Goals

- 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: Methods

- 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

If switch is known and user_motivation is high

Fragment 1

Fragment 2

Fragment K
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

Adaptive Recommendation

- Goals:
  - Actively recommend relevant items from the system’s document space to users
  - Inverse paradigm - push vs. pull: the user is passive (not really querying), the system is active - recommending
- Methods:
  - Content-based, collaborative, feature-based, demographic, hybrid
Implicit Query - Recommenders

- A user identifies 1 or several objects as being of interest
- The recommender system suggests “similar” objects from the DS
- Shares a lot in common with IF
  - Mechanism is similar to IF
  - User interface and overall paradigm is similar to browsing support (watch user’s browsing and other actions)

Parameters for recommenders

- Short term of long term “interest”
  - 1-2 “interesting” items in one session
  - many items over time
- Direct or indirect “rating”
  - Direct rating
  - “Looking at”: click, browse
  - Other ways to show interest:
    -
    -
- Type of the filtering engine
Types of “filtering”

- Content based:
  - find items similar to the set of interesting by content
  - Bag of words or features

- Collaborative:
  - Find users who have similar opinion with you
  - What else these user consider interesting?

- Demographic and rule-based

- Hybrid

Case Study: Paper Recommender

- Suggest publication for several reasons
  - Similar to just downloaded paper
  - Cites just downloaded paper
  - Cited by just downloaded paper
  - Most frequently downloaded together
  - Added since last visit and cited by earlier downloaded paper
  - Most popular but not yet considered

- Doubled # downloads per visit!
Adaptation in Paper Recommender

- **What?**
  - Learns relative importance of the reasons by watching agreements to suggestions
- **When?**
  - Clicked "suggest", downloaded paper, idle time
- **Adaptability (user can specify)**
  - Which reasons not to consider
  - Topics of interests
- **URL:**
  - http://www.ics.uci.edu/~pazzani/Publications/Publications.html

Case Study: Paper Recommender

- How the users shows their interests?
- Long term or short term interests?
- Which technology is better - content-based or collaborative filtering?
Case Study: XLibris

- The user reads the text and annotates it using a pen

- XLibris can generate *marginal links and further reading list*

- How the users shows their interests?

- Long term or short term interests?

- Which technology is better - content-based or collaborative filtering?
Case Study: WATSON

- WATSON system (Northwestern U)
  - The user types or work in GUI
  - The system observes his/her work
  - The recommendation window shows relevant resources
- Instant Queries

Case Study: WATSON

- How the users shows their interests?
- Long term or short term interests?
- Which technology is better - content-based or collaborative filtering?
Case Study: MovieCentral

- The users rate movies
- The system can suggest best bets
- Users keep rating movies while checking best bets

http://www.moviecentral.com
Alternative URL: http://www.movielens.umn.edu/
Case Study: Amazon.com

■ How the users shows their interests?
■ Long term or short term interests?
■ Which technology is better - content-based or collaborative filtering?

Adaptive hypermedia: Why?

☑ Different people are different
☑ Individuals are different at different times
☑ "Lost in hyperspace"
   ➔ Large variety of users
   ➔ Variable characteristics of the users
   ➔ Large hyperspace
Where it can be useful?

- Web-based education
  - ELM-ART, AHA!, KBS-Hyperbook, MANIC
- On-line information systems
  - PEBA-II, AHA!, AVANTI, SWAN, ELFI, ADAPTS
- E-commerce
  - Tellim, SETA, Adaptive Catalogs
- Virtual and real museums
  - ILEX, HYPERAUDIO, HIPS, Power, Marble Museum
- Information retrieval, filtering, recommendation
  - SmartGuide, Syskill & Webert, IfWeb, SiteIF, FAB, AIS

What can be adapted?

- Hypermedia = Pages + Links
- Adaptive presentation
  - content adaptation
- Adaptive navigation support
  - link adaptation
Adaptive Hypermedia Technologies

Adaptive Presentation
- Adaptive multimedia presentation
- Natural language adaptation
- Inserting/removing fragments
- Altering fragments
- Sorting fragments
- Dimming fragments

Adaptive Text Presentation
- Canned text adaptation
- Stretch
text

Adaptation of Modality
- Direct guidance
- Hiding
- Disabling
- Removal

Adaptive Link Sorting
- Sorting fragments

Adaptive Link Finding
- Dimming fragments

Adaptive Link Annotation
- Map adaptation

Adaptive Link Generation

Adaptive Navigation Support: Goals

- Guidance: Where I can go?
  - Local guidance (“next best”)
  - Global guidance (“ultimate goal”)

- Orientation: Where am I?
  - Local orientation support (local area)
  - Global orientation support (whole hyperspace)
Adaptive Navigation Support: Methods

- Direct guidance (WebWatcher)
- Restricting access
  - Removing, disabling, hiding
- Sorting
- Annotation
- Generation
  - Similarity-based, interest-based
- Map adaptation techniques

Example: Adaptive annotation

Annotations for topic states in Manuel Excell: not seen (white lens); partially seen (grey lens); and completed (black lens)
Example: Adaptive annotation

1. Concept role
2. Concept state
3. Current section state
4. Linked sections state

What can be adapted: links

- Contextual links (“real hypertext”)
- Local non-contextual links
- Index pages
- Table of contents
- Links on local map
- Links on global map
Link types and technologies

<table>
<thead>
<tr>
<th></th>
<th>Direct guidance</th>
<th>Sorting</th>
<th>Hiding</th>
<th>Annotation</th>
<th>Map adaptation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual links</td>
<td>OK</td>
<td></td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Non-contextual links</td>
<td>OK</td>
<td>OK</td>
<td>?</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Table of contents</td>
<td>OK</td>
<td></td>
<td>?</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>OK</td>
<td></td>
<td></td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Local map</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td></td>
</tr>
<tr>
<td>Global map</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

Adaptive navigation support: evaluation

- Sorting
  - HYPERFLEX, 1993
- Annotation (colors) and hiding
  - ISIS-Tutor, 1995
- Annotation (icons)
  - InterBook, 1997
- Hiding
  - De Bra’s course, 1997
Evaluation of sorting

- HYPERFLEX: IR System
  - adaptation to user search goal
  - adaptation to “personal cognitive map”
- Number of visited nodes decreased (significant)
- Correctness increased (not significant)
- Goal adaptation is more effective
- No significant difference for time/topic

Annotation and hiding: ISIS-Tutor

- An adaptive tutorial for CDS/ISIS/M users
- Domain knowledge: concepts and constructs
- Hyperspace of learning material:
  - Description of concepts and constructs
  - Examples and problems indexed with concepts (could be used in an exploratory environment)
- Link annotation with colors and marks
- Removing links to “not relevant” pages
Sample index page (annotation)

Sample index page (hiding)
InterBook: concept-indexed ET

- “Knowledge behind pages”
- Structured electronic textbook (a tree of “sections”)
- Sections indexed by domain concepts
  - Outcome concepts
  - Background concepts
- Concepts are externalized as glossary entries
- Shows educational status of concepts and pages

Book view
Adaptive annotation can:

- Reduce navigation efforts
  - Results are not significant (variety of styles?)
- Reduce repetitive visits to learning pages
  - Significant - if applied properly
- Encourage non-sequential navigation
- Increase learning outcome
  - For those who is ready to follow and advice
- Make system more attractive for students

Web Recommenders vs. ANS

- Recommenders originate from filtering systems and use an old “search” approach, ANS originates from hypertext and focuses on navigation support
- Even advanced recommenders use simple 1-D “list of links” presentation, ANS use 1.5-D presentation
  - Power of a recommendation engine could be enhanced by power of a proper interface
- Modern AH systems require content knowledge, modern recommender systems can create/extract it.
A broader picture - adaptive IS

- Users work with an adaptive IS, the system attempts to develop a user model (knowledge, interests, goals) and assist each user adaptively
- There are different ways to model the user (knowledge, interests, collaborative, content-based…)
- There are different ways to use the knowledge to assist the user

Case Study: Knowledge Sea

- What the users are doing
- What kind of information the system is able to extract from watching the users and how
- How the system assists the users in the process of information access?