International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 21, No. 2 (2007) 375–395

CONTENT CLASSIFICATION AND RECOMMENDATION TECHNIQUES FOR VIEWING ELECTRONIC PROGRAMMING GUIDE ON A PORTABLE DEVICE

JINGBO ZHU

Institute of Computer Software and Theory, Northeastern University
Shenyang, P.R. China
zhujingbo@mail.neu.edu.cn

MATTHEW Y. MA

IPVALUE Management Inc., USA
mattma@ieee.org

JINHONG K. GUO

Panasonic Princeton Laboratory, USA
kguo@research.panasonic.com

ZHENXING WANG

Institute of Computer Software and Theory, Northeastern University
Shenyang, P.R. China
wzhx1983@gmail.com

With the merge of digital television (DTV) and the exponential growth of broadcasting network, an overwhelmingly amount of information has been made available to a consumer's home. Therefore, how to provide consumers with the right amount of information becomes a challenging problem. In this paper, we propose an electronic programming guide (EPG) recommender based on natural language processing techniques, more specifically, text classification. This recommender has been implemented as a service on a home network that facilitates the personalized browsing and recommendation of TV programs on a portable remote device. Evaluations of our Maximum Entropy text classifier were performed on multiple categories of TV programs, and a near 80% retrieval rate is achieved using a small set of training data.

Keywords: Electronic programming guide; EPG; recommendation; content classification; domain of interest; OSGi; portable device.

1. Introduction

With the advent and widespread use of mobile phones and more recently the launch of 3G services, delivery of entertainment media and metadata such as electronic programming guide (EPG) has become more desirable. Examples of these content

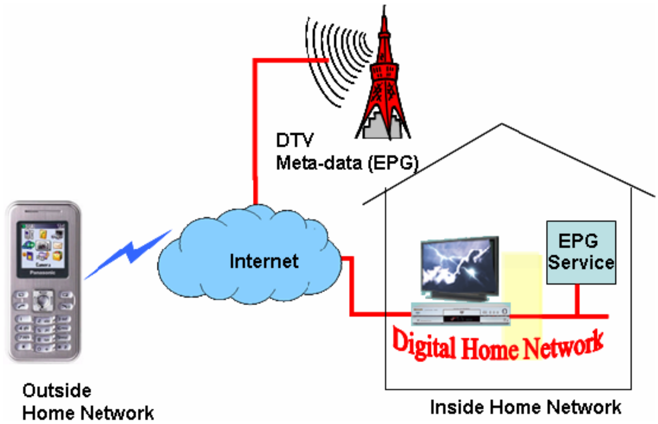


Fig. 1. EPG delivery system for a portable device.

delivery services on a mobile network include the new DVB-H standard and the recent trials of mobile-TV service in US and Europe. However, existing technologies and infrastructures all have their shortcomings, particularly the subscription fees associated with these add-on services to the mobile network. Furthermore, only simple search/browsing functions are available due to limited computing power of the mobile phone. As the number of channels available on the TV broadcasting network increases, it becomes more challenging to deal with the overwhelmingly expanding amount of information provided by the EPG. In our view, the realization of delivering broadcast EPG information to a mobile phone must integrate a personalized EPG recommendation service that helps alleviate the burden of storage and network bandwidth requirement posted on the mobile phone. This EPG recommendation service should also leverage (a) the available content from the Internet, for which consumers have already paid premium for broadband services; (b) existing higher computing power peripherals such as digital video recorder (DVR) or media server at home; (c) open architecture and standard protocols for delivering EPG data to a mobile device. Based on these considerations, we propose an EPG delivery system for a portable device, as shown in Fig. 1.

In our proposed system, the EPG content comes from the Internet to leverage the existing broadband infrastructure at home. The delivery of EPG to a portable device is based on the ongoing open platform OSGi[19] and SIP.[22] The collection, processing and recommendation of EPG content are realized through an EPG Service on a home media server, which is connected to the home network. The EPG service can be realized as a software bundle that is advertized on the home network. The EPG service provides recommended EPG list to a mobile phone, as well as receives user's feedback in order to provide personalized service. The communication between EPG Service and mobile phone can be carried via SIP protocol. Details of networking architecture are described in our prior work.[5,16]

Whereas home and mobile networking are enabling technologies for the proposed EPG delivery system, the focus of this paper is on the EPG recommender system. In a home and mobile environment as illustrated in Fig. 1, the EPG recommender system is a crucial technology required for reducing the amount of data to be delivered to a portable device. In our research, we employ a content based recommendation engine using a statistical approach, which overcomes the disadvantages associated with traditional keyword search or keyword matching. Furthermore, we introduce the concept of "domain of interest", which truly reflects the program of interest from users' perspectives across multiple genres. The recommender based on the "domain of interest" yields higher recommendation rate than that without it.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 introduces our proposed EPG recommender system. Our core contribution in domain identification and content recommendation are described in Sec. 4, followed by prototypes and experiments in Sec. 5. Finally, we conclude this paper in Sec. 6.

2. Related Work

Unlike the delivering of media content to portable devices, fewer efforts have been made on EPG recommendation for viewing on a portable device. In this section, we review some of prior research in the general area of EPG recommendation.

Various recommender approaches have been proposed based on the EPG content, particularly category information. Ehrmantraut *et al.*[8] and Gena[9] adopted both implicit and explicit feedback for personalized program guide. Takagi *et al.*[24] 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 subcategories of drama as the top layer of the ontological structure to represent a user's taste. In a more recent approach, Blanco *et al.*[4] used TV ontology to build a collaborative recommendation system by defining similarity measures to quantify the similarity between a user's profile and the target programs. However, how to map the target program to the predefined categories is still a crucial problem, and in so-called TV ontology there is no acceptable current standard for the categories of TV programs.

Isobe *et al.*[11] described a set-top box based scheme that associates the degree of interest of each program with viewer's age, gender, occupation, combined with favorite program categories in sorting EPG content. Yu *et al.*[29] proposed an agent based system for program personalization under TV Anytime environment[25] using similarity measurement based on VSM (Vector Space Model). 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.*[20] presented a TV recommender system using fuzzy linguistic summarization technique to cope with both implicit and explicit user profiles. This system largely depends on the quality of meta-data and solely on DVB-SI standard.[23]

Cotter *et al.*[6] described an Internet based personalized TV program guide using an explicit profile and a collaborative approach. Xu *et al.*[27] 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 add considerable network overhead on a portable device. Hatano *et al.*[10] proposed a content searching technique based on user metadata for EPG recommendation and filtering, in which four types of metadata are considered. The user attributes such as age, gender and place of residence are considered to implement an adaptive metadata retrieval technique. However, these attributes are too general for EPG recommendation. Personalized profiles for EPG recommendation mainly depend on users' interest or characteristics. Even though age and gender play certain roles, they are not the deciding factors.

Ardissono *et al.*[1,2] implemented personalized recommendation of programs by relying on the integration of heterogeneous user modeling techniques. However, users cannot often declare their preferences in a precise way.

Some recommender systems attempted to integrate multiple prediction methods, such as neural network, given the user's reactions to the system's recommendations. One approach to merging multiple methods is to use relevance feedback technique, which can benefit from an informed tuning parameter. For example, Shinjo *et al.*[21] implemented an intelligent user interface based on multimodal dialog control for audio-visual systems, in which a TV program recommender based on viewing-history analysis is included. The profiler is built using groups of keywords analyzed from EPG content.

Our work presented in this paper attempts to address two important perspectives in an EPG recommender systems: (1) a home network based framework to support the EPG recommender system for viewing on a portable device with a vision that the system is to be deployed in an embedded entertainment device in the future and targeted on a portable device (such as PDA or mobile phone) for TV program viewing; (2) a linguistic based approach to extract good feature vectors from available information to be used in a recommender classifier.

Whereas commonly used keyword based recommendation technique inherits certain limitations, we intend to focus our research on content based recommender system. The reasons are two-fold: one is that content based recommendation system provides a flexible framework that allows the integration of richer information that may truly reflect users' preferred content; secondly, the EPG content available for the TV is abundant so that typical sparseness concerns associated with any content filtering and machine learning techniques are diminishing. Within the same paradigm, we also introduce a concept of "domain of interest" across multiple genres. The details of our system will be described in the following sections.

3. Proposed System

3.1. *Overview*

Figure 2 shows the architecture of our EPG recommender system. A portable device communicates with the EPG recommender system via the SIP network.[22] The

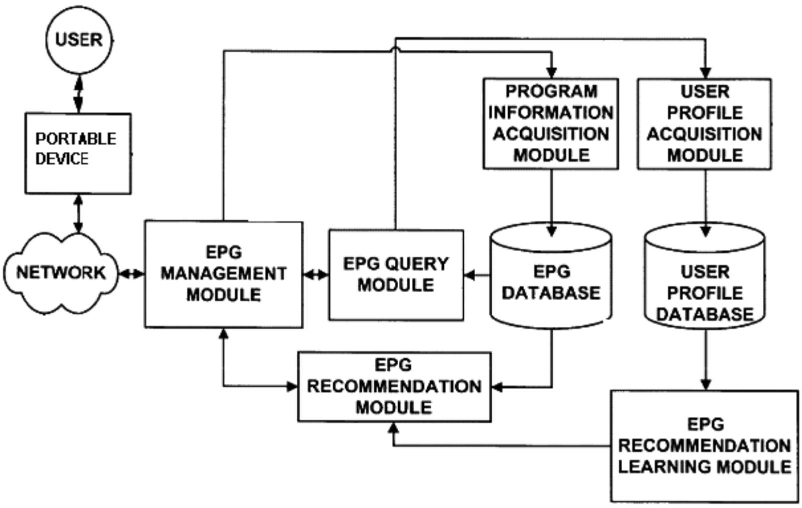


Fig. 2. EPG recommendation system architecture.

EPG recommender consists of program information acquisition module, user profile module, EPG recommendation module, and EPG management and query modules.

The EPG management module is responsible for sending EPG data to and receiving user request and feedback from the portable device. Program information acquisition module collects program information from the Internet, parses text data, converts the data into a structural form usable for our recommender. 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 the XML format, and delivered to the 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. The user profile data primarily comprises of user's preference associated with each program. Examples of user profile data include the duration that each program is being watched (by the user), user's relevance feedback e.g. "like" or "don't care" etc.

The EPG recommendation learning module dynamically adjusts and optimizes the parameters of the recommendation algorithm according to the user profile data. The EPG recommendation module recommends programs in the database based on users' preferences. Our focus is on the recommendation module and associated learning module which will be described in detail.

3.2. *EPG recommender system*

Our general multi-engine EPG recommender system, as shown in Fig. 3, uses a series of filters to enhance the accuracy of recommendation and narrows down the search range. Five filters: time, station, category, domain, and content filter, are implemented in the recommendation process. A 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 works in a similar way.

Category and domain filters can be executed automatically without requiring users to preselect a criterion. Category refers to the genre of the program and it is normally available from the EPG data. Domain information more closely reflects a user's domain of interest and is broader. Domain information may cover programs across multiple categories. For example, "sports" domain includes any programs concerning sports, regardless it is a "movie", "news" or "sports" program. A trained probabilistic model is built for category filtering or domain filtering, such that the probability of a program being recommended is computed as:

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

where C denotes the set of categories/domains, c_i denotes a category/domain, and $N(c_i)$ denotes the frequency of c_i.

Since domain information is not available in provided EPG data, it can be obtained via text classification of EPG content based on the Maximum Entropy model in our system.

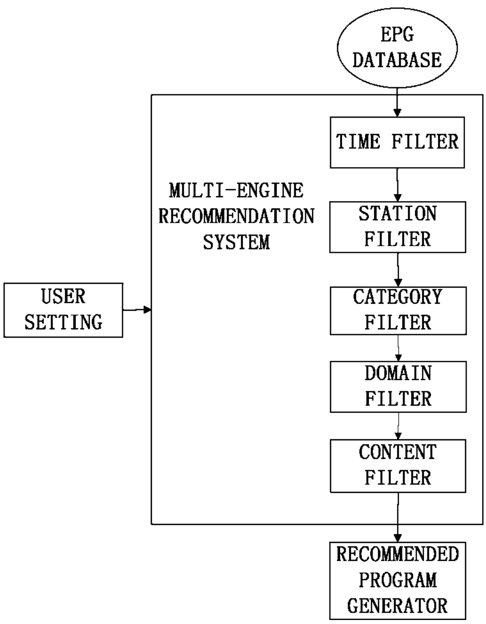


Fig. 3. EPG recommendation.

The content filter is designed to recommend programs based on the EPG content. It is more comprehensive as the EPG content being used in the recommendation comprises of all information in an EPG data such as station names, program titles, program descriptions, time interval, and actors. At program content level, a corpus is constructed such that it includes preferred and non-preferred programs. A binary content classifier is built using the Maximum Entropy model.

After filtering, the 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 display according to the user's predefined style sheet.

4. Domain Identification and Content Recommendation

4.1. *Classification problem and design choice*

A typical EPG entry has several attributes: *program title, time, channel/station, program information, duration, rating,* and *category.* The following is a sample downloaded from TV Guide[26]

Program title: *Bend It Like Beckham*
Time: *Oct 03 09:00pm*
Channel/Station: *IFC 550*
Program Info: *An 18-year-old plays for a women's soccer team but conceals it from her parents.*
Duration: $2:00$
Rating: *PG-13*
Category: *Movie*
***Domain*:** Sports (need to be calculated)

As shown above, we added a new field to the downloaded EPG: Domain (of interest). As previously described, domain may reflect more of a user's interest. To perform domain filtering, domain needs to be assigned to each TV program.

In our proposed system, the classification problem is visited twice. First, program content recommendation can be formularized as a binary text classification. In other words, the task of binary classification is to automatically assign "like" or "don't care" labels to each program based on its EPG content. Secondly, the domain of interest needs to be identified for each TV program. This is a nonbinary text classification problem, in which each TV program is classified as one of the domains of interest based on its EPG content.

The text classification (TC) is a common natural language processing technique. To choose a text classifier, we considered several techniques. When provided with enough training data, a variety of techniques for supervised learning algorithms have demonstrated satisfactory performance for classification task, such as Rocchio,[12,15] SVM,[13] decision tree,[14] Maximum Entropy[18] and naive Bayes models.[17] In using these models, EPG text content can be represented as a high-dimensional vector using bag-of-words model as input for training and testing processes. All the items in the vector are treated equally. Even though feature selection is used to reduce the dimension of a vector, some salient features from diverse sources, such as TIME, STATION, ACTOR and DOMAIN, can still lose their significance in the large-dimension feature vector.

Maximum Entropy (ME) model[3] is a general-purpose machine-learning framework that has been successfully applied to a wide range of text processing tasks, such as statistical language modeling, language ambiguity resolution, and text categorization. The advantage of ME model is its ability to incorporate features from diverse sources into a single, well-balanced statistical model. In many classification tasks such as text classification[18] and spam filtering,[30] ME model often outperforms some of the other classifiers, such as naive Bayes and KNN classifier. SVM also has been performing well in many classification tasks, however, due to its high training cost, SVM is not a suitable choice for adaptive recommendation on the run. Our goal is to design a recommendation system, which dynamically updates itself as user profile is being updated. ME model is thus selected in both EPG domain identification and content recommendation.

4.2. *Maximum entropy model*

The domain information is identified by our system based on the original EPG content. Figure 4 shows a diagram of such process. Program vectors that construct the vocabulary are formed using the bag-of-words model. However, the dimension of the count matrix is very high in the feature space due to the complexity of high-dimensional text data. Therefore, feature selection is performed to lower the feature space. This step is also crucial in adapting traditional ME classifier to a mobile environment. When constructing the vocabulary, stop words are removed from the list in the training corpus.

Using bag-of-words model, a classified program can be represented as a vector of features and the frequency of the occurrence of that feature is in the form of $P = \langle tf_1, tf_2, \ldots, tf_i, \ldots, tf_n \rangle$, where $\mathbf{n}$ denotes the size of feature set, and tf_i is the frequency of the ith feature. Given a set of training samples $\mathbf{T} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_\mathbf{N}, \mathbf{y}_\mathbf{N})\}$ where $\mathbf{x_i}$ is a real value feature vector and $\mathbf{y_i}$ is the target domain, the maximum entropy principle states that data $\mathbf{T}$ should be summarized with a model that is maximally noncommittal with respect to the missing information. Among distributions consistent with the constraints imposed by $\mathbf{T}$, there exists a unique model with the highest entropy in the domain of exponential models of the form:

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

where $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ 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 $P_\Lambda(y|x)$ is a probability distribution. Furthermore, it has been shown that the ME model is also the Maximum Likelihood solution on the

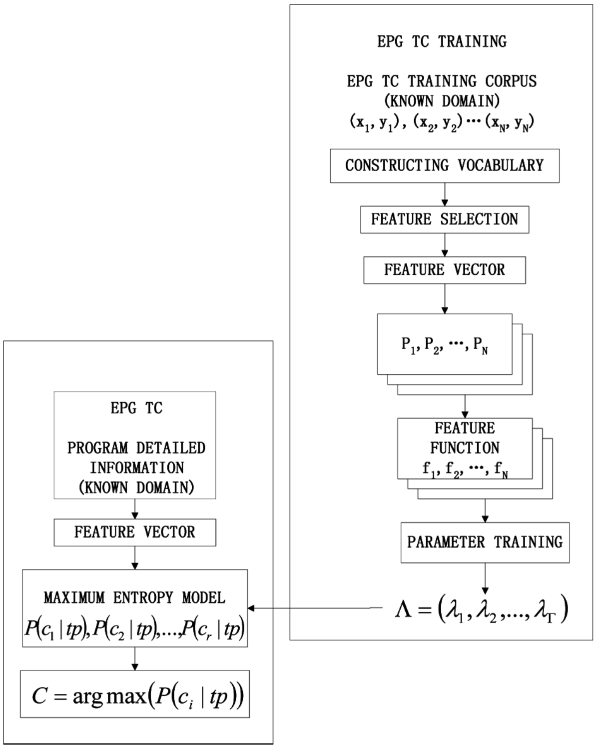


Fig. 4. ME-based classification process.

training data that minimizes the Kullback–Leibler divergence between P_Λ and the uniform model. Since the log-likelihood of $P_\Lambda(y|x)$ 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:

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

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 f_i under empirical distribution $\tilde{p}$ and model p, respectively.

4.3. *Feature dimension reduction*

Feature dimension reduction (also referred to as feature pruning or feature selection) is employed to reduce the size of the feature space to an acceptable level, typically several orders of magnitude smaller than the original. The benefit of dimension reduction also includes a small improvement in prediction accuracy in some cases.[28]

Two approaches, feature selection and feature extraction can be employed for this purpose. Feature selection refers to algorithms that output a subset of the input feature sets. Feature extraction creates new features based on transformations or combinations of the original feature set. Instead of using all the available features in the observation vectors, the features are selected based on some criteria of removing noninformative terms according to corpus statistics, such as document frequency, $\chi 2$ statistic, information gain, term strength and mutual information methods.[28]

The $\chi 2$ statistic is one of the best performing scoring functions for feature selection in text content classification. 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 times 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 training samples, the term "goodness measure" is defined to be:

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

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. In our content recommendation system, feature selection is done by selecting words that have the highest $\chi 2$ statistic of the class variable.

4.4. *Domain identification*

Domain information can be obtained from a corpus of EPG data through a training process. Domain classification, formulated as a nonbinary text classification problem, is performed using Maximum Entropy classifier.

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

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

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

The training programs are represented as the following: $TP: tp_1, tp_2, \ldots, tp_i, \ldots, tp_n \longrightarrow T = (V,C) : (v_1,c_1),(v_2,c_2),\ldots,(v_i,c_i),\ldots,(v_n,c_n)$, where TP denotes the training program set, in which all programs are labeled as corresponding domain information. tp_i denotes training program i, V denotes the vectors, and C denotes the domains.

The feature function set F can be constructed using Eq. (2) and the parameters $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ of the ME model are estimated using the feature function set F and the training samples (V,C). Using Eq. (1), given a test program $tp, P(c_1|tp), P(c_2|tp), \ldots, P(c_i|tp), \ldots, P(c_n|tp)$ for each domain can be computed. The domain $c : c = \mathrm{argmax}(P(c_i|tp))$ is assigned to the test program tp.

4.5. *Content classifier for recommendation*

Unlike some existing systems that need users to provide keywords to establish a user profile, we utilize an explicit user feedback model. In this model, each choice by the user to indicate their preference: "like" or "don't care" is fedback into the learning module. The EPG recommendation process is also based on maximum entropy model and works in a similar way as shown in Fig. 4.

In EPG content recommendation, several features were extracted from the raw EPG database based on users' preference on each program. These features are divided into the following groups.

(1) Station-Name: The corresponding value for the selected station is 1.
(2) Time: Time interval the program is played. We divide a day into 24 intervals.
(3) Lexicon: Title, Episode Title, and Program Information. We construct a vocabulary using these three fields in training data. The string of the token w in Eq. (2), which is included in the vocabulary, is used as a feature.
(4) Category Feature: This is usually included in EPG data.
(5) Actors.

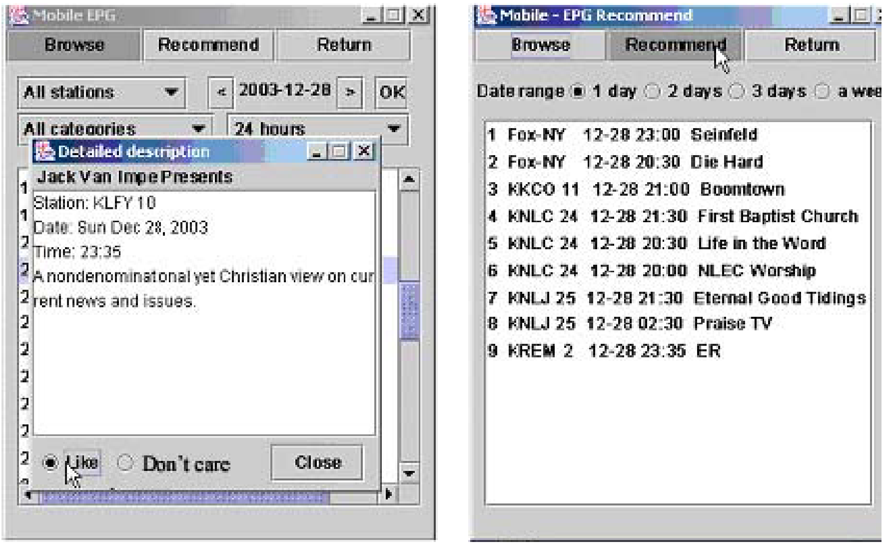
As shown in Fig. 4, feature functions are obtained from feature vectors. The calculation of the ME model Λ parameters requires the use of feature vectors and training corpus, which consists of raw EPG database and user profile. In extreme cases, if the user is only interested in one domain, the recommendation classifier would be a binary classifier that only outputs "like" or "don't care" for all program content.

5. Prototype and Experiments

5.1. *Prototype*

We built a prototype framework to enable the downloading of EPG from the Internet 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[19] 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 Sharp Zaurus PDA with installed SIP[22] support, which allows simple text based messages to be carried between the mobile device and the home network devices.

The prototype also enables a mobile client with three functions — EPG browsing (by date, channel etc.), Program Details (for specific program) and EPG recommendation. Figure 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 "don't care" button is provided so user can give some relevance feedback to the recommendation module after reviewing the program details.



(a) (b)

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

5.2. *Experimental database and protocol*

In our experiments, we downloaded two-weeks of EPG for 30 channels, resulting in 1Mbytes of EPG data. The training data contains 21,277 TV programs collected from DirecTV[25] between August 8 and August 12, 2005. This data is used for generating user profile data as the training corpus. The testing was performed on 7394 TV programs collected between August 13 and August 20, 2005. Each EPG data entry includes id, time, title, a brief synopsis, duration, rating and category. Most common categories contained in the EPG are series, movie, shopping, sports, special and news. The first experiment was designed to test the effectiveness of the ME classifier. The second experiment was designed to evaluate the performance of the recommender system. To do this, we divided the TV programs into groups using the category information provided by the EPG. The result of user recommendation is evaluated for each category. The corpus is collected when a user provides relevance feedback to the system upon receiving the training EPG. The testing EPG data are tagged by the same user and the recommendation result is compared with the tag.

Table 1 shows the training data and testing data sizes of the various categories of programs.

5.3. *Evaluation of ME classifier for domain identification*

To test the effectiveness of the text classification for domain information, we utilized the category information included in the EPG data to avoid the need to manually tag the entire database. This is because category information (tag) is

Table 1. Experimental data size.

	Total Number of Programs	Training Data	Testing Data
Series	15808	11746	4062
Movie	5248	3883	1365
shopping	2576	1911	665
Sports	1807	1347	460
Special	1653	1217	436
News	1579	1173	406

readily available in EPG data whereas domain data is not. We can argue that the measuring of text classification on category can be a good indicator on the text classification performance on domain information using the same ME classifier.

For EPG training corpus, we use various fields to obtain the features of our ME model. For testing data, we remove the category information and let the ME classifier decide the category, which is then compared with the ground truth (i.e. removed category label). A quantitative measurement for performance is defined as following.

$$F1 = \frac{\text{precision} \times \text{recall} \times 2}{\text{precision} + \text{recall}}.$$

The overall F1 value is calculated as

$$\frac{\frac{1}{n}\sum_{j=1}^{n} P_j \bullet \frac{1}{n}\sum_{j=1}^{n} R_j \bullet 2}{\frac{1}{n}\sum_{j=1}^{n} P_j + \frac{1}{n}\sum_{j=1}^{n} R_j}$$

where P_j is the precision value of the jth category and R_j is the recall value of the jth category; n is the number of categories.

For our experiment, the training data includes 29,800 different words. Table 2 shows the experimental result. In this experiment, we use all EPG information as features (excluding ID and, of course, category). The F1 value is shown in Table 2.

As in Table 2, the series, movie and shopping categories yield better results. This is because there are more programs available for series, movie and shopping. Additionally, the synopsis/comments have more details for these programs. The sports and news programs, however, do not have detailed comments; some of such programs have no comments at all. The programs in special category are something

Table 2. Evaluation of ME classifier using all content.

	Precision	Recall	F1
Series	0.8680	0.9169	0.8918
Movie	0.9394	0.9137	0.9264
Shopping	0.8706	0.8283	0.8490
Sports	0.7313	0.8014	0.7647
Special	0.6968	0.5745	0.6297
News	0.8432	0.6104	0.7082
Overall F1	0.8249	0.7742	0.7987

difficult to classify. These programs do not fit well into all the other categories. Meanwhile, they are much diversified among themselves.

Overall, the ME classifier shows close to 0.8 F1 in the category identification, with over 0.9 on movie, followed by 0.89 F1 and 0.85 F1 for series and shopping, respectively. Although the actual performance on domain identification needs to be verified, this experiment gives promising results on the performance of our proposed ME classifier.

5.4. *Evaluation of content recommendation*

To conduct the experiment for content recommendation, the training and testing data were tagged by different users (four groups) as "like" and "don't care" manually. We did not use the full set of training data. Instead, we conducted experiments using training set of 100, 200, 300, 400 and 500 programs to show the effect of training data size on the recommendation result. The training data are selected randomly from the five days of training data. For example, 20 training data were selected randomly from each of the five days of training data to form the 100 training data for one experiment. A ME classifier was trained using these 100 training samples. And the testing data are processed through this classifier to get an experimental result. Such process is repeated five times using five sets of randomly selected training data. Taking 100 training data as an example, there are five randomly selected sets of training data, each containing 100 programs. Therefore, the recommendation is performed five times, and the final result for recommending the TV programs of a specific category is obtained by averaging the five testing results. Similar experiments are conducted for 200, 300, 400 and 500 training samples.

The ME classifier utilizes the following features in our experiment: time, title and program description (synopsis), program duration as well as program rating. As we have discussed above, the recommendation is performed on each category. We list the experimental results in Tables 3–8.

Table 3 shows the experiment results of TV series recommended for user A. Using 500 training samples as an example, 189 out of the 500 training samples are the ones that user selected. 1670 program segments in the testing data are tagged by the user as liked series. The ME classifier classified 1360 program segments as those to recommend to the user. Out of these 1360 program segments, 1137 are correctly

Table 3. Result of "series" recommended for user A.

Training size	100	200	300	400	500
Training data (liked)	36	74	101	145	189
Testing data (liked)	1670	1670	1670	1670	1670
Classified as liked	1295	1320	1334	1345	1360
Correctly classified	1041	1087	1102	1117	1137
Recall (%)	62.33	65.09	65.99	66.89	68.08
Precision (%)	80.39	82.35	82.61	83.05	83.60
F1	70.21	72.71	73.33	74.20	75.02

classified. The recall for this classifier is calculated as $1137/1670 = 0.6808$, and the precision for this classifier is calculated as $1137/1360 = 0.836$.

Similarly, we conducted experiments on movies, shopping and sports programs, as well as specials and news for user A. The results are listed in Tables 4–8, respectively.

Table 4. Result of "movies" recommended for user A.

Training size	100	200	300	400	500
Training data (liked)	37	63	106	131	170
Testing data (liked)	427	427	427	427	427
Classified (liked)	325	353	377	384	391
Correctly classified	197	221	243	259	277
Recall (%)	46.14	51.76	56.91	60.66	64.87
Precision (%)	60.62	62.61	64.46	67.44	72.70
F1	52.26	56.63	60.45	63.98	67.64

Table 5. Result of "shopping" programs recommended for user A.

Training size	100	200	300	400	500
Training data (liked)	23	51	74	103	129
Testing data (liked)	165	165	165	165	165
Classified (liked)	120	136	143	147	152
Correctly classified	84	97	105	111	119
Recall (%)	50.91	58.79	63.64	67.27	72.12
Precision (%)	70.00	71.32	73.43	75.51	78.29
F1	59.23	64.47	68.19	70.90	75.02

Table 6. Result of "sports" programs recommended for user A.

Training size	100	200	300	400	500
Training data (liked)	43	90	132	174	221
Testing data (liked)	202	202	202	202	202
Classified (liked)	154	167	173	180	186
Correctly classified	116	127	135	143	149
Recall (%)	57.43	62.87	66.83	70.79	73.76
Precision (%)	75.32	76.05	78.03	79.44	80.11
F1	65.19	68.99	71.75	74.78	76.92

Table 7. Result of "specials" recommended for user A.

Training size	100	200	300	400	500
Training data (liked)	41	83	117	165	204
Testing data (liked)	163	163	163	163	163
Classified (liked)	102	113	121	127	135
Correctly classified	65	73	80	87	95
Recall (%)	39.88	44.79	49.08	53.37	58.28
Precision (%)	63.73	64.60	66.12	68.50	70.37
F1	48.83	53.10	56.44	59.88	63.65

Table 8. Result of "news" recommended for user A.

Training size	100	200	300	400	500
Training data (liked)	43	92	130	175	239
Testing data (liked)	184	184	184	184	184
Classified (liked)	125	143	157	164	170
Correctly classified	97	112	126	134	142
Recall (%)	52.71	60.87	68.48	72.83	77.17
Precision (%)	77.60	78.32	80.25	81.71	83.53
F1	62.57	68.54	73.79	77.04	80.06

The above results are summarized in Fig. 6. As shown, the horizontal axis represents the number of training samples used in training the classifier. The vertical axis represents the F1 values for each experiment. Different curves represent different categories. As training size gets larger, the F1 results get better.

Similar experiments were conducted for different users. We calculate the average F1 value of 500 training samples for different TV programs (by average) for users A–D and present this result in Fig. 7. As shown, the recommendation results are consistent among different users.

Further analysis on our experimental results is as follows:

1. "Movies" and "specials" cover a much broader range of interest. Thus, the synopsis is not as homogeneous as "news", "sports", etc. The F1 value is generally lower than the other categories. This is especially true for "specials" which can be so diversified that training for an efficient classifier can become a very difficult task.

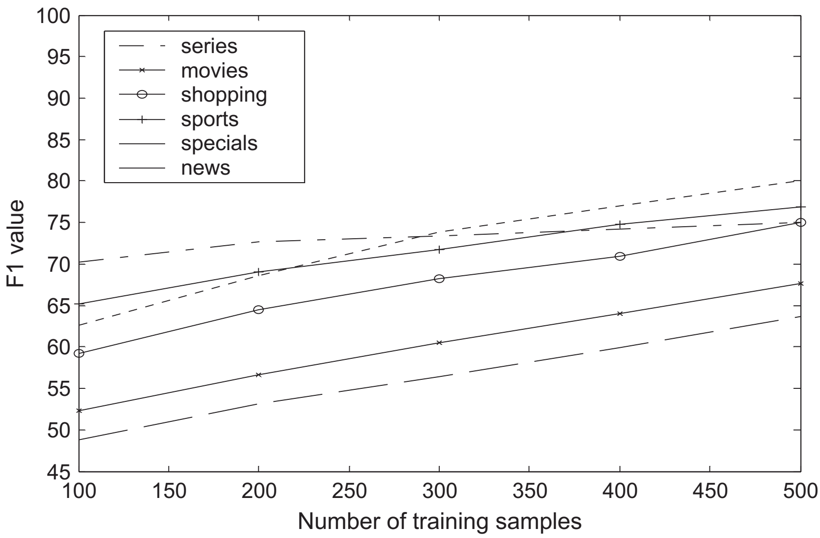


Fig. 6. F1 value of EPG recommendation for user A.

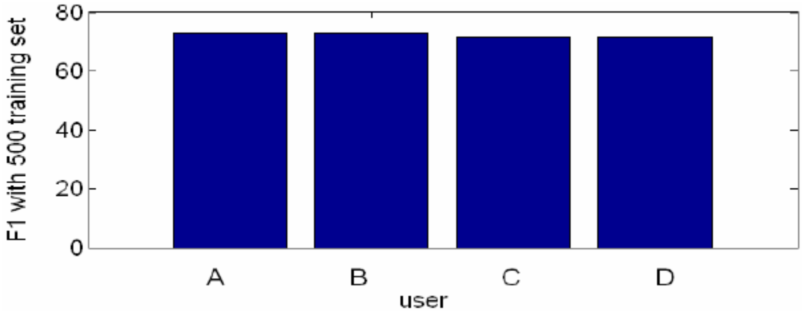


Fig. 7. F1 value of EPG recommendation using 500 training data.

2. The size of training data plays an essential role in the recommendation system. As we observe from the experimental result, the accuracy increases almost linearly with the size of the training data. In practice, the training size can be made much larger than the current experiment and the accuracy of the system should improve considerably. The system can also be re-trained once more data is available through user feedback.
3. The differences among different users can be attributed to the fact that some of the users' preferences happen to be the program with more ambiguous EPG description than the other user. With more training data, these differences among users should decrease.

5.5. *Preliminary evaluation of domain based recommendation*

Although domain based recommendation is not part of the current prototype system, we have conducted preliminary evaluation to study the behavior of recommender based on domain information. The experimental results are described as follows.

We have selected a data set of 1807 programs marked as "sports" domain by the user. These programs turned out to be distributed among multiple categories as shown in Table 9. The recommender recommends "sports" related programs and this result is compared against user's predefined labels. The recommendation results are shown in Table 10.

Furthermore, the above 122 correctly classified programs from 500 training sets are distributed among multiple categories as shown in Table 11. Among the correctly classified programs for Sports domain, those categorized as "Sports" only comprise a fraction of the same domain. There are other Sports related programs in "Series", "Movie", "Shopping", "Special" and "News" categories. Clearly, the domain of interest is an influential part of recommendation, and cannot be replaced by category (genre) based recommendation. Therefore, domain based recommendation should be continuously explored in the future.

Table 9. Sports domain data collected from different categories.

Series	Movie	Shopping	Sports	Special	News
408	25	310	786	71	207

Table 10. Experimental result of sports domain recommendation for user A.

Training size	100	200	300	400	500
Training data (liked)	39	81	123	159	208
Testing data (liked)	169	169	169	169	169
Classified (liked)	106	123	135	144	151
Correctly classified	88	103	110	116	122
Recall (%)	52.07	60.95	65.09	68.64	72.19
Precision (%)	83.01	83.74	81.48	80.56	80.79
F1	64.00	70.54	72.36	74.12	76.25

Table 11. Distribution of sports domain data correctly classified.

Series	Movie	Shopping	Sports	Special	News
30	3	22	53	4	10

6. Conclusion

Among home entertainment services, electronic programming guide (EPG) is perhaps the most appealing application for television, and its services continue to grow in the emergence of the new digital TV market. Our proposed system features an EPG collection from nonproprietary data sources (i.e. HTML on the Internet) and an EPG recommender based on text classification.

The prototype of EPG recommender is implemented under standard open architecture home networking platform and the viewing of EPG on a portable device is enabled through existing SIP network. The presented work and prototype have suggested a feasible architecture and technology for providing a personalized EPG service that can be deployed on the home network. The average EPG browsing and recommendation time on the portable device (from user sending a UI command until the requested content is displayed) is below 1 s.

As far as we are aware, the proposed work is one of the few 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 experimental results on a small scale EPG data set has shown a text classification rate of 90% and a recommendation rate of near 80%.

Our next step is to systematically collect EPG training corpus and achieve a more reliable ME model through larger scale training. Our future work also lies in further exploration of the behavior of domain information in its contribution to the recommender system. Our preliminary evaluation of domain based recommendation has already shown promising results in this perspective.

References

1. L. Ardissono, C. Gena, P. Torasso, *et al.*, Personalized recommendation of TV programs, *Proc. 8th AI*IA Conf.*, Lecture Notes in Artificial Intelligence, Pisa (2003).
2. L. Ardissono, C. Gena and P. Torasso, User modeling and recommendation techniques for personalized electronic program guides, in *Personalized Digital Television: Targeting Programs to Individual Viewers*, Human-Computer Interaction Series, eds. L. Ardissono, M. T. Maybury and A. Kobsa (Kluwer Academic Publishers, 2004), pp. 3–6.
3. A. Berger, S. D. Pietra and V. D. Pietra, A maximum entropy approach to natural language processing, *Comput. Ling.* **22**(1) (1996) 58–59.
4. Y. Blanco, J. Pazos, M. Lopez, A. Gil and M. Ramos, AVATAR: an improved solution for personalized TV based on semantic inference, *IEEE Trans. Consumer Electron.* **52**(1) (2006) 223–232.
5. G. Chang, C. Zhu, M. Y. Ma, W. Zhu and J. Zhu, Implementing a SIP-based device communication middleware for OSGi framework with extension to wireless networks, *IEEE CS Proc. First Int. Multi-Symp. Computer and Computational Sciences (IMSCCS—06)* **2** (IEEE/Computer Society Press, China, June 2006), pp. 603–610.
6. P. Cotter and B. Smyth, PTV: intelligent personalised TV guides, *Proc. 12th Innovative Applications of Artificial Intelligence (IAAI) Conf.* (2000).
7. DirectTV Guide. http://www.directv.com
8. M. Ehrmantraut, T. Herder, H. Wittig and R. Steinmetz, The personal electronic program guide — towards the pre-selection of individual TV programs, *Proc. CIKM'96* (1996), pp. 243–250.
9. C. Gena, Designing TV viewer stereotypes for an electronic program guide, *Proc. 8th Int. Conf. User Modeling* **3** (2001), pp. 274–276.
10. J. Hatano, K. Horiguchi, M. Kawamori and K. Kawazoe, Content recommendation and filtering technology, *NTT Tech. Rev.* **2**(8) (2004) 63–67.
11. T. Isobe, M. Fujiwara, H. Kaneta, U. Noriyoshi and T. Morita, Development and features of a TV navigation system, *IEEE Trans. Consumer Electron.* **49**(4) (2003), pp. 1035–1042.
12. D. J. Ittner, D. D. Lewis and D. D. Ahn, Text categorization of low quality images, in *Symp. Document Analysis and Information Retrieval*, Las Vegas, 1995.
13. T. Joachims, Text categorization with support vector machines: learning with many relevant features, *Machine Learning: ECML-98, Tenth European Conf. Machine Learning* (1998), pp. 137–142.
14. D. Lewis, A comparison of two learning algorithms for text categorization, *Symp. Document Analysis and IR* (1994).
15. D. Lewis, R. Schapire, J. Callan and R. Papka, Training algorithms for linear text classifiers, *Proc. ACM SIGIR* (1996), pp. 298–306.
16. M. Y. Ma, J. Zhu, J. K. Guo and G. Chang, Electronic programming guide recommender for viewing on a portable device, *Proc. Workshop of Web Personalization, Recommendation Systems and Intelligent User Interfaces (WPRSIUI)* (Reading, UK, October, 2005).
17. A. McCallum and K. Nigam, A comparison of event models for naïve Bayes text classification, *AAAI-98 Workshop on Learning for Text Categorization* (1998).
18. K. Nigam, J. Lafferty and A. McCallum, Using maximum entropy for text classification, *IJCAI-99 Workshop on Machine Learning for Information Filtering* (1999), pp. 61–67.
19. OSGi: Open Services Gateway Initiative. http://www.osgi.org

20. A. Pigeau, G. Raschia, M. Gelgon, N. Mouaddib and R. Saint-Paul, A fuzzy linguistic summarization technique for TV recommender systems, *Proc. IEEE Int. Conf. Fuzzy Systems* (2003), pp. 743–748.
21. H. Shinjo, U. Yamaguchi, A. Amano, *et al.*, Intelligent user interface based on multimodel dialog control for audio-visual systems, *Hitachi Rev.* **55** (2006) 16–20.
22. SIP: Session Initiation Protocol. http://ietf.org/html.charters/sip-charter.html.
23. Specification for Service Information (SI) in DVB Systems, DVB Document A038 Rev. 1 (May 2000).
24. T. Takagi, S. Kasuya, M. Mukaidono and T. Yamaguchi, Conceptual matching and its applications to selection of TV programs and BGMs, *IEEE SysInt. Conf. Systems, Man and Cybernetics* **3** (1999) 269–273.
25. TV Anytime Forum. http://www.tv-anytime.org
26. TV Guide. http://www.tv-guide.com
27. J. Xu, L. Zhang, H. Lu and Y. Li, The development and prospect of personalized TV program recommendation systems, *Proc. IEEE 4th Int. Symp. Multimedia Software Engineering (MSE)* (2002).
28. Y. Yang and J. P. Pedersen, Feature selection in statistical learning of text categorization, *14th Int. Conf. Machine Learning* (1997), pp. 412–420.
29. Z. Yu, X. Zhou, X. Shi, J. Gu and A. Morel, Design, implementation, and evaluation of an agent-based adaptive program personalization system, *Proc. IEEE 5th Int. Symp. Multimedia Software Engineering (MSE)* (2003).
30. L. Zhang, J. Zhu and T. Yao, An evaluation of statistical spam filtering techniques, *ACM Trans. Asian Lang. Inform. Process. (TALIP)* **3**(4) (2004) 243–269.

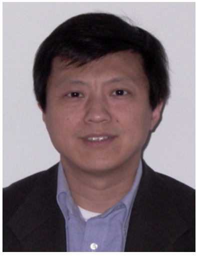
Jingbo Zhu received a Ph.D. in computer science from Northeastern University, P.R. China in 1999, and has been with the Institute of Computer Software and Theory of the same university since then. Now he is a full professor in the Department of Computer Science, and is in charge of research activities within the Natural Language Processing Laboratory.

He has published more than 70 papers, and holds one United States patent.

Dr. Zhu's current research interests include natural language parsing, machine translation, text topic analysis, knowledge engineering, machine learning and intelligent systems.

Zhenxing Wang received his B.S. degree in computer science from Northeastern University, P.R. China in 2001. He then continued his graduate study at the Institute of Computer Software and Theory in Northeastern University, Shenyang, China. Now he is a master's degree candidate supervised by Prof. Jingbo Zhu at the Natural Language Processing Laboratory.

His current research interests include natural language parsing and machine learning.



Matthew Y. Ma received his Ph.D. from the Electrical and Computer Engineering Department at Northeastern University, Boston, Massachusetts. He obtained M.S. and B.S., both in electrical engineering from State University of New York at Buffalo and Tsinghua University, Beijing, respectively. Dr. Ma is currently with IPVALUE Management Inc. Prior to that, he worked as a Senior Scientist at Panasonic R&D Company of America for 11 years, where he managed a research group which focused on Panasonic's document and mobile imaging business. Dr. Ma has 11 granted US patents and is the author of several dozen conference and journal publications. He is the associate editor of the *International Journal of Pattern Recognition and Artificial Intelligence* (IJPRAI). He is also the guest editor of IJPRAI special issues in Intelligent Mobile and Embedded Systems (2006), and Personalization Techniques and Recommender Systems (2007). Dr. Ma has been an affiliated professor at Northeastern University, China since 2002. He served as program/session chair and program committee member for numerous international conferences.

His primary research interest includes image analysis, pattern recognition and natural language processing, and their applications in home networking and ambient intelligence of smart appliances.

Jinhong K. Guo received her B.S. degree from Tsinghua University in China, M.S. degree from Northeastern University in U.S. and Ph.D. degree from the University of Maryland, College Park, all in electrical engineering. She is a senior scientist with Panasonic Princeton Laboratory.

Her research interests include image and signal processing, pattern recognition, computer security, machine learning and natural language processing.