ISSN:0975-9646

Content Based Recommendation System Using SOM and Latent Dirichlet Allocation Model

Amit Kumar Nandanwar, Geetika S. Pandey

Dept. of Computer Science, S. A. T. I.,
Vidisha, India

Abstract— **The content-based recommendation systems, is a systems that recommend any information to a user based upon a description of the page or document and a profile of the user's interests. Content-based recommendation systems may be used in a variety of domains ranging from recommending web pages, news articles, restaurants, television programs, and items for sale. Although the details of various systems differ, content-based recommendation systems share in common a means for describing the items that may be recommended, a means for creating a profile of the user that describes the types of items the user likes, and a means of comparing items to the user profile to determine what to recommend. This paper presents a novel content-based recommendation method which recommends web pages content of a user's recent interests in a page. Traditionally, a web page is recommended based on a comparison between a user's profile and web contents that are represented as a set of feature keywords. This paper proposes a new approach to representing and extracting page using SOM and Latent Dirichlet Allocation (LDA) Model. Most of the techniques presented are based on URL or title of pages for recommendation this method is only based on content of a page for recommendation. This paper proposed Content Based Recommendation System Using Latent Dirichlet Allocation Model.**

Keywords— **SOM, LDA, Recommendation system, data mining.**

I. INTRODUCTION

Internet has stirred the fast development of web sites equipped with rich resources in a variety of application sectors. However, on-line readers are often apt to get lost in such an environment due to its complicated structure and huge amount of information. Therefore, a new design method that can adapt a Web site to user needs is of great importance to improve the usability and user retention of the Web site. The success of such an adaptation feature, also called Web personalization, heavily relies on the system's capability to anticipate users' future needs. Web personalization already finds important applications in e-business (such as Amazon.com and google.com), e-learning and so on.

As the World Wide Web continues to grow at an exponential rate, the size and complexity of many web sites grow along with it. For the users of these web sites it becomes increasingly difficult and time consuming to find the information they are looking for. To help users find the information that is in accordance with their interests a web site can be personalized. Recommender systems can improve a web site for individual users by dynamically adding hyperlinks. Recommendation systems produce a ranked list of items on which a user might be interested, in the context of her current choice of an item. Recommendation systems are built for movies, books, communities, news, articles etc.

Every large collection needs a certain structure to make it easy for visitors to find what they are looking for. A web site can be structured by dividing its web pages into content pages and navigation pages. The content pages provide the user with the interest items while the navigation pages help the user to search for the interest items. This is not a strict classification however. Pages can also be hybrid in the sense that they both provide content as well as navigation facilities. Furthermore, what is a navigation page for one user may be a content page to another and vice-versa. In general however, this classification provides a way of describing the structure of a web site and how this structure can be improved for individual users by dynamically adding hyperlinks.

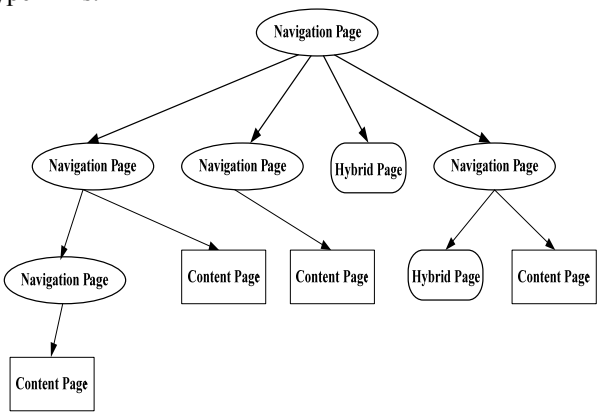

Fig. 1 Structure of sample web site

Figure 1 shows an example of a web site with a typical tree structure. The content pages are found at the bottom of the tree while the navigation pages are found at the top. A recommender system can display its recommendations by dynamically creating hypertext links to content pages that contain the items a user might be interested in. Several factors determine whether or not a recommended page should be linked to the page that is shown to the user. Sometimes content pages are only recommended if they contain items that are similar to the item(s) shown on the current page. Another consideration for dynamic linking is the proximity to the recommended page. The distance between two pages is determined by the minimal number of links it takes to navigate from one page to another. There is not much use of linking the current page to a recommended page if the distance between the two pages is 1. The further the distance, the more useful a dynamically created link becomes.

II. RELATED WORK

A. Recommendation Systems

The popularization of computers and the Internet have resulted in an explosion in the amount of digital information. As a result, it becomes more important and difficult to retrieve proper information adapted to user preferences [2][5]. In general, there are two types of recommendation systems, collaborative filtering systems [3][4] and content-based filtering systems [7, 9, 18, 12,37].

B. Collaborative Recommendation

In collaborative filtering, items (e.g., web pages) are recommended to a particular user when other similar users also prefer them. The definition of "similarity" among users depends on applications. For example, the similarity may be defined as users having similar ratings of items or users having similar navigation behavior. This kind of recommendation systems is the first one that uses the artificial intelligence technique to do the personalized job [5]. A collaborative filtering system collects all information about users' activities on the website and calculates the similarity among the users. If some users have similar behavior, they will be categorized to the same user group. When a user logins into the web site again, the system will first compute the group most similar to the user, using methods like the k-nearest neighborhood, and then recommend to the user the items that the group members prefer. Examples of collaborative recommendation systems include the Amazon Net Book Store, Tapestry, Firefly, Referral Web, PHOAKS, Siteseer, GroupLens, Ringo and so on. However, a pure collaborative filtering system has several drawbacks and issues, including that the coverage of item ratings could be very sparse, hence yielding poor recommendation efficiency; it is difficult to provide services for users who have unusual tastes; and the user clustering and classification problems for users with changing and/or evolving preferences.

C. Content-based Recommendation Systems

The content-based approach to recommendation has its roots in information retrieval and information filtering [22] research. Because of the significant and early advancements made by the information retrieval and filtering communities and because of the importance of several text-based applications, many current content-based systems focus on recommending items containing textual information, such as documents, Web sites (URLs), and Usenet news messages. The improvement over the traditional information retrieval approaches comes from the use of user profiles that contain information about users' tastes, preferences, and needs. The profiling information can be elicited from users explicitly, e.g., through questionnaires, or implicitly learned from their transactional behavior over time.

More formally, let Content be an item profile, i.e., a set of attributes characterizing item s. It is usually computed by extracting a set of features from item s (its content) and is used to determine appropriateness of the item for recommendation purposes. Since, as mentioned earlier, content-based systems are designed mostly to recommend text-based items, the content in these systems is usually described with keywords. For example, a content-based component of the Fab system [21], which recommends Web pages to users, represents Web page content with the 100 most important words. Similarly, the Syskill & Webert system [25] represents documents with the 128 most informative words. The "importance" (or "informativeness") of word ki in document dj is determined with some weighting measure wij that can be defined in several different ways.

As stated earlier, content-based systems recommend items similar to those that a user liked in the past [27]. In particular, various candidate items are compared with items previously rated by the user, and the best-matching item(s) are recommended. More formally, let Content Based Profile be the profile of user c containing tastes and preferences of this user. These profiles are obtained by analyzing the content of the items previously seen and rated by the user and are usually constructed using keyword analysis techniques from information retrieval. For example, Content Based Profile can be defined as a vector of weights (wc1, …,wck), where each weight wci denotes the importance of keyword ki to user c and can be computed from individually rated content vectors using a variety of techniques. For example, some averaging approach, such as Rocchio algorithm [26], can be used to compute Content Based Profile as an "average" vector from an individual content vectors [21]. On the other hand, [25] use a Bayesian classifier in order to estimate the probability that a document is liked. The Winnow algorithm [23] has also been shown to work well for this purpose, especially in the situations where there are many possible features [24]. As was observed in [21, 28], content-based recommender systems have several limitations that are described in the rest of this section.

D. Keyword Extraction from Text Documents

One important research issue related to content-based recommendation is the keyword analysis for text documents so that their characterization can be extracted and represented. Often some weighting scheme is used to select discriminating words [18]. Some researchers adopt the multinomial text model [17] in which a document is modeled as an ordered sequence of word events drawn from the same vocabulary set. A naive Bayesian text classifier is trained to represent user interests and to produce rankings of books that conform to the user's preference [9]. The naive Bayes' assumption states that the probability of each word event is dependent on the document class but independent of the word's context and position. While this assumption might be valid for their book recommendation case, it is not applicable in the web page recommendation situation considered in this paper, since no pre-defined document classes are specified for each content page.

III. SELF ORGANIZING MAPS(SOM) AND K-MEANS ALGORITHM

SOM [35] is a kind of unsupervised learning technique of Neural Networks [35] which helps in reducing the high dimensional data into low dimensional data and visualizes that. Based on competitive learning principles, SOM helps in clustering data together for analysis and in clustering similar sessions together. By analyzing these clusters we can find frequently accessed pages by a set of similar users.

A. SOM Algorithm

1. Assign random values to weight vectors of a neuron.
2. Provide an input vector to the network.
3. Traverse each node in the network
 a) Find similarity between the input vector and the network's node's weight vector using Euclidean Distance.
 b) Find the node that produces the smallest distance which is the Best Matching Unit (BMU)
4. Update the nodes in the neighborhood of BMU by changing the weights using the following equation:

$$Wv\,t{+}1 = Wv\,t + (t)\alpha(t)(D\,t - Wv\,t)$$

Where,

- t denotes current iteration
- λ is the limit on time iteration
- *Wv* is the current weight vector
- *D* is the target input
- θ(*t*) is the neighborhood function In this algorithm neighborhood function has been derived using Gaussian function.
- α(*t*) is learning rate due to time

5. Increment t and repeat from step2 while *t*< λ. The k sessions and the set of m unique URLs are the input to the SOM network. The input is represented by a two dimensional matrix of order m x k.

B. K-means Algorithm

K-means [35,36] is also considered to be one of the important tools for clustering problems. K-means works using the following steps: 1. Place K objects points into the space that are to be clustered. object points always represent initial group centroids.
2. Assign each object point to the group that has *the closest* centroid.
3. When all object points have been assigned, re-calculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the object points into groups from which the metric to be minimized can be calculated. The algorithm aims to minimize an objective function:

$$J = \sum_{j=1}\sum_{i=1} \left\| x_i^{(j)} - c_j \right\|^2$$

Where $\left\| x^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point and the cluster centre. It is an indicator of the distance of the n data points from their respective cluster centers.

IV. LATENT DIRICHLET ALLOCATION

The general idea of Latent Dirichlet Allocation (LDA) is based on the hypothesis that a person writing a document has certain topics in mind. To write about a topic then means to pick a word with a certain probability from the pool of words of that topic. A whole document can then be represented as a mixture of different topics. When the author of a document is one person, these topics reflect the person's view of a document and her particular vocabulary.

In the context of tagging systems where multiple users are annotating resources, the resulting topics reect a collaborative shared view of the document and the tags of the topics reect a common vocabulary to describe the document. More generally, LDA helps to explain the similarity of data by grouping features of this data into unobserved sets.

A mixture of these sets then constitutes the observable data. The method was first introduced by Blei et al. [31] and applied to solve various tasks including topic identification [32], entity resolution, and Web spam classification [30]. The modeling process of LDA can be described as finding a mixture of topics for each resource, i.e., P(z|d), with each topic described by terms following another probability distribution, i.e., P(t|z). This can be formalized as

$$P(t_i|d) = \sum_{j=1}^{z} P(t_i|z_4 = j)\, P(z_i = j|d),$$

Where P(ti|d) is the probability of the ith term for a given document d and zi is the latent topic. P(ti|zi = j) is the probability of ti within topic j. P(zi = j|d) is the probability of picking a term from topic j in the document. The number of latent topics Z has to be defined in advance and allows to adjust the degree of specialization of the latent topics.

LDA estimates the topic term distribution P(t|z) and the document topic distribution P(z|d) from an unlabeled corpus of documents using Dirichlet priors for the distributions and a fixed number of topics. Gibbs sampling [32] is one possible approach to this end: It iterates multiple times over each term ti in document di, and samples a new topic j for the term based on the probability P(zi = j|ti, di, z-i) based on Equation 2, until the LDA model parameters converge.

$$P(z_i = j|t_i, d_i, z_{-i}) \propto \frac{C_{t_ij}^{TZ} + \beta}{\sum_t C_{t_ij}^{TZ} + T\beta} \frac{C_{d_ij}^{DZ} + \alpha}{\sum_t C_{d_ij}^{DZ} + Z\alpha}$$

CTZ maintains a count of all topic{term assignments, CDZ counts the document{topic assignments, z-i represents all topic{term and document{topic assignments except the current assignment zi for term ti, are the (symmetric) hyperparameters for the Dirichlet priors, serving as smoothing parameters for the counts. Based on the counts the posterior probabilities in Equation 1 can be estimated as follows:

$$P(t_i|z_4 = j) = \frac{C_{t_ij}^{TZ} + \beta}{\sum_t C_{t_ij}^{TZ} + T\beta}$$

$$P(z_i = j|d_i) = \frac{C_{d_ij}^{DZ} + \alpha}{\sum_t C_{d_ij}^{DZ} + Z\alpha}$$

V. PROPOSED ALGORITHM

Input: Corpus and Number of Topics.

Output: Topic proportions of documents and word topic distribution.

Note: Number of topic will be determined experimentally so play with different number of topics.

Step1: Collection of web corpus sample and user web navigation pattern.

Step 2: Preprocessing remove stop word, stemming and morphological analysis of corpus.

Step 3: Apply Latent Dirichelet Allocation on corpus to obtain a soft clustering on terms.

Step 4: Use some clustering algorithm to obtain cluster of document in corpus using topic proportion resulting from above step. (Like k-means, spectral, self organization map and some soft clustering algorithm).

Step 5: When user is navigating through web pages it is likely that his next web page will be somewhat related to

previous 2-3 pages browsed by him. Combine his last 2-3 documents into a single document and apply LDA inference on it to determine the topic proportions using Word-Topic Distribution Using the same distance measure as used in Step 4 for clustering find the distance of above vector from different clusters.

Step 6: Once the cluster is found, for each document in cluster find the closeness of last 2-3 documents to document. Output (say Top 5) documents which are close to last 2-3 Documents.

Step 7: Determine measures like Accuracy, Precision and Recall of proposed algorithm to evaluate how good it is.

As mention above proposed algorithm uses corpus and no. of topics as input for processing data these input got after preprocessing of dataset then through the processesing various tasks are applied on this data and finally top 5 recommended file are shown as result the whole system can be understand by the architecture given in Figure 2.

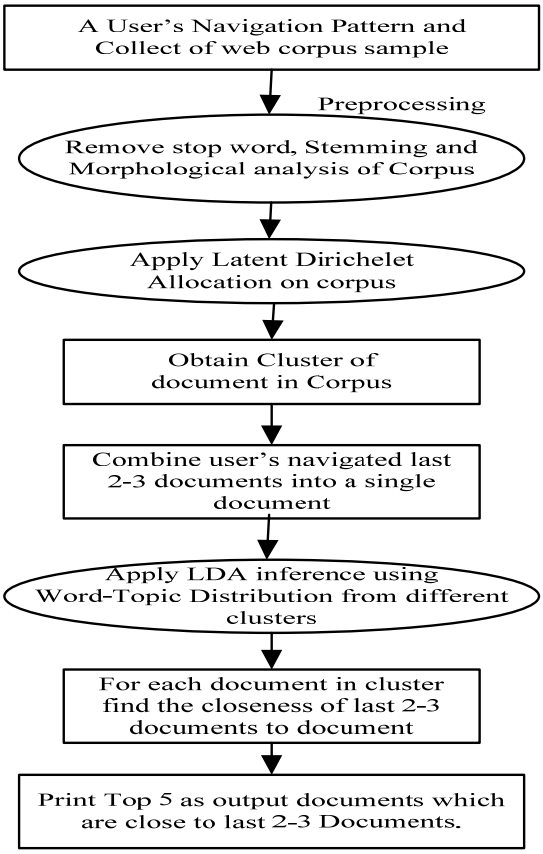

Fig. 2 Architecture of proposed Algorithm

VI. EXPERIMENTAL RESULTS

A. 20-Newsgroups Document Collection

The 20-newsgroups document collection is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups. It was originally collected by Ken Lang, probably for his paper [34]. Now, it has become a popular data set for the experiments in text applications of machine learning techniques such as text classification and clustering. In this document collection, each news group constitutes a different category, with varying overlaps between them; some news groups are much related and others are not related at all. The main purpose of choosing this collection is to test the capacity of the phrase-based document clustering approach against noise.

In our experiments, we directly used this data set. This data set contains 2,000 documents, 100 documents for each news group. For each document in the data set, the text of the message headers and e-mail addresses are ignored in our experiments. After the document preprocessing, the average length of documents is about 131 words.

B. Document Preprocessing

Stopwords are frequently occurring, insignificant words that appear in documents. They are useless to index or use in search engines and other information retrieval systems. Stopwords Lists and stemming algorithms are two commonly used information retrieval techniques for preprocessing text documents. We also use a standard stopwords List and he Porter's suffix-stripping algorithm [33] to process the documents to get "clean" documents. However, we note that there still exist some frequently occurring words slightly affecting the accuracy of the phrase-based document similarities.

Although tf-idf weighting scheme has provided a solution to reduce the negative effect of stopwords, almost all popular document clustering approaches including the STC algorithm still prefer to consider these words as the new stopwords, and ignore them in computing document similarities. For example, the STC algorithm maintains a stoplist that is supplemented with Internet-specific words, e.g., "previous," "java," "frames," and "mail." A word appearing in the stoplist, or that appears too often or rare in the documents receives a score of zero in computing the score s(B) of a base cluster.

Before the document clustering, a document "cleaning" procedure is executed for all documents in the data sets: First, all nonword tokens are stripped off. Second, the text is parsed into words. Third, all stopwords are identified and removed. Fourth, the Porter's suffix-stripping algorithm [33] is used to stem the words. Finally, all stemmed words are concatenated into a new document. Since the length of a word is variable, it is quite difficult to implement a suffix tree based on words directly. To solve the problem, we build a wordlist to store all keywords in alphabetical order. The similar ideas are often used in some text retrieval approaches for simplifying the computation complexity, such as the inverted index systems. In the wordlist, a unique integer number (called a word_id) is assigned to each keyword so that we can use the word_id to replace the corresponding word in the "cleaned" document. Finally, each document becomes an array of word_ids for the suffix tree construction.

C. Samples of Topics and User Navigational Preference Distribution

We use LDA to identify the semantics of latent topics from the contents of prominent pages contributing significantly to each topic. We first present 3 examples out of 30

discovered topics in Table 1. To illustrate theses topics, we also list the URLs of prominent pages as well as their corresponding probabilities (based on ø), respectively. Meanwhile, the estimate of each user session's association with multiple topics (θ) could be used to model each user's navigational preference over topic space.

TABLE I
EXAMPLES OF TOPICS DISCOVERED FROM 20-NEWSGROUPS DATASET

Topic 0th		Topic 1th		Topic 29th	
Topic	**Proba.**	**Topic**	**Proba.**	**Topic**	**Proba.**
Univers	0.01557	Israel	0.02438	Game	0.03274
State	0.01232	Jew	0.02242	Team	0.02136
Nation	0.00987	Arab	0.01719	Plai	0.01721
Inform	0.00925	Isra	0.01662	Player	0.01476
Institute	0.00859	War	0.01454	Win	0.01110
School	0.00847	Jewish	0.01334	Hockei	0.00926
Center	0.00809	Peac	0.00989	Season	0.00921
Confer	0.00742	State	0.00870	Fan	0.00800
Research	0.00718	Palestinian	0.00831	Score	0.00758
April	0.00664	Country	0.00823	Hit	0.00734
Include	0.00651	Attack	0.00790	Baseball	0.00725
Washington	0.00640	Kill	0.00780	Leagu	0.00677
Commun	0.00576	Nazi	0.00727	Goal	0.00576
Student	0.00573	Muslim	0.00708	Run	0.00564
Convent	0.00549	Nation	0.00680	Pitch	0.00558
Present	0.00544	World	0.00665	Period	0.00483
Educ	0.00537	American	0.00634	Playoff	0.00468
Publish	0.00532	Land	0.00613	Won	0.00444
Announc	0.00526	Occupy	0.00492	Nhl	0.00437
Contact	0.00523	Civilian	0.00476	Game	0.03274

D. The Performance Evaluation on Large Document Data Sets

Figure 3 is a snapshot of simulation of proposed method for client program. Client searches for Data\alt.atheism\51133 file for this file recommender system recommended file names as shown in screen. Table 2 shows the some values for proposed method results overall.

TABLE II
SOME RECOMMENDED RESULTS FOR SELECTING A FILE.

File Selected	Files Recommended using LDA
Data\comp.windows.x\67113	Data\comp.graphics\39012
	Data\sci.space\61351
	Data\comp.graphics\38706
	Data\comp.windows.x\67458
	Data\comp.sys.ibm.pc.hardware\60778
	Data\comp.windows.x\67100
	Data\comp.windows.x\67397
	Data\comp.graphics\38771
	Data\comp.graphics\38761
	Data\comp.sys.ibm.pc.hardware\61001
Data\rec.autos\102862	Data\comp.sys.ibm.pc.hardware\51948
	Data\rec.sport.baseball\104454
	Data\comp.os.ms-windows.misc\10605
	Data\comp.graphics\38978
	Data\sci.electonics\52736
	Data\comp.graphics\39061
	Data\comp.windows.x\67218
	Data\sci.space\60188
	Data\comp.graphics\39065
	Data\rec.sport.baseball\104363

Fig. 3 Snapshot of client selecting file and system recommended files.

Table shows result of proposed algorithm gives recommended files for further reading as user select file to read. Here are two files examples which show effectiveness of proposed algorithm for recommendation system using LDA model. In table first column have file which is user selected and second column show recommended files for currently selected file for user.

VII. CONCLUSION

In this paper we have investigated the use of Latent Dirichlet Allocation for content based recommendation. In general, our LDA-based approach is able to elicit a shared topical structure from the content based recommendation system using LDA effort of multiple users, whereas associations are more focused on simple terminology expansion. However, approach is succeed to some degree in overcoming the recommended content of individual content based practices. LDA achieves better accuracy, and in particular recommends more specific information, which are more useful for search.

The main contribution of Latent Dirichlet Allocation method is to offers efficient recommended content to user to make better searching during surfing. It could be very useful for any type of text information because this method uses all content for recommendation in place of topic of content or just url of content it gives better recommendation to user to make surfing easy. User can get automatically information of his interest. Regarding data sets, we also want to experiment with datasets from different domains, to check whether photo, video, or music content based sites show different system behavior influencing our algorithms.

REFERENCES

[1] P. Melville, R.J. Mooney, R. Nagarajan Content-Boosted Collaborative Filtering for Improved Recommendations, Proceedings of the 18th National Conference on Aritificial Intelligence (AAAI-2002), July 2002, Edmonton, Canada.

[2] M. D. Mulvenna, S. S. Anand, and A. G.. Buchner (2000). Personalization on the Net Using Web Mining. Communications of the ACM, 43(8), 123-125.

[3] D. M. Nichols (1997). Implicit Rating and Filtering. Proceedings of the Fifth Workshop on Filtering and Collaborative Filtering, 31-36.

[4] B. Mobasher, R. Cooley, and J. Srivastava (2000). Automatic Personalization Based on Web Usage Mining. Communications of the ACM, 43(8), 142-151.

[5] D. Riecken (2000). Personalized Views of Personalization. Communications of the ACM, 43(8), 27-28.

[6] Chen, et al. (1993). Some Distributional Properties of Mandarin Chinese – a Study Based on the Academia Sinica Corpus", Proceedings of the First Pacifc Asia Conference on Formal & Computational Linguistics, 81-95.

[7] M. Balabanovic and Y. Shohan (1997). Fab: Content-based, Collaborative Recommendation. Communications of the ACM, 40(3), 66-72.

[8] Ho, et al. (1993). Using Syntactic Markers and Semantic Frame Knowledge Representation in Automated Chinese Text Abstraction. Proceedings of the First Pacific Asia Conference on Formal & Computational Linguistics. 122-131.

[9] R. J. Mooney and L. Roy (2000). Content-based Book Recommending Using Learning for Text Categorization. Proceedings of ACM Conference on Digital Libraries, 195-204.

[10] B. Krulwich, and C. Burkey (1996). Learning User Information Interests Through Extraction of Semantically Significant Phrases. Proceedings of AAAI Spring Symposium on Machine Learning in Information Access. Standford, CA.

[11] M. Kwak and D. S. Cho (2001). Collaborative Filtering With Automatic Rating for Recommendation. Proceedings of ISIE 2001 IEEE International Symposium on Industrial Electronics, 625-628.

[12] J. W. Kwak, and N-I. Cho (2003). Relevance Feedback in the Content-based Image Retrieval System by Selective Region Growing in the Feature Space, Signal Processing-Image Communication, 18(9), 787-799.

[13] K. Lang (1995). Newsweeder: Learning to Filter Netnews. Proceedings of the 12th International Conference on Machine Learning. Tahoe City, CA.

[14] C-H, Lee, Y-H Kim, and P-K Rhee (2001). Web Personalization Expert with Combining Collaborative Filtering and Association Rule Mining Technique. Expert Systems with Applications, 21, 131-137.

[15] C. N. Lee (1999). Understanding the Text Book of Primary School based on How-net. MD. Thesis, Institute of Computer Science and Information Engineering, National Cheng Kung University.

[16] K-L Lee (1999). Intention Extraction and Semantic Matching for Internet FAQ Retrieval. MD. Thesis, Institute of Computer Science and Information Engineering, National Cheng Kung University.

[17] A. McCallum, and K. Nigam (1998). A Comparison of Event Models for Naive Bayes Text Classification. Proceedings AAAI 1998 Workshop on Text Categorization. Adison, WI, 41-48.

[18] N. K. Mimouni, F. Marir, and F. Meziane (2000). An Intelligent Agent for Content-based and Retrieval of Documents. Proceedings of CBIR2000 Conference, Brighton, UK.

[19] I. Schwab, W. Pohl, and I. Koychev (2000). Learning to Recommend From Positive Evidence. Proceedings of Intelligent User Interfaces. ACM Press, 241-247.

[20] F. H. Wang, and H. M. Shao (2004). Effective Personalized Recommendation based on Time-framed Navigation Clustering and Association Mining. Expert Systems with Applications, 27(3), 365-377.

[21] Balabanovic, M. and Y. Shoham. Fab: Content-based, collaborative recommendation. Communications of the ACM, 40(3):66-72, 1997.

[22] Belkin, N. and B. Croft. Information filtering and information retrieval. Communications of the ACM, 35(12):29-37, 1992.

[23] Littlestone, N. and M. Warmuth. The Weighted Majority Algorithm. Information and Computation, 108(2):212-261, 1994.

[24] Pazzani, M. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, pages 393-408, December 1999.

[25] Pazzani, M. and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. Machine Learning, 27:313-331, 1997.

[26] Rocchio, J. J. Relevance Feedback in Information Retrieval. SMART Retrieval System – Experiments in Automatic Document Processing, G. Salton ed., PrenticeHall, Ch. 14, 1971.

[27] Tran, T. and R. Cohen. Hybrid Recommender Systems for Electronic Commerce. In Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press, 2000.

[28] Shardanand, U. and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In Proc. of the Conf. on Human Factors in Computing Systems, 1995.

[29] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In SIAM Conference on Data Mining (SDM), pages 47{58, April 2006.

[30] I. Br, D. Sikl_osi, J. Szabo, and A. A. Benczfur. Linked latent dirichlet allocation in web spam filtering. In AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web, pages 37{40, New York, NY, USA, 2009. ACM.

[31] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993{1022, January 2003.

[32] T. L. Grifiths and M. Steyvers. Finding scientific topics. Proc Natl Acad Sci U S A, 101 Suppl1:5228{5235, April 2004.

[33] M. Porter, "New Models in Probabilistic Information Retrieval," British Library Research and Development Report, no. 5587, 1980.

[34] K. Lang, "Newsweeder: Learning to Filter Netnews," Proc. 12th Int'l Conf. Machine Learning (ICML '95), pp. 331-339, 1995.

[35] Web Usage Mining Using Self Organized Maps, Paola Britos, Damián Martinelli, Hernan Merlino, Ramón García-Martínez IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.6, June 2007.

[36] Chishe Wang; Qi Shen; Linjun Zou. "Research and Application of Web Recommendation System Based on Cluster Mode" (ICEE), 2010 International Conference on 2010 , Page(s): 1445 – 1447

[37] Fong, A.C.M.; Baoyao Zhou; Hui, S.C.; Hong, G.Y. "Web content recommender system based on consumer behavior modeling " IEEE 2011 , Page(s): 962 – 969.