Electronic Programming Guide Recommender for Viewing on a Portable Device

Matthew Y. Ma[1], Jinhong K. Guo[1], Jingbo Zhu[2], Guiran Chang[2]

[1]Panasonic Digital Networking Laboratory
Two Research Way, Princeton, NJ 08540, USA

[2]Institute of Computer Software and Theory, Northeastern University
Shenyang, P.R. China

Abstract. With the merge of DTV and the exponential growth of broadcasting network, an overwhelmingly amount of information have become available at views' homes. Therefore, it becomes increasingly challenging how consumers can receive the right amount of information at the right time for their entertainment needs. We proposed an electronic programming guide (EPG) recommender based on natural language processing techniques. Particularly, the recommender has been implemented as a service on a home network that facilitates the browsing and recommendation of TV programs on a portable remote device and such system is found to be feasible. Preliminary experiments have shown a precision of 81%.

1 Introduction

As the number of channels available on the broadcasting network increases, it becomes more challenging to deal with the overwhelmingly expanding amount of information provided by the electronic programming guide (EPG) and delivering personalized information to the consumer. Consumers can access the EPG via subscription based cable network, the Internet, or services offered by device vendors such as Tivo. However, existing method of multicasting of EPG feeds static contents to users on the same network and do not provide personalized contents. Additionally, EPGs provided through the cable operators are proprietary and do not interface with other data format on the Internet or from other sources. Thirdly, set-top boxes with program suggestion are generally primitive as most systems employ simple category, title, and keyword matching on the EPG contents.

To address such problems, previous work such as Ehrmantraut et. al. [0] and Gena [0] adopted both implicit and explicit feedback for personalized program guide. Takagi et. al. [0] proposed a conceptual matching scheme to be applied to TV program recommendation by fusing of conceptual fuzzy sets and ontology. This work is limited to drama category and the approach is primarily based on program sub-categories of drama as the top layer of the ontological structure to represent user's taste. In recent research, Isobe et. al [0] described a STB based scheme that associates the de-

Y. Ma M., K. Guo J., Zhu J. and Chang G. (2005).
Electronic Programming Guide Recommender for Viewing on a Portable Device.
In *Proceedings of the 1st International Workshop on Web Personalisation, Recommender Systems and Intelligent User Interfaces*, pages 79-88
DOI: 10.5220/0001422800790088

gree of interest of each program with viewer's age, sex, occupation, combined with favorite program categories in sorting the EPG. Yu et. al [0] proposed an agent based system for program personalization under TV Anytime environment [0] using similarity measurement based on VSM. This work, however, assumes that the program information is available on a large storage media and does not address the problem of data sparseness and limited categories supported by most EPG providers. Pigeau et. al. [0] presented a TV recommender system using fuzzy linguistic summarization technique to coupe with both implicit and explicit user profile. This system largely depends on the quality of meta-data and solely on DVB-SI standard [0].

Cotter et. al [0] describes an Internet based personalized TV program guide using an explicit profile and a collaborative approach. Xu et. al [0] also presented some interesting conceptual framework for TV recommendation system based on Internet WAP/SOAP. For portable devices, however, this system inherits the limitations of SOAP/HTTP based technologies, which are considerable network overhead on a portable device.

Our work attempts to address two important perspectives in EPG recommender systems: 1) a home network based framework to support the EPG recommender system for viewing on a portable device; 2) a linguistic based approach to extract from available information source good feature vectors that can be utilized for recommender classifier. Details are discussed in the later sections.

2 Overview

Figure 1 shows the architecture of the EPG recommender system. A portable device communicates with the EPG recommender system via various network protocols, such as infrared, Wi-Fi, WAP or SIP [0]. The EPG recommender consists of program information acquisition module, user profile module, EPG recommendation module, and EPG management and query modules.

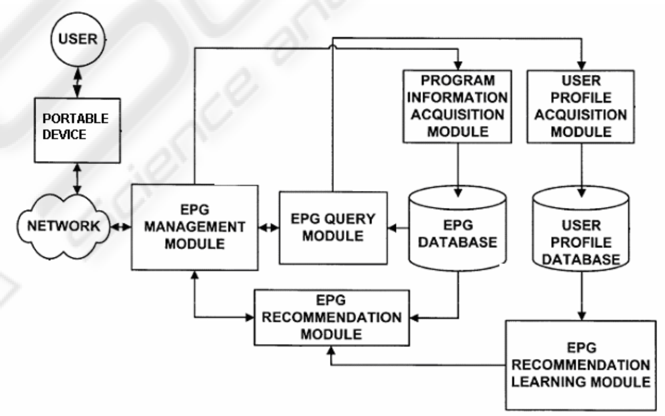

Fig. 1. EPG recommendation system architecture.

The EPG management module is responsible for packing and unpacking data bundles to and from the portable device. The data bundle generally refers to a package that includes application types (such as user requests) and associated data (such as user defined EPG categories for browsing).

Program information acquisition module collects program information from web sites, parses the text data, converts the data into structural data, and stores the structured data in the EPG database. Meanwhile, user profile acquisition module collects user profile data and stores it in the user profile database.

The EPG query module receives and parses the XML data in the bundle to get the content information specified by the user. The query result is packaged in XML format, and delivered to EPG management module in a data bundle. One copy of the query result is delivered to the user profile acquisition module for acquisition of user profile data.

EPG recommendation and learning module dynamically adjusts the parameters of the recommendation algorithm according the user profile. EPG recommendation module recommends programs in the database based on users' preferences.

3 EPG Recommendation System

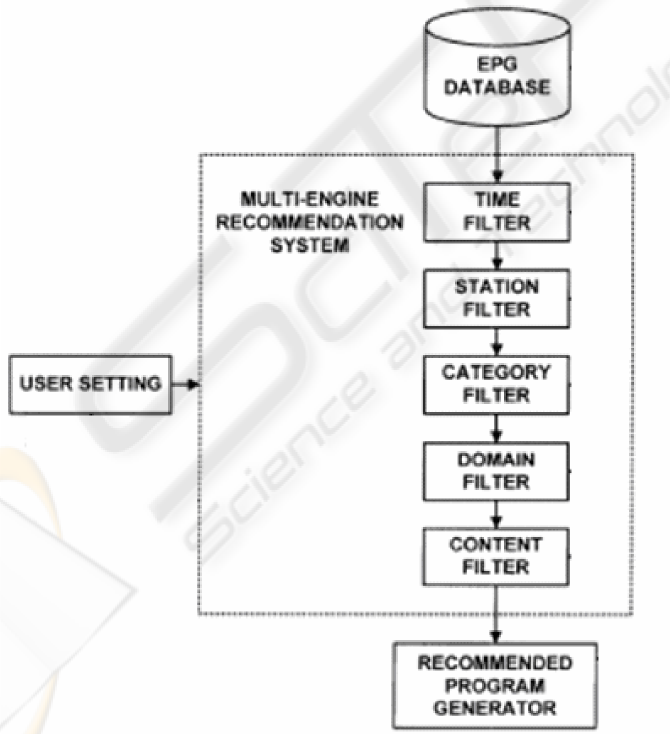

Fig. 2. EPG recommendation.

The EPG recommendation system utilizes the EPG data and user profile to recommend programs. Automatic recommender combined with user preset filters are used to enhance the accuracy of the recommended programs and decrease the search range. The overall architecture is shown in Fig. 2. Five filters: time, station, category, domain, and content filter, are implemented in the recommendation process. The user can predefine a filter setting, for example, a time period from 2004-10-6::0:00 to 2004-10-8::24:00. A default time setting can also be defined, such as the current week. Time filtering can remove all programs that do not play within the specified time period. Station filtering removes the programs that are not on the defined stations from the remaining candidate programs.

Category refers to the genre of the program. Domain information, on the other hand, refers to users' area of interest. Examples of domain information include sports, politics etc. If a user is interested in sports, he may be interested in all the categories that are related to sports, for example, sports news, movies about sports, and documentary about sports.

In both category and domain filter setting, user is provided with three choices: to bypass recommendation; to use automatic recommendation; or to manually select one or more categories/domains. Trained classifiers are used to recommend the program once the user selects to use automatic category or domain recommendation.

Content filter is designed to recommend programs based on the EPG contents. It is more comprehensive as the contents are comprised of all information in an EPG data such as station names, program titles, program descriptions, time interval, and actors. Similarly, in content filter setting, a user can choose to bypass or use automatic content recommendation, which invokes a trained content recommendation classifier.

The recommendation classifiers are further explained and illustrated in Fig. 3. Three classifiers are built for the recommendation via a learning process. Program category data is extracted from user profile database for a particular user by category data extractor. The probability of these extracted categories is computed as:

$$P(c_i) = \frac{N(c_i)}{\sum_{j=1}^{|C|} N(c_j)},$$

where C denotes the set of categories, c_i denotes a category, and $N(c_i)$ denotes the frequency of c_i. Trained category classifier can therefore recommend the programs using the sorted category list in the order of these probabilities.

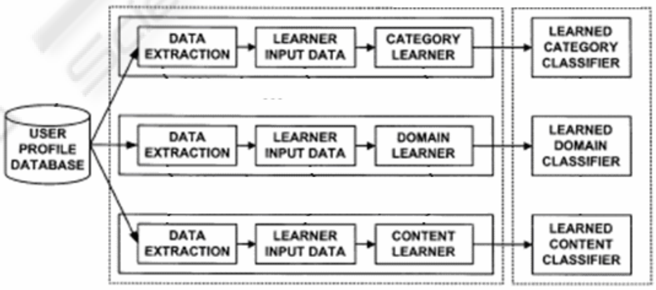

Fig. 3. EPG recommendation learning.

Similarly, at the program domains level, the probability of these extracted domains is computed as:

$$P(d_i) = \frac{N(d_i)}{\sum_{j=1}^{|D|} N(d_j)},$$

where D denotes the set of Domains, d_i denotes a domain, and $N(d_i)$ denotes the frequency of d_i. .

At the program content level, a corpus is constructed that includes preferred and non-preferred programs. The content classifier is trained from the corpus using maximum entropy. The details of maximum entropy classifier will be described in the next section.

After the filtering process, recommended program generator places the recommended programs into a human readable format, e.g. XML format. The formatted program information are packaged in a data bundle and sent to the portable device for presentation according to the user's predefined style sheet.

4 Maximum Entropy Classifier

Maximum entropy classifier has been employed in two processes. Since domain information is not readily available from the EPG data, maximum entropy technique is used for text classification. Domain information is classified from EPG data via a maximum entropy text classifier that is trained from a corpus. In the second process, maximum entropy model is used to obtain the content classifier for recommendation as shown in Fig. 3. Such maximum entropy model is obtained from a trained EPG database with integrated user profile.

A. Domain Information and Text Classification

We utilize detailed program information (abstract or description) in EPG to further extract characteristics of programs, particularly the domain information. Program information data can be obtained either directly from the service providers or from Internet professional websites, such as TV Guide [0] and TitanTV [0]. This information forms the basis of the EPG database and is in a semi-structural text format such as HTML and/or XML.

For text classification, a training corpus is collected by tagging a collection of programs into predefined domains. Fig. 4 shows the classification process. First, program vectors that construct the vocabulary are formed by using the bag-of-words model. Because the count matrix is high dimensional in the feature space due to the complexity of high dimensional text data, feature selection is performed to lower the feature space. When constructing vocabulary, stop words are removed from the list in the training corpus.

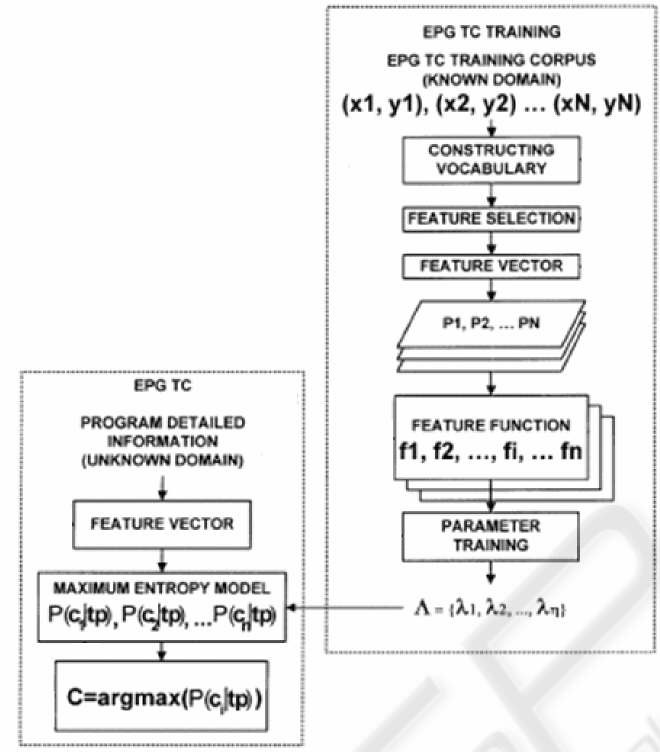

Fig. 4. Classification from detailed program information.

The $\chi 2$ statistic measures the lack of independence between a word ***t*** and a domain ***c***. Using the two-way contingency table of a word ***t*** and a domain ***c***, where ***A*** is the number of times ***t*** and ***c*** co-occur, ***B*** is the number of time the ***t*** occurs without ***c***, ***C*** is the number of times ***c*** occurs without ***t***, ***D*** is the number of times neither ***c*** nor ***t*** occurs, and ***N*** is the total number of documents, the term "goodness measure" is defined to be:

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)}$$

The $\chi 2$ statistic is zero if ***t*** and ***c*** are independent. For each domain, the $\chi 2$ statistic can be computed between each entity in a training sample and that domain to extract the features.

The programs can be represented as a vector of features and the frequency of the occurrence of that feature in the form of *P* = <*tf1, tf2, …, tfi,… , tfn* >, where ***n*** denotes the size of features set, and ***tfi*** is the frequency of the i^{th} feature.

Maximum entropy (ME) model is a general-purpose machine-learning framework that has been successfully applied to a wide range of text processing tasks [0][0]. Given a set of training samples $T=\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ where x_i is a real value feature vector and y_i is the target domain, the maximum entropy principle states that data ***T*** should be summarized with a model that is maximally noncommittal with respect to missing information. Among distributions consistent with the constraints imposed by ***T***, there exists a unique model with highest entropy in the domain of exponential models of the form:

$$P_\Lambda(y \mid x) = \frac{1}{Z_\Lambda(x)} \exp\left[\sum_{i=1}^{n} \lambda_i f_i(x, y)\right] \qquad (1)$$

where $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ are parameters of the model, $f_i(x, y)$'s are arbitrary feature functions of the model, and $Z_\Lambda(x) = \sum_y \exp[\sum_{i=1}^{n} \lambda_i f_i(x, y)]$ is the normalization factor to ensure $P_\Lambda(y \mid x)$ is a probability distribution. Furthermore, it has been shown that the maximum entropy model is also the Maximum Likelihood solution on the training data that minimizes the Kullback-Leibler divergence between P_Λ and the uniform model. Since the log-likelihood of $P_\Lambda(y \mid x)$ on training data is concave in the model's parameter space Λ, a unique maximum entropy solution is guaranteed and can be found by maximizing the log-likelihood function:

$$L_\Lambda = \sum_{x,y} \tilde{p}(x, y) \log p(y \mid x)$$

where $\tilde{p}(x, y)$ is an empirical probability distribution. Our current implementation uses the Limited-Memory Variable Metric method, called L-BFGS, to find Λ. Applying L-BFGS requires evaluating the gradient of the object function L in each iteration, which can be computed as:

$$\frac{\partial L}{\partial \lambda_i} = E_{\tilde{p}} f_i - E_p f_i$$

where $E_{\tilde{p}} f_i$ and $E_p f_i$ denote the expectation of $\boldsymbol{f_i}$ under empirical distribution $\tilde{p}$ and model p respectively.

The feature function in our algorithm is defined as the following:

$$f_{w,c'}(d, c) = \begin{cases} 0 & c \neq c' \\ n(w, d) & c = c' \end{cases} \qquad (2)$$

where, ***n(w,d)*** denotes the frequency of the word ***w*** in program ***d***.

The training programs are represented as follows:

TP: $tp_1, tp_2, \ldots, tp_i, \ldots, tp_n \rightarrow T = (V, C): (v_1, c_1), (v_2, c_2), \ldots, (v_i, c_i), \ldots, (v_n, c_n)$

where *TP* denotes training programs set, tp_i denotes training program *i*, *V* denotes the vectors, and *C* denotes the domains. The feature function set *F* can be constructed using Equation (2) and the parameters $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ of the ME model are estimated using the feature function set *F* and the training samples *(V, C)*. Using Equation (1), $P(c_1|tp), P(c_2|tp), \ldots, P(c_i|tp), \ldots, P(c_n|tp)$ for each domain can be computed. Finally, the domain

$$c: c = argmax(P(c_i|tp))$$

is selected.

B. Content Classifier for Recommendation

Unlike some existing systems that prompt user to provide keywords to establish a user profile, we utilize explicit feedback system that allows mobile users to indicate their preferences relating to each program information viewed. The user preference is later integrated into the EPG database. The EPG recommendation process is also utilizing maximum entropy model and works in a similar way as shown in Fig. 4.

In EPG content recommendation, upon user's choice of preference on each program, several features were extracted from the raw EPG database. These features are divided into several groups. 1) Station-Name Feature: The corresponding value for the selected station is 1. 2) Time Feature: time the program is played. We divide a day into 24 intervals. 3) Lexicon Feature: Title, Episode Title, and Program Information. First, we construct a vocabulary using these three fields in training data. The string of the token *w*, which is included in the vocabulary, is used as a feature. 4) Category Feature: This information is usually contained in EPG data from content providers. 5) Actors Feature.

As shown in Fig. 4, feature functions are obtained from feature vectors. EPG recommendation and learning module dynamically adjusts the parameters of the recommendation algorithm according to the user profile by calculating the maximum entropy model Λ. The calculation of Λ parameters requires the use of feature vectors and training corpus, which consists of raw EPG database and added user profile. In an extreme case, if user is only interested in one domain, the recommendation classifier would be a binary classifier that only outputs "like" or "dislike" for all input program content.

5 Prototype and Experiments

In our experiments, EPG recommender was implemented on a small corpus, about one month's EPG for 30 channels, resulting in 1Mbytes of EPG data. In addition, we built a prototype framework to enable the downloading of EPG from home network and viewing on a portable device. The EPG collection and recommendation system is implemented on a home network, where EPG algorithm is running on a home server that supports OSGi[0] framework. The OSGi (Open Service Gateway Initiative) framework provides an open execution environment for applications to run on heterogeneous devices, particularly, it provides flexibility for content providers to upload updates to consumers' devices. The portable device is a mobile device that supports SIP[0], which allows simple text based messages to be carried between the mobile device and the home network devices. Additionally, it provides streaming support for our future extension.

The prototype also enables a mobile client with three functions - EPG browsing (by date, channel etc.), Program Details (for specific program) and EPG recommendation. Fig.5 shows a mobile user interface for (a) EPG program details and (b) a recommended program list. As shown at the bottom of Fig.5(a), a "like" and "dislike" button is provided so user can give some relevance feedback to the recommendation module after reviewing the program details.

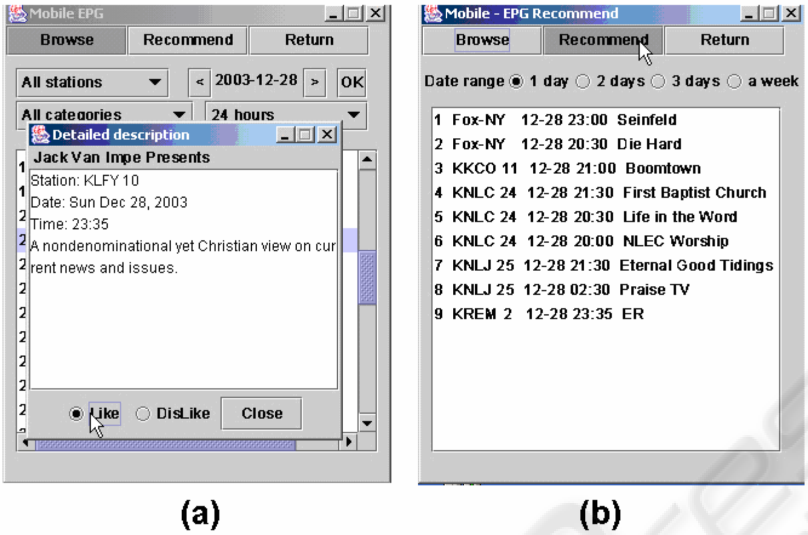

Fig.5. (a) EPG program details and **(b)** recommended program list on a mobile device.

We have conducted a preliminary experiment and used only program information in the training. Among the four weeks of EPG data, one week is used for generating user's profile data as training corpus, whereas the other three weeks are used for testing. The training corpus is collected when a user provides relevance feedback to the training EPG. In our experiment, user is only concerned with sports domain in both training and recommendation. The recommendation on the other three weeks EPG data is judged by the same user. A precision rate of 81% was achieved. Because each recommendation for a specific domain is likely independent from that of another domain, we can expect similar performance once we expand the recommendation engine to multiple domains in the future.

6 Conclusion

Among home entertainment services, electronic programming guide (EPG) is perhaps the most appealing applications for television, and its services continue to grow in the emergence of new digital TV market. Our proposed system features EPG collection from non-proprietary data sources (i.e. HTML on the Internet) and an EPG recommender based on text classification and maximum entropy model. As we are aware, the proposed work is the first of its kind using natural language processing techniques for TV recommender and the result is promising. A relevance feedback is also implemented to provide dynamic personalized EPG service. The prototype of EPG recommender is implemented under OSGi environment and the viewing of EPG on a portable device is enabled through SIP network.

The presented work and prototype have suggested a feasible architecture and technology for providing personalized home network based EPG service. Our next step is to systematically collect EPG training corpus and also conduct text classification and EPG recommender evaluation. In addition, how relevance feedback can be best provided through user's daily TV viewing experience implicitly on the portable device or on a home server would be a challenge. Third, there is a future need to address browsing/sending graphics and streaming in EPG information via the home network.

References

1. TV Guide. http://www.tv-guide.com
2. TV Anytime Forum. http://www.tv-anytime.org
3. TitanTV Guide. http://www.titantv.com
4. Specification for Service Information (SI) in DVB Systems, DVB Document A038 Rev. 1, May 2000.
5. OSGi: Open Services Gateway Initiative. http://www.osgi.org
6. SIP: Session Initiation Protocol. http://ietf.org/html.charters/sip-charter.html.
7. T. Isobe, M. Fujiwara, H. Kaneta, U. Noriyoshi and T. Morita, Development and features of a TV navigation system, IEEE Transactions on Consumer Electronics, Vol. 49, Issue 4, Nov. 2003, pp. 1035-1042.
8. T. Takagi, S. Kasuya, M. Mukaidono and T. Yamaguchi, Conceptual matching and its applications to selection of TV programs and BGMs, IEEE SysInt. Conf. On Systems, Man and Cybernetics, Vol. 3, Oct, 1999, pp. 269-273.
9. J. Xu, L. Zhang, H. Lu and Y. Li, The development and prospect of personalized TV program recommendation systems, Proceedings of the IEEE 4th Int. Symposium on Multimedia Software Engineering (MSE), 2002.
10. A. Pigeau, G. Raschia, M. Gelgon, N. Mouaddib, R. Saint-Paul, A fuzzy linguistic summarization technique for TV recommender systems, Proceeding of the IEEE Int. Conf. On Fuzzy Systems, 2003, pp. 743-748.
11. Z. Yu, X. Zhou, X. Shi, J. Gu and A. Morel, Design, implementation, and evaluation of an agent-based adaptive program personalization system, Proceedings of the IEEE 5th Int. Symposium on Multimedia Software Engineering (MSE), 2003.
12. A. Berger, S. Della Pietra , V. Della Pietra, A maximum entropy approach to natural language processing, Computational Linguistics, 1996, 22(1): 58-59
13. C. Gena, Designing TV viewer stereotypes for an Electronic Program Guide, Proceedings of the 8th International Conference on User Modeling, 2001, 3:274-276
14. M. Ehrmantraut, T. Herder, H. Wittig and R. Steinmetz, The personal Electronic Program Guide- towards the pre-selection of individual TV programs, In Proc. Of CIKM'96, 1996, pp.243-250
15. P. Cotter and B. Smyth, PTV: Intelligent Personalised TV Guides. Proceedings of the 12th Innovative Applications of Artificial Intelligence (IAAI) Conference, 2000.
16. L. Zhang, J. Zhu and T. Yao, An Evaluation of Statistical Spam Filtering Techniques, ACM Transactions on Asian Language Information Processing (TALIP), Vol. 3, No.4, pages 243-269, December 2004.