Information Retrieval: Assignment 1: By Poonam Bhide

Document filtering is an application that stores a large number of queries or user profiles and compares these profiles to every incoming document on a feed. Documents that are sufficiently similar to the profile are forwarded to that person via email or some other mechanism.

Question 1: *Describe the components of a filtering engine using a block diagram of the architecture, a flowchart of the filtering process, and text explaining the function of the components. Use the same level of detail that we gave in the second lectures. For instance, don't just say that the filter needs "text acquisition" but that it needs "format conversion" and "stemming", to name only one example.*

Answer 1: There is huge corpus of documents, which are available in various formats. Texts, audio, email etc. The idea of document filters is to provide a user with the relevant information by filtering the incoming documents as per user's interests. [2]

The "interestingness" of a web page can be defined as the relevance of the page with respect to the user's long-term information goals. Document collections change with time, but information needs are static. For document filtering, long-term information is represented as a user profile

Document / Information filtering offers tool for discriminating the relevant and irrelevant information to users by providing personal assistance. Relevant information can be determined by 'interest of the user'. . Feedback on the interestingness of a set of previously visited sites, demographic location, browsing history, and location can be used to learn a profile that would predict the interestingness of unseen sites. [1,2,3,4,5]

Following are important terminologies and processes involved in document filtering:

User profiles (Profiles):

- Represent long-term information needs.
- Can be stored either as Boolean queries, relevant and non-relevant documents or relational constraints.
- This representation depends on the type of filtering model i.e. static filtering (fixed profiles) or adaptive filtering model (profiles change).

Assumptions[2]:

- Document filtering application may have thousands or even millions of profiles
- Many new documents will enter the filtering system daily
- Most profiles are represented as text or a set of features

Filtering process [2]:

- Build an inverted index for the profiles by creation, transformation.
- Distill incoming documents as "queries" and run against profile index
- Periodically feedback is taken by users in the form of the pages that were of their

interest.

- Positive and Negative both feedbacks are given and profiles of the users are updated.
- Indexes for profiles are modified.
- Following are the steps of this process in the form of flowchart

STEPS FOR FILTERING

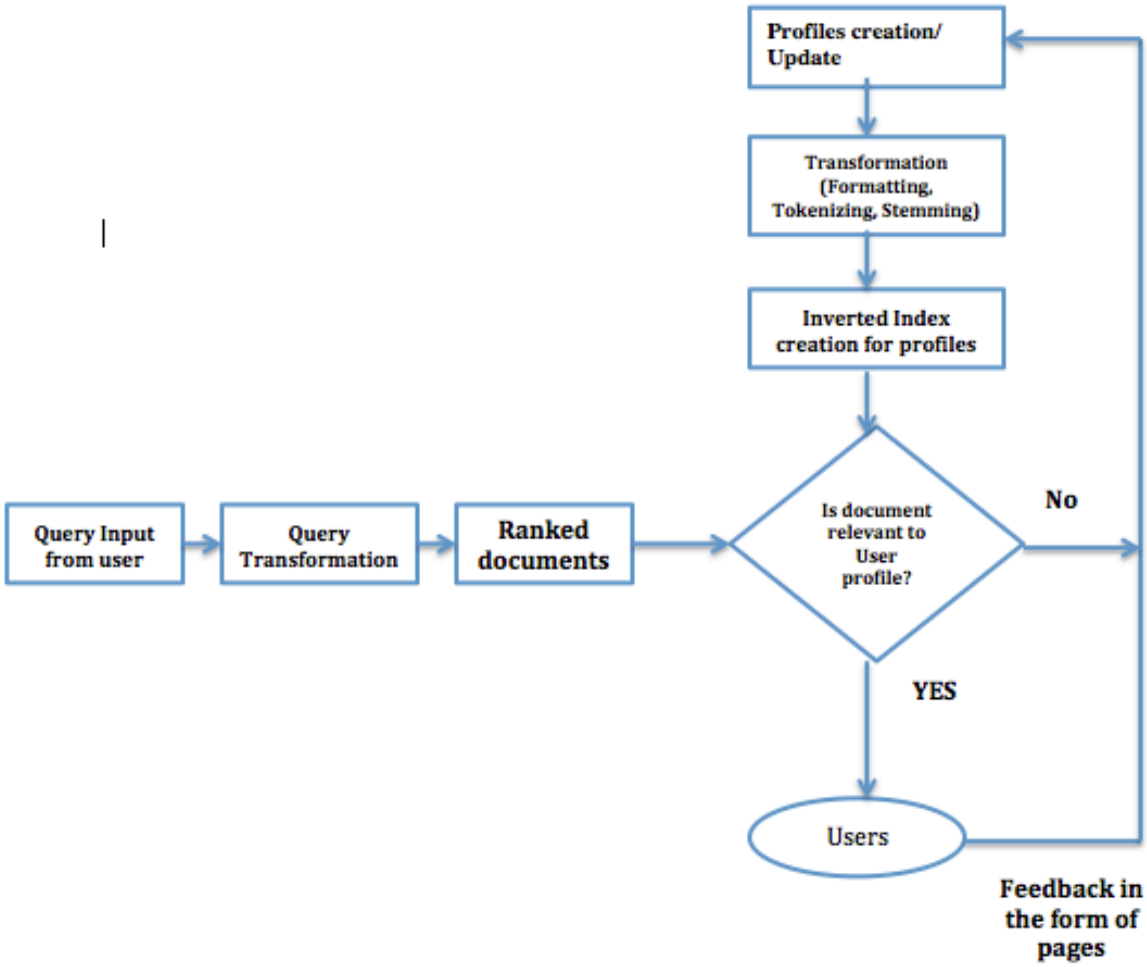

Figure 1: Steps of filtering

Block Diagram for Filtering Application (Figure 2 shown below)

Following are the main components of filtering application:

1. **User Profile Base**
   User Profile Processor initially creates the user profiles. User profiles can be either content based or rules based. These profiles do not have any fixed format or structure. This is a repository of User Profiles.

2. User Profile Processor

This user profile processor will be responsible for profile creation or any modification. It will have many components to generate or update the user profiles.

a. User created profiles: This will contain information, which is explicitly entered by the users.

b. System created profiles: A set of data items which have already been judged by the user as relevant, are analyzed by software (using stemming algorithms), in order to identify the most frequent and meaningful terms in the text. Those terms, weighted according to the frequency of their appearance, constitute the user profile. Profiles are also generated based on the user stereotype which would contain the demographic location, browsing history etc.

Feedback from users in the form of pages goes to User Profile Processor that applies some algorithm like Bayesian classifiers to determine probability of the possible terms in the user profiles. POS Tagging can also be used to determine the linguistic patterns.

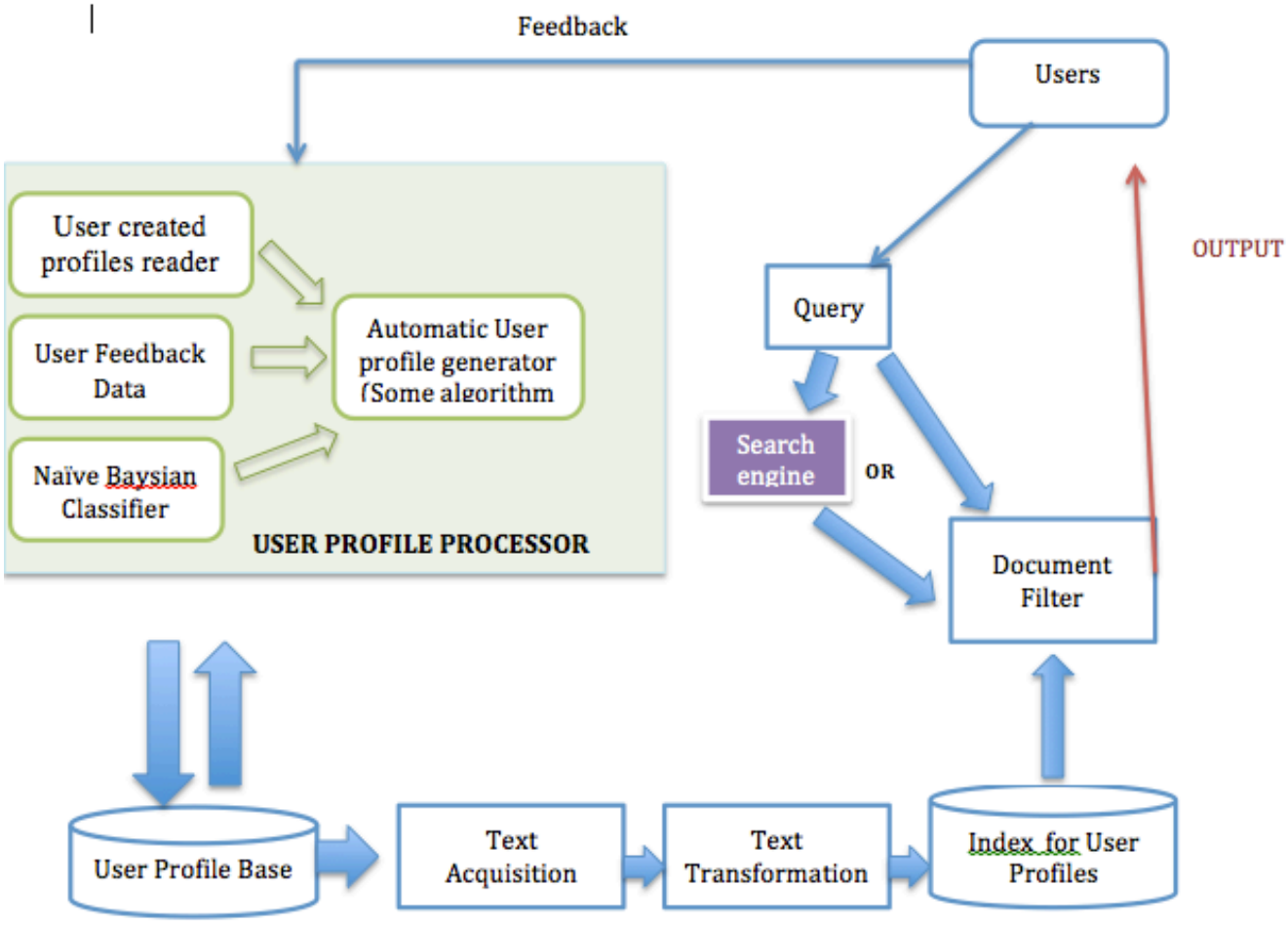

Figure 2: Block Diagram for Filtering Application

3. **Text Acquisition (User Profiles)**
   The data that is stored in "User Profile Base" is taken by Text Acquisition module. Following are the functionalities performed by it :
   a. Formatting / Conversion: This module converts the text into consistent formats like Feeds.
   b. The feeds are usually in the form of XMLs. XMLs are lightweight and good for faster access

4. **Text Transformation (User Profiles)**
   This would first make tokens of terms in user profiles.
   Stemming will remove the unwanted words and also use the root form of word.

5. **Inverted Index User Profile**
   This will generate inverted indexes for the User profiles and store them in Document filter. Inverted index are calculated based upon the term frequency. Terms are represented as vectors. Hence for every user profile there will be a corresponding vector of terms depicting the interest of that user.

6. **Document Filter**
   This filter will take following 2 inputs:
   - Query results produced by the typical IR system
     OR directly the queries by users.
   - User Profile Index

   The queries are run in conjunction with the user profile index.

   The "Search Engine" in purple box in figure 2 (highlighted in solid) will have all the blocks for index creation and query processing. Black box will have all the steps like Text Acquisition (Content from feeds, formatting into XMLS like RSS feeds).,Text Transformation (Tokenizing, Stemming, POS-Tagging) and creation of inverted index which is used for query processing. Since here the focus is on document filtering i.e. User Profiles the relevant details are described detail.[1]

7. **Users**
   The successful query (OUTPUT in figure) results are provided to the users. The results could be in any form like web page, documents, emails etc. This over all process would produce more relevant results to users. The precision would definitely improve if such technique is followed in conjunction with typical IR systems.

Question 2: *Explain the major differences compared to a search engine. Consider issues such as specific efficiency problems and the usefulness of ranking in a filtering application.*

Answer 2:

In information retrieval systems the search engines mainly take the query and give relevant documents based upon the query's relevance. There are many ranking algorithms to achieve maximum precision in the query results. Goal of search engine is to obtain information from the base, which helps the person in problem management. Document filtering or Information filtering is mainly associated with the Users. [3]

In Information filtering, the user's needs are expressed as User profiles. A profile represents user's long-term needs. Whenever any query is given the results are filtered and matched with the user profiles. Hence Document filtering does intelligent job by analyzing what a user would want from the profiles and then produce the results. Where a person with a one-time goal and one-time query typically concerns IR with single uses of the system, a person or persons with long-term goals or interests concern information filtering with repeated uses of the system. [3]

Text-related issues. For information filtering, the timeliness of a text is often of overriding significance. For IR, this has typically not been the case. [3] The search engines are more efficient than the Information filtering systems as the amount of data and processing for filtering takes more time than search engines. Also search engine might fetch some page, which have very high ranking, but document filter may not approve it. But if some intelligent algorithm like Bayesian Classifier passes the document because of its higher probability the user may not miss any relevant document.

I believe the ranking will be useful in Document filtering applications because ranking will give relevant query results and from those query results if user profile index are used the result of the query would definitely have higher precision than just ranking. As user's profile and interests would be taken into consideration along with query results, the chances of getting more accurate results will be more. Hence in the block diagram I have given output of IR systems to document filter. The performance due to this approach of ranking in conjunction with filtering might be lowered but the by distributing the load and using the performance tuning techniques the performance can be improved. The method of using both the approaches together will definitely help in improving the precision.

References:

[1] Slides by Prof. David Smith
[2] http://www.search-engines-book.com/slides/
[3] http://ftp.cse.buffalo.edu/users/azhang/disc/disc01/cd1/out/papers/sigir/p313-kuflik.pdf
[4] http://www.cs.usc.edu/assets/003/83240.pdf
[5] http://www-nlpir.nist.gov/related_projects/tipster/gen_ir.htm
[6] https://www.ischool.utexas.edu/~i385d/readings/Belkin_Information_92.pdf
[7] http://download.springer.com/static/pdf/845/art%253A10.1023%252FA%253A1007369909943.pdf?auth66=1379740506_21da32b3426df92f62d2266882214f60&ext=.pdf