An Automated System for Web Portal Personalization

Charu C. Aggarwal and Philip S. Yu

IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
{ charu, psyu }@us.ibm.com

Abstract

This paper proposes a system for personalization of web portals. A specific implementation is discussed in reference to a web portal containing a news feed service. Techniques are proposed for effective categorization, management, and personalization of news feeds obtained from a live news wire service. The process consists of two steps: first manual input is required to build the domain knowledge which could be site-specific; then the automated component uses this domain knowledge in order to perform the personalization, categorization and presentation. Effective schemes for advertising are proposed, where the targeting is done using both the information about the user and the content of the web page on which the advertising icon appears. Automated techniques for identifying sudden variations in news patterns are described; these may be used for supporting news-alerts. A description of a version of this software for our customer web site is provided.

1 Introduction

In recent years, the growth of the world wide web has lead to the proliferation of a large number of electronic commerce sites which require the use of personalization systems. Such systems include *recommender systems* and *information filtering techniques* [1, 9, 10, 11, 18, 23]. In addition, electronic commerce sites and portals may have needs for applications in which the user-behavior at the site is used in order to make recommendations about advertisements or other products. Such personalized recommendations comprise effective target marketing techniques, and increases the potential likelihood of the recommended item to be of interest to a given customer. A detailed survey and references on such techniques may be found [16, 17, 19].

One example of an environment in which personalization can be offered that of news wire services via live online text news feeds at a web site. Often, particular sites may be interested in news of a particular nature, topic, and organization which may be different from the natural topical schemes used by the news providers. A subscription based news service may also wish to present users with news items determined to be of greatest interest to them. Furthermore, electronic systems for news dissemination change the nature of the advertising which can be performed at that site, since the advertisements can be placed dynamically on web pages depending upon the content of the page and the user who is accessing the news article. This kind of targeted advertising greatly enhances its value from the point of view of merchants. For such cases, effective methods for news filtering, presentation, recommendation, and advertising are required using personalization software at the web site itself. The features provided by a given personalization system are critically dependent upon its architecture and the goals with respect to which it was created. Much of the recent research has touched on various algorithmic and architectural aspects [2, 3, 4, 8, 10, 11, 18, 19, 20] of personalization systems; an integrated view of the system architecture, recommendation algorithms and interfaces is valuable from the point of view of both researchers and implementors.

In this paper, we discuss a personalization system for management of news. We discuss an overview of the various features and also highlight the different choices which may be made while offering these features to the different users. We integrate several personalization and text mining algorithms into this system and also discuss some novel features which have not been discussed elsewhere in the literature. We illustrate our system using a personalized news feed system built for a company specific web site.

**Proceedings of the 28th VLDB Conference,
Hong Kong, China, 2002**

The features which this system can provide are the following:

- Methods for news filtering: News which are not relevant to the given site are automatically filtered out.

- Methods for news organization: News may be organized based on particular categories which are specific to that particular site.

- Personalized news presentation: The system can provide news which is personalized and tailored to individual people. Thus, for example, if a person has been tracked to be a fan of sports related pages, then news corresponding to this topic is presented to him.

- Personalized advertisements: We discuss techniques for providing personalized advertisements. Advertisements are placed on web pages depending upon both the content of the web page and the user viewing it. This creates the possibility of targeted advertising on the internet.

- Methods for news alerts: The system supports techniques for supporting news alerts: sudden bursts in news on a given topic; something which we detect by mining sudden changes in trends.

- Effective search capability: The system provides semantic keyword search capability; it is possible to search for news items related to to a given keyword, but not necessarily containing it.

This paper discusses a personalization software which is tailored to portals containing news feeds. However, the techniques and architecture discussed can be utilized for applications such as development of personalized portals. Such applications are still in their infancy, but have great demand because of the commercial advantages from the target marketing opportunities thus created. Often portals provide the primary gateway which controls the user accesses to different web pages and sites. It is clear that the potential from personalizing this access in order to improve the click-through rates of web sites has advantages to both merchants and consumers. We discuss the application of our system for such cases.

This paper is organized as follows. In the next section, we discuss the system architecture and engines which are needed for this personalization software. We provide details of the text mining techniques which may be used in order to support the different features which have been enumerated above. Section 3 provides details of the actual interface which is used for supporting news filtering on a web site. In section 4, we discuss how the techniques of this paper may be applied to generic portal-based applications. The conclusions and summary are presented in section 5.

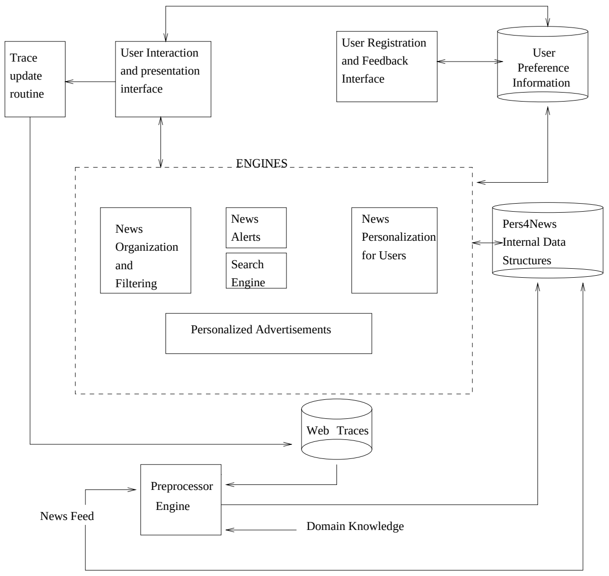
Trace update routine
User Interaction and presentation interface
User Registration and Feedback Interface
User Preference Information
ENGINES
News Organization and Filtering
News Alerts
Search Engine
News Personalization for Users
Pers4News Internal Data Structures
Personalized Advertisements
Web Traces
Preprocessor Engine
News Feed
Domain Knowledge

Figure 1: The System Architectures

2 System Architecture and Engines

The overall system architecture for the method is illustrated in Figure 1. The system has four main components; the user interface component which holds the software for the users to interact with the system (including backend software for generating traces from user behavior), the pre-processing software which uses the trace, domain knowledge and newsfeed to preprocess the data, the pre-processed data itself, and the different engines which use this pre-processed data in order to actually perform the recommendations.

The idea of separating the pre-processor from the engines which provide the recommendations is a simple but important one. The text mining algorithms and models which are used for finding the trends and patterns in the data are often compute-intensive and require hours in order to process effectively. On the other hand, a personalization application has online requirements; recommendations and personalization needs to be performed when the users access web pages. This leads to a natural OLAP (Online Analytical Processing) approach to personalization; separate out the tasks which cannot be performed in real time and process them periodically in order to create intermediate data structures and characterizations. The exact rate at which one is willing to perform this periodic processing depends upon the computational resources available at that site. Clearly, more frequent processing provides up-to-date recommendations. A natural choice is to perform the processing daily during the night.

Some examples of the pre-processed data contained in the internal data structures include summary information on user browsing behavior, site specific text behavior, temporal text trend behavior and the statistics from the different models which are executed periodically. These characterizations, trends and statistics are stored in the internal data structures of the architecture. In addition to the pre-processed data, the internal data structures also store the raw text of the news articles. The various engines of the system interact with these data structures in order to deliver the personalized results in real time. These results are then utilized by the presentation interface (see Figure 1) which dynamically constructs the personalized web page for the user.

It is assumed that each user of this system can be identified in some way, typically through the use of a registration interface. During this registration process the user may also be prompted for his preferences which are used for the purpose of personalized presentation. This information is stored in the User Preference Information block of Figure 1. Subsequently, the user needs to login whenever he uses the system, so that the appropriate part of the user preference and behavioral profile can be identified. The various engines of the system have access to this user preference behavior in order to personalize the presentation of the news. Details of each user's browsing activities are kept track of with the use of a trace update routine. This may be used for the automated part of the personalization (which does not require user feedback). The various engines for performing the personalization are described in greater detail in later subsections.

2.1 Data Preparation and Pre-processing

In this section, we discuss the techniques for data preparation and pre-processing which are necessary for the effective use of of the software. This pre-processed data is stored in the internal data structures of the software. The actual process is performed periodically in a batch mode. The resulting data structures are used to provide the users with personalized web pages in real time. The pre-processing is of the following kinds:

- Independent pre-processing: This kind of pre-processing is independent of the actual user behavior or the domain information available at the site. The only purpose is to find the lexical-chains in the news in order to determine the aggregate lexical patterns. It has been observed in earlier work [12, 13] that many text retrieval algorithms face a qualitative performance problem because of the inherent synonymities and ambiguities (polysemy) in textual descriptions. Thus, two documents containing very different vocabulary could be similar in subject material. Similarly, two documents sharing considerable vocabulary could be topically very different. Often, the ambiguity of the term can be resolved only by viewing it in the context of other terms in the document. In order to deal with this issue, we used pre-processing techniques in order to construct lexical-chains. Each chain thus generated describes a set of words which are related to similar topics and hence cooccur frequently in documents. This is like an automatic thesaurus. A typical example of a lexical-chain could be:
  army (22), troops (18), regiment (9), barracks (12)...
  The numbers in brackets denote the weights (relative importance) corresponding to each topic. This lexical-chain may be generated using word-clustering techniques as discussed in [7].[1] We will see that these lexical chains can be effectively used to determine the summary trends in the data.

- Domain dependent pre-processing: As indicated earlier, it may be desirable to categorize and filter the news-feeds depending upon a site-specific categorization. To this purpose, the sets of topics along with sample articles for those topics are provided. These are used in order to find the most relevant lexical chains to the different topics. This information is in-turn used for categorization and organization of the news feeds as they come in. More details of this technique are provided in a later section. Note that in many cases, the domain-specific information may also be obtained from the news-providers themselves, if the topical categorization of the news provider matches that of the site manager.

- User-specific pre-processing: This kind of pre-processing has to do with effective tracking of the aggregate user-behavior. In order to do so, the web traces are processed periodically and the access patterns of individual users is characterized. Then clustering techniques are used in order to partition the database into groups of similar users. The prominent articles accessed by the users of different clusters are characterized and indexed during this process. The index structures required for implementing collaborative filtering techniques are also constructed during this phase. We provide more details about these individual techniques in a later section. In addition, the models for delivering personalized advertisements are formulated and solved during this phase. The resulting summary parameters are stored for later use during the recommendation process.

- Site-specific pre-processing: It is useful to maintain the aggregate access of the different articles over the past few days in order to determine the

[1] A typical estimate of the number and length of the lexical chains are as follows: in one of our applications there were 800 lexical-chains, and a typical lexical-chain contained an average of 100 words.

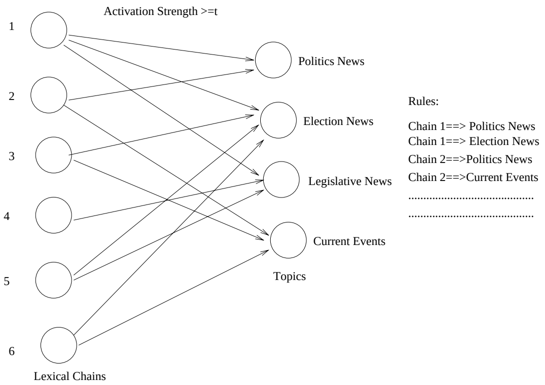

Figure 2: Matching lexical-chains to topics

hot topics and trends. This is achieved by periodically processing the web traces and storing the resulting statistics separately.

The pre-processor maintains a set of update API's which invoke the routines which maintain the internal data. It is preferred to invoke these routines periodically, along with some flexibility to invoke the API's when there are large amounts of new data arriving into the system.

2.2 News Organization and Filtering

In order to perform the training, we first match the various categories of news articles to the different lexical-chains. In order to do this we would like to get an idea of the nature of the similarity between the lexical chains and the news articles from various topics. First, we convert each lexical-chain into a meta-document by treating the weight of each word in the chain as a pseudo-frequency. Then, we calculate the average similarity of the documents in that topic to the lexical-chain using a standard similarity measure (such as the cosine, dice or jaccard coefficients [22]). This similarity value is referred to as the *activation weight*. An lexical-chain is related to a given topic, if the corresponding activation weight is above a pre-defined activation threshold t. This pre-defined parameter is chosen as matter of experience by the domain expert. An example of a small subset of such a relationship is shown in Figure 2. This relationship is used to generate different kinds of rules which match the lexical chains to the different topics relevant to that web site. The affinity of a given rule is determined by the corresponding activation weight between the chain on the LHS (Left Hand Side) of the rule and the topic on the RHS (Right Hand Side) of the rule. These rules are stored in the internal data structures of the system architecture as illustrated in Figure 1.

In order to organize the news articles, we first find all the lexical-chains whose similarity to the document is larger than t. Then the documents are represented in terms of these lexical chains for the purpose of classification. For a given document, we assign a score to each of the topics by summing the affinities of the all the rules which are relevant to the chains in that document and the corresponding topic. For example, consider the case when a particular news article contains chains 3, 5, and 6 of Figure 2, and the affinity values for the relevant rules are as follows:
Chain 3 $\Rightarrow_{0.23}$ Election News
Chain 3 $\Rightarrow_{0.21}$ Current Events
Chain 5 $\Rightarrow_{0.18}$ Election News
Chain 5 $\Rightarrow_{0.31}$ Legislative News
Chain 6 $\Rightarrow_{0.20}$ Election News
Chain 6 $\Rightarrow_{0.27}$ Current Events
The corresponding affinity values are indicated just below the $\Rightarrow$ sign. Thus, the news article is relevant to Election News (score $= 0.23 + 0.18 + 0.20$), Current Events (Score $= 0.21 + 0.27$), and Legislative News (Score $= 0.31$). These scores may be used in order to organize and filter the news articles by finding the highest scored news articles for each topic. Thus, some news-articles may be relevant to multiple topics or no topic at all for that particular site. As the newsfeed enters the site, the categorization and filtering engine first determines the corresponding lexical chains in the article, and then the relevant rules which need to be fired. For this purpose, the categorization and filtering engine interacts with the internal data structures which index the rules by the different lexical chains on the left hand side of the rule. These indices are periodically generated during the pre-processing phase. Once these chains have been determined, the appropriate scores for each topic are calculated and the news articles and filtered and organized into the proper sections.

2.3 Providing Personalized News Items to Users

The system can also provided personalized news which is tailored to individual users. This system can be effectively implemented when there is a system for tracking the news items which are being accessed by the users. The tracking could be either session-specific or could extend over multiple sessions. In order to do so, we track the behavior of users in terms of the lexical-chains present in the news articles accessed by them. A lexical-chain is said to be present in a news article, when its similarity of the corresponding meta-document to the news article is larger than the pre-defined activation threshold t. Thus, a data representation is created in which we have one record for each subscriber, and each feature is a continuous variable corresponding to a lexical chain. The feature value for the feature variable corresponding to a given lexical chain is its relative frequency with respect to all lexical chains accessed by the subscriber. Then, clustering techniques are used in order to determine the clusters of most similar customers. Note that straightforward clustering techniques are impossible to use in such cases, because of the extremely large dimensionality[2] of the problem. For such cases, the *dimensionality curse* becomes a serious impediment. Recently proposed projected clustering techniques [6] can be used effectively in such situations. These methods determine clusters by adaptively finding subsets of points together with subspaces in which those subsets of points cluster well. For each cluster, the most commonly read news articles are determined. In order to do this, we need to maintain indices associated with each cluster so that the relevant news articles can be quickly identified. Personalized news items are recommended to individual customers by presenting the news-articles which are read by other users who are in the same or nearby clusters.

In addition, users may also be able to pick and choose news items which are of special interest to them. In order to provide this capability, a user-interface needs to be provided, which can be used in order to provide feedback about individual user interests. This feedback may be used in order to directly organize and filter the news-articles for individual users directly based on their interests. Another option is to use collaborative filtering [4, 20] techniques which identifies other news-articles which have been read by similar users. Our techniques differ from some of the earlier methods in that it recognizes the sparsity of ratings that users are willing to volunteer. This makes it difficult to build a peer group of other similar users since there are not many common items which have been rated by the different users. For this purpose, we define peers by defining a directed graph in which each user is represented by a node, and an edge is determined by the commonality in ratings of two users. Two nodes are connected by an edge if (1) enough number of commen items are rated by both users. (2) The pattern of the ratings of one user can be transformed to the other with the use of a simple linear transform. A detailed description of the method may be found in [4]. A peer-group of an individual user is constructed by finding all the users within a certain radius of a node. Thus, this generalizes the notion of peer groups to include users who are indirectly connected to the target user, by a path of common pattern of ratings. Then, we find the most frequent news-articles which have been read by this peer group and report these as possible recommendations. In order to do so, we maintain inverted indexes which maintain a list of the users who have accessed each article. These inverted indexes are generated periodically during the pre-processing phase discussed earlier.

2.4 Personalized Advertisements

Advertising is a considerable source of revenue for web sites, and has often been successful in paying for various services such as email [24]. This also has considerable potential for content providers such as news services. Advertisements on web pages typically occur as either inline or text links. We assume in our model that each web page on a server has a certain number of standard sized *slots* containing either the inline images themselves or text links to the actual pages. We say that an advertisement is *exposed* when a web page containing a slot with the advertisement is served to a client. We say that an advertisement is *clicked through* when a client chooses the link corresponding to an exposed advertisement. The probability of a client viewing such a link when it is actually presented is dependent on how well the advertisement matches the interests of that person. Typically advertisers are willing to pay much more for placing advertisements in sites where the administrator can target the advertisement to the population viewing it. Specifically, advertisers often measure the number of times a person sees and/or clicks on an advertisement. The cost of an advertising contract may often depend upon the rate of click-throughs on an advertisement when it is exposed.

Such advertisements can be tailored to the content of the document and therefore to the user viewing them. The idea here is to maximize the probability that users actually click-through on advertisment icons when they are exposed on a given web page. In addition, web sites may have contractual obligations and pricing models for exposing certain advertisements for a pre-specified number of times in a given time-period. Our technique uses a network flow optimization model in order to calculate the number of times advertisements are assigned to various pages in order to maximize the click-throughs and also meet the contractual obligations for advertisement exposures. The model admits flexible pricing models and contractual arrangements. In addition, it takes into account factors such as web page access rate distribution, content affinities and user behavior. The actual model is periodically solved by the pre-processor in batch mode using the data collected from user access statistics. The model then provides intermediate data which is stored in the internal data structures. This intermediate data is used in order to make quick decisions on real time assignments of advertisements to requests for web pages.

The advertisement model takes the popularity and the content classification of the pages into account for the purpose of performing the assignments. This is done by a two-stage process. We note that even though the initial stage is not absolutely necessary, it is recommended, since a few simple pieces of input from the site administrator can improve the effectiveness dramatically. The first stage is a rough “keyword assignment” stage in which the site administrator assigns a certain number of specific keywords to each of the the

[2] As discussed earlier, a typical dimensionality would be around 800.

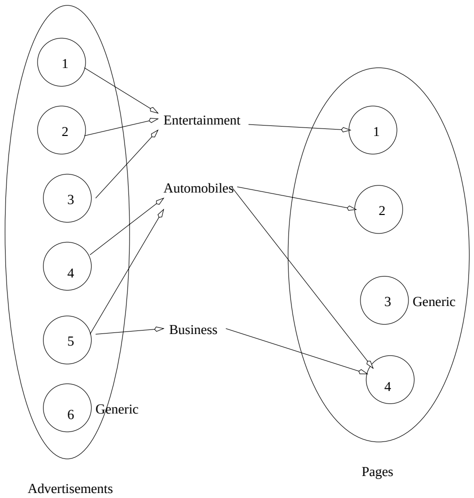

Figure 3: Keyword Matching of Advertisments and Pages

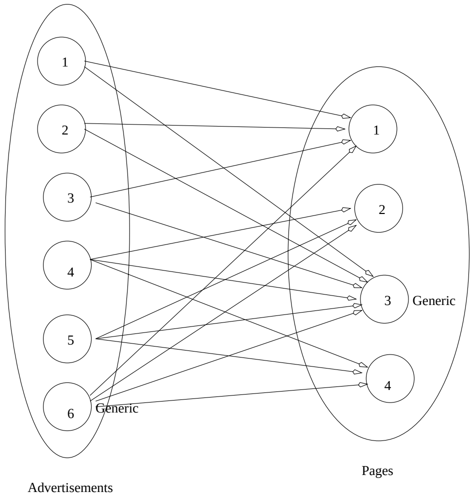

Figure 4: Possible Assignments using Keyword Matching

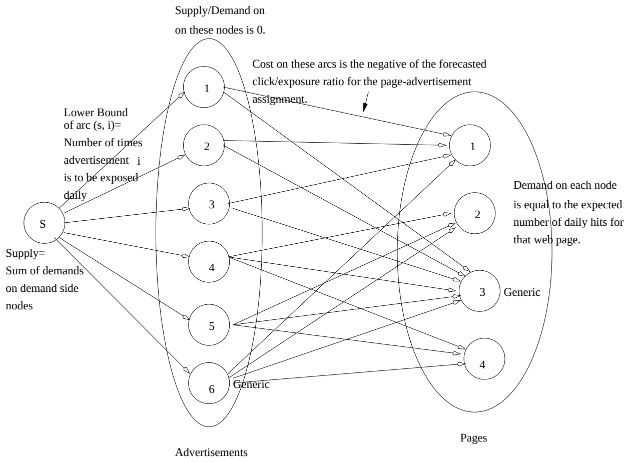

Figure 5: The model corresponding to the advertisement assignment problem

web pages corresponding to their content. (We assume that a keyword “generic” is also available as a “catchall phrase” so that lazy administrators have the choice of allowing the process of assignment to completely depend upon statistical information if necessary.) The precise choice of keywords is site specific and is based on the administrator’s understanding of a rough classification of the kinds of advertisements which may be hosted on that server. This process is used to generate what we refer to as a “possibility graph”, corresponding to the possibilities for assigning advertisements to pages. An example of a keyword assignment process is illustrated in Figures 3 and 4. Figure 3 shows an initial keyword assignment, while the Figure 4 (possibility graph) shows all the corresponding possibilities for actual assignments of advertisements to pages.

The construction of the possibility graph is a key step in obtaining good assignments of web pages to advertisements. The primary focus of this section is to show how good assignments may be generated by solving a network optimization problem for which the parameters are determined by using statistically forecasted data. An important piece of statistical information used in order to create the model is the *click-exposure ratio*.

Click/Exposure Ratios: The click exposure/ratio is the ratio of the number of times an advertisement is accessed to the number of times a user views the page containing the advertisement itself. The higher this value, the greater the success rate of the advertising process. The click/exposure ratio is specific to both the page which is being accessed, and the particular advertisement occuring on that page. In order to estimate this value, we use the click and exposure statistics of each day. At the end of each day, the statistics of the click-exposure ratio for each pair of web pages and advertisement are collected. At the beginning of each day, the data of the previous day is used in order to solve a network optimization model (as will be discussed in Figure 5) which decides the final advertisement assignments to pages. This is done by the preprocessor module of the system.

We now discuss how to actually construct the optimization model in order to find a solution to the advertisement assignment problem. The model assumes that we wish to maximize the total number of click-throughs on advertisements. Hence, in this case, the model is relatively simple, and does not discuss the possibility of offering flexibility in contractual arrangements.

We construct the network $G = (N_1 \cup N_2 \cup \{s\}, A)$ in which each node $i \in N_1$ corresponds to an advertisement, and each node $j \in N_2$ corresponds to a web page. An arc (i, j) from $i \in N_1$ to $j \in N_2$ exists if and only if the advertisement i is a valid assignment for page j in the possibility graph. The capacity of each of these arcs is ∞, and the flow x_{ij} denotes the number of times that the advertisement i should be assigned to web page j. The cost of the arc (i, j) is the negative of the click/exposure ratio for that page-advertisement assignment, and thus represents the degree of desirability to assign the advertisement to the corresponding page. Note that the network G is obtained from the possibility graph by adding a source node s and adding arcs from the source node to each node $i \in N_1$. We would like the flow on each of these arcs (s, i) to represent the number of times that advertisement i is exposed. Consequently, in conjunction with the contractual requirements between the site administrator and the advertiser, the lower bound on the arc (s, i) is equal to the number of times that the advertisement i needs to be exposed on a daily basis. The cost on each of the arcs emanating from the source node is zero units. The demand of node $j \in N_2$ should be equal to the number of times that the web page is viewed on a daily basis. This value is estimated using the web logs which track hits to each web page. The supply of the node s is equal to the sum of the demands on all the web page nodes. The overall model is illustrated in Figure 5.

After a minimum cost flow is obtained in this network, the flow x_{ij} on the arc (i, j) determines the number of times that the advertisement i should be assigned to the web page j on that day. The total cost of the flow is the negative of the total expected number of click-throughs of advertisements. Therefore, this mechanism maximizes the total number of click-throughs of the advertisements.

Once the preprocessor module has solved the optimization model, the advertisement engine uses this information in order to make the advertisement assignments to requests for web pages in real time. A simple way of using this information in order to perform the assignment is to probabilistically assign the advertisement i to a request for web page j with probability proportional to the corresponding flow value x_{ij}. It is also possible to generate more involved models which take the contractual obligations of a user into account. We will omit a detailed discussion of this model, which is provided in [5].

2.5 Detecting Hot Topics and News Alerts

An important issue is to be able to detect sudden changes in the the trends of the various news items. Some methods for finding hot topics in newsfeeds [21] have been proposed, but these cannot detect arbitrary changes in trends, but are dependent on a fixed set of pre-decided topics. On the other hand, our system is capable of detecting any sudden changes in news patterns. In order to do so, we maintain the frequencies and standard deviations of the various lexical-chains in the different documents. For each of the lexical-chains, we maintain the following information:

- $\mu(w)$: Mean fractional frequency of a given lexical-chain w for each of the past n days.
- $\mu_i(w)$ Mean fractional frequency of a given lexical chain w for the ith day during the last n days.
- $\sigma(w)$: Variation in the fraction frequency of each lexical-chain in each of the past n days. The value of $\sigma(w)$ is calculated as follows:

$$\sigma(w) = \sqrt{\frac{\sum_{i=1}^{n} (\mu(w) - \mu_i(w))^2}{n-1}} \qquad (1)$$

- $f(w)$: Fraction frequency of a given lexical-chain on current day.

All these parameters are calculated by the preprocessor module of the system. In order to detect abnormal variations, we calculate the deviation-value for each lexical-chain which is given by:

$$\text{Deviation}(w) = \frac{f(w) - \mu(w)}{\sigma(w)} \qquad (2)$$

This value characterizes how much of the deviation of a lexical-chain presence above average presence can be characterized by randomness.[3] All chains with a value of $Deviation(w)$ which is above a certain threshold (which was typically chosen to be 3) were identified. All news-items in the current day which are activated by these lexical-chains are reported in the news-alert. One of the interesting aspects of this technique is that the use of lexical-chains in order to find hot-topics results in detection of trend-changes which are independent of the topical structure at that site.

[3] Under the assumption of a normal distribution on the fractional presence, a deviation value of +3 indicates excessive presence with 99.9% probability.

In addition to the above technique for hot topic generation in the current architecture, many other techniques for detection and monitoring of temporal variations in news patterns are being developed. Similar techniques can also be used for determing trend detection over longer time periods or seasonal variations in news patterns. For example, the end of every two years sees a sudden spurt in election news because of the congressional and other election news; the end of every year sees a variations in news patterns because of the holiday season and so on. These variations can be captured and presented in the form of seasonally hot topics. The above-discussed methods suffice to find the hot topics in the current season by changing the length of time over which the hot-topic is calculated. When the same techniques as discussed above are used with higher thresholds on the time-periods for which variations in behavior are calculated, then more long-term variations in news-patterns may also be calculated. In order to detect the periodicity of the patterns, methods for detecting cyclicity in the patterns [15] may be used.

Interesting long term changes in the trends [14] of news patterns may also be interesting to detect. One possible solution is to look at the gradual variations in the patterns of the different chains which are accessed by users. These patterns could vary because of changes in the demographics of the users visiting the web site, or the changes in interests of the users themselves. In either case, this information is useful for detecting important changes in trends in the data.

2.6 Contextual Search Capability

The system is capable of contextual search capability which performs effective document-to-document similarity search. This is a key-word search engine which searches documents based on the dominant topics present in them by relating the keywords to the different topics. This is of somewhat similar flavor to Latent Semantic Indexing techniques [13], except that our techniques are significantly more efficient in providing results to users in real time. In order to do so in a fast and scalable way, internal indices need to be built which allow quick determination of topical behavior of the relevant documents, which is characterized by the lexical chains. In order to do this, we maintain a list of all the Document Identifiers which show significant presence within a given lexical chain. Thus, we maintain inverted lists for each lexical chain. The indexes are built and updated periodically as a pre-processing technique. In order to find the matching documents for a given target, we perform a two-stage process. In the first stage, we find all the lexical chains which are contained in a given document. In the second stage, we compute all those documents which contain these lexical chains with the use of this index.

3 Application and Interfaces for real systems

The system discussed above has so far been implemented in two applications:
(1) A web site for a specific company which personalizes news based on certain general categories. In addition to news customization, the web site is also capable of other kinds of customization which are not described in this paper. (2) An internal corporation web site which categorized the news feeds based on different industry service units, and presented personalized items to different users.
The above two systems provided different sets of features, depending upon the individual needs of that site. In this paper, we will discuss details of the former case.

The company specific web site supported user-registration which was helpful in tailoring the news to the individual users. Thus, new users are prompted to register and provide information about themselves. In addition, a user is prompted for his subject preference information. This preference information is used for making the recommendations. As illustrated, the interface provides options to the user in indicating the topics that he is most interested in. Subsequently, a user needs to login in order to use the site. The login of a user into the site initiated a session which was kept track of on an individualized basis by the web traces.

The right hand side of the browser display contained the headline news and company-related news, whereas the left hand side contains the news items which are tailored to individual users. We proceed to describe each of these two categories:

- Current Headline News and company specific news: These news could be presented by using both the lexical chains in the news article and a direct knowledge of the relevance of the keywords occuring in the news. For example, if the keyword corresponding to the company name appeared in the news, then this was indicative of the fact that the news was company related. Further, lexical-chains which were strongly related to the company news were also used in order to identify the important news items. Current headline news was presented separately based on the latest news items received. In addition, our software also has the capability for detecting hot-topics, which could be included in the headline news.

- Categorization and Filtering based on user-specified topics: The system provided categorization and filtering based on user-specified topics. For this purpose, a user-interface provided a number of choices to the users. This process of checking the preference items is done in a one-time process during new user registration. Correspondingly the left side of browser contained a list called *My Personal News.* Each of the links in this list points to the news items in the corresponding topics. In order to categorize the news based on the different topics, we deteriming the dominant lexical-chains in each news article and matched them to the different topics, as described earlier.

This kind of implementation is based on the preference to give users greater control over the news that they wish to browse. Therefore we used explicit specification by using an interface. (as opposed to providing users with personalized news-items based on their access behavior.) Our other application based on the internal site was completely automated and required no manual feedback. The choice of using a completely automatic system or a manual feedback interface is based on the natural tradeoff between giving users greater control and the ease in obtaining the feedback when it is indeed solicited. An additional feature to be kept in mind while designing such systems is to not let the personalization process become invasive [11] as it may often violate privacy concerns for web site management. The process of soliciting individual feedback is usually considered less invasive since any information used for personalization is based on information which is volunteered by the user.

4 Applications to Personalized Portals

One of the components of the project is on building intelligent tools for personalization of portals. This may extend to providing personalized recommendations for specific items on e-commerce applications, or may involve providing personalized advertisements. Many of the news feed filtering, organization, and mining techniques are being used in this system because of the natural synergy of both applications in using similar kinds of text mining tools.

Some examples of features which could be provided in a personalized portal in a similar way to the methods discussed for news personalization are as follows:

- Personalized Tracking of User Interests: The browsing pattern of a user at a portal site may be used in order to determine his or her interests. Alternatively, individual feedback may be solicited from the user in order to determine the set of topics that he is most interested in. This is similar to the system we discuss for tracking user behavior via his personalized interests.

- Individualized Recommendations for web pages: At a conceptual level there is very little difference between providing automated recommendations for news articles and providing the same for individual web pages. Often it may be desirable to recommend individual web pages to users at the portal. These web pages could correspond to the main links for different electronic commerce sites. For example, if it is already known that a user is interested in CDs or Videos, then the portal may provide him with recommended hyperlinks to the web sites for different electronic commerce sites. Often, it may be desirable to recommend sets of promotion items to individual customers. In order to do so, we can use the text descriptions of the individual promotion items and recommend them in a similar way as individualized news articles for users.

- Personalized Advertisements: A useful ability of a portal is to be able to place advertisements depending upon the individual user browsing the site. To this effect, web pages may often be constructed dynamically, with appropriate slots for advertisements. To this effect, if the user-interests are already known, then it is relatively easy to place in those advertisements which he is most likely to be interested in. The implementation of this requires the same techniques as the placement of promotion items if there are no contractual obligations for advertisements. On the other hand, when such obligations are indeed present, then the models and techniques discussed here can be readily modified in order to deal with this situation.

- Determining Hot Topics on the Portal: Often a portal may experience sudden spurts of interest in individual topics. Note that in this case, we are referring to the *user access behavior* as opposed to the news feed behavior. These are useful to detect, since they can be exploited for the purpose of time-targeted advertisements. If it is known that certain topics are of greater user interest at a given moment, then the advertisements on that subject may be given prominent slots on the portal site. The process of determining the interesting topics is similar to the process of determining the hot topics in a news feed at a web site.

In addition, the generic architecture for implementing the user tracking is very similar to the architecture discussed in earlier sections for the management of newsfeeds. These techniques are currently being implemented in a personalized portal project.

5 Conclusions and Summary

In this paper, we described a system which is capable of effective organization, tracking and personalization of a web portal containing a news feed service. We discussed the overall architecture as well as the individual techniques which were used by this system for effective content management of newsfeeds, detection and reporting of news alerts, personalized presentation and search capability. The system has been developed as a complete package, with an emphasis on developing novel, fast, accurate and scalable techniques. The actual implementation and user-interface for a version of this software was discussed. We are currently working on the development of an integrated system architecture for personalized portals based on the methods discussed in this paper.

References

[1] G. Adomavicius, A. Tuzhlin. User Profiling in Personalization Applications through Rule Discovery and Validation. *KDD Conference*, 1999.

[2] C. C. Aggarwal, S. C. Gates, P. S. Yu. On the merits of using supervised clustering for building categorization systems. *KDD Conference*, 1999.

[3] C. C. Aggarwal, P. S. Yu. Data Mining Techniques for Personalization. *Data Engineering Bulletin*, March 2000.

[4] C. C. Aggarwal, J. L. Wolf, K. L. Wu, P. S. Yu. Horting Hatches an Egg: Fast Algorithms for Collaborative Filtering. *KDD Conference*, 1999.

[5] C. C. Aggarwal, J. L. Wolf and P. S. Yu. A Framework for the Optimizing of WWW Advertising. *Springer-Verlag- Lecture Notes in Computer Science*, Vol. 1402, pp. 1-10, 1998.

[6] C. C. Aggarwal, P. S. Yu. Finding Generalized Projected Clusters in High Dimensional Spaces. *SIGMOD Conference*, 2000.

[7] C. C. Aggarwal, P. S. Yu. On effective conceptual indexing and similarity search in text. *ICDM Conference*, 2001.

[8] C. C. Aggarwal, P. S. Yu. On Text Mining Techniques for Personalization. *Lecture Notes in Computer Science*, Vol. 1711, pp. 12-18, Springer, 1999.

[9] M. Balabanovic. An adaptive web page recommendation service. *First International Conference on Autonomous Agents*, 1997.

[10] M. Balabanovic, Y. Shoham. Fab: content-based collaborative recommendation. *CACM*, Volume 40, no 9, pp. 55-72, March 1997.

[11] P. K. Chan. A non-invasive learning approach to building web user profiles. *KDD99 Workshop on Web Usage and User Profiling.*

[12] S. Chakrabarti, B. Dom, R. Agrawal, P. Raghavan. Using taxonomies, discriminants and signatures for navigating in text databases. *VLDB Conference*, 1997.

[13] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, R. A. Harshman. Indexing by Latent Semantic Analysis. *JASIS*, 41(6), pp. 391-407, 1990.

[14] B. Lent, R. Agrawal, R. Srikant. Discovering Trends in Text Databases. *KDD Conference*, 1997.

[15] B. Ozden, S. Ramaswamy, A. Silberschatz. Cyclic Association Rules. *ICDE Conference*, 1998.

[16] Y. Freund, R. Iyer, R. Shapire, Y. Singer. An Efficient Boosting Algorithm for Combining Preferences. *International Conference on Machine Learning*, Madison WI, 1998.

[17] D. Greening. Building Consumer Trust with Accurate Product Recommendations. LikeMinds White Paper LMWSWP-210-6966, 1997.

[18] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordan and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, Vol. 40, No. 3, pp. 77-87, 1997.

[19] P. Resnick and H. Varian. Recommender Systems, *Communications of the ACM*, Vol. 40, No. 3, pp. 56-58, 1997.

[20] U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating Word of Mouth. *SIGCHI Conference*, 1995.

[21] M. Shewhart, M. Wasson. Monitoring a newsfeed for hot topics. *KDD Conference*, 1999.

[22] G. Salton, M. J. McGill. Introduction to Modern Information Retrieval. *Mc Graw Hill*, 1983.

[23] T. W. Yan, H. Garcia-Molina. Index Structures for Information Filtering Under the Vector Space Model. *ICDE Conference*, 1994.

[24] http://www.hotmail.com