Implicit Feedback Mechanism to Manage User Profile Applied in Vietnamese News Recommender System

Nguyen Thac Huy, Do The Chuan, Viet Anh Nguyen*
Vietnam National University — University of Engineering and Technology, Hanoi, Vietnam.

* Corresponding author. Tel.: 844 37547463; email: vietanh@vnu.edu.vn
Manuscript submitted August 13, 2015; accepted October 17, 2015.
doi: 10.17706/ijcce.2016.5.4.276-285

Abstract: The appearance of a vast array of online newspapers makes appropriate news recommendation for users become critically important. A number of researches in this area have been conducted in recent years. This paper presents the Vietnamese content-based news recommender system. Based on information about news stories that users have read and given feedback, the system can automatically determine news stories that users may want to read later. The center of the system is a hybrid user profile which can diversely model news reading features of individual users thanks to a combination of short-term and long-term information models along with self-description. The paper also introduces and evaluates a mechanism for implicit feedback. After installing the system and conducting several experiments, as well as collecting feedback from users, it is shown that the system is relatively adaptable to users and can recommend appropriate news with a high degree of accuracy, which significantly increases the effectiveness of gathering information compared to traditional news reading.

Key words: Implicit feedback, news recommender, user modeling.

1. Introduction

Since its advent, the internet has constantly thriving at an astonishing pace, which contributes greatly to the progress of humankind. Most importantly, the internet has created a "flattening world", allowing every person, and every institution to easily connect with each other regardless of time difference or geographical distance.

However, being a Netizen is often synonymous with a demand (or a need) to receive and process huge amounts of information from various sources. The most typical source of news is online newspapers, which have become increasingly popular recently.

That the amount of information needs to be processed has been increasing at an escalating rate whereas the amount of time users can spend is limited poses a challenge to people's capacity for processing information. When users need to find answers to a particular issue, search engines like Google or Yahoo can meet their demand. However, in some cases users do not even know what they are looking for, especially in case of news stories. Therefore, users often have to surf various online newspapers such as VietnamNet.vn, dantri.com.vn or tinhte.vn to search for information that *may grasp their attention*. Due to differences in users' preferences, needs and interests, the process of searching, receiving and processing new information is affected by "noise", which refers to information that is no longer available/existent or practically useless to users. These above reasons create a need for developing News recommender systems. In recent years, a

number of researches on news recommendation have been conducted to help users optimize the amount of time spent on reading news stories every day. Several *News recommender systems* have been devised, namely *iCurrent*, *Pulse*, and notably *Google News* by Google.

However, similar systems or services for online Vietnamese newspapers are limited in terms of both number and the ability to meet users' demand. Let's take the Baomoi webpage (http://baomoi.com) as an example because it possesses similar features to those of the system we want to develop in this paper. Baomoi web page allows users to create personalized folders by declaring some key words, and the page recommends users relevant news stories from various online newspapers. However, the system has some shortcomings such as: 1) News stories which are not the user's interest are still frequently recommended after quite a long time of operation; 2) Keyword-based news recommendation is sometimes inaccurate as some keywords may appear in different contexts, and belong to different news categories; 3) News recommendation capability has not caught up with the swift change in users' taste. The above reasons make us realize the importance of developing a *News recommender system* in Vietnamese. The research conducted to set up this system is presented in the following parts of the paper. Specifically, Section 2 discusses *Information Retrieval*, *Information Filtering*, and *Recommender Systems* as well as the intrinsic property of news stories compared to other types of information. Section 3 introduces xenoNews news recommender system, including the system model, detailed system architecture, user information modeling, and user profile building. Section 4 presents experiments conducted to measure the overall performance of the content-based recommender system; with the center is the hybrid user profile (including short-term model, long-term model, and self-description). Besides, another experiment on measurement of the effectiveness of Time-coded feedback collection mechanism in Front-end (Website) is also reviewed. The last section is discussion and evaluation.

2. Literature Review

2.1. Information Retrieval

Information Retrieval – IR has become a research topic since the 1950s when people began to store enormous amounts of information in computers; and finding useful information from such collections became urgent. Early IR systems were Boolean systems which had some shortcomings, such as it was hard for a user to form a good query, and there was no document ranking. However, at the present, users of IR systems expect IR systems to do ranked retrieved. Consequently, the vector space models, the probabilistic models and the language models are used [1].

2.2. Information Filtering

Information Filtering (IF) focuses on filtering content according to user profiles. A user profile is created by two ways (a) the user explicitly inputs his/her information, or (b) the system creates the profile by implicitly learning the user's transactional behaviors [2]. The user receives information he need automatically based on his own profile in the system. One of the strengths of the IF system is the adaptability to long-term interest of the user. Information may be delivered to the user in the form of notification messages, or the IF system automatically performs an action on behalf of the user.

2.3. Classification of Recommender Systems

Due to the increasing amounts of information available on the Internet, people will quickly become overloaded with information and multimedia data. Information overload is no longer just a concept, but has become a reality. Consequently, there exists a need for automated information retrieval methods. Intelligent Information Agents were developed with the ability to identify and exploit information based on personal features [3]-[5].

Ratings can be estimated in many different ways using methods from machine learning, approximation theory or heuristics. Recommender systems are often classified according to their approach to rating estimation. The classification was first mentioned in the articles [6]-[8] and was further developed in later researches; for instance, the research by Balabanovic and Shoham [9], [10] mentioned three types: 1) Content-based recommendations: the user will be recommended items similar to the ones the user preferred in the past; 2) Collaborative recommendations: the users will be recommended items that people with similar tastes and preferences liked in the past; 3) Hybrid approaches: combine collaborative and content-based methods.

2.4. Content-Based Recommendation

The content-based approach to recommendation has its roots in researches on information retrieval [11], [12], and information filtering [12]. Because of the early researches made by the information retrieval and filtering communities, and because of the importance of text-based applications, many current content-based systems focus on recommending items containing textual information, such as documents, websites or news stories. The improvement over the traditional information retrieval approaches comes from the use of *user profiles* that contain information about users' preferences, attributes and needs.

Let *Content(s)* be an *item profile* containing attributes of item *s*. It is usually a set of features extracted from *s* and is used to determine the appropriateness of the item for recommendation purposes. As mentioned earlier, content-based systems are designed mostly to recommend text-based items, specific traits in *item profile* are usually *keywords*.

As mentioned earlier, content-based systems recommend items similar to those that a user liked in the past [3], [5]. Particularly, various items are compared with items previously rated by the user, and the best matching items are recommended. Formally, let *ContentBasedProfile(c)* 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 rated by the user and are often constructed by keyword analysis techniques in information retrieval. For example, *ContentBasedProfile(c)* can be defined as a vector of weights (w_{c1},... w_{ck}), where each weight w_{ci} denotes the importance of keyword k_i to user *c*, and can be computed from vectors rated by the user. Various techniques have been used. For instance, some averaging approaches, such as the Rochio algorithm [13] can be used to compute *ContentBasedProfile(c)* as an average vector from individual content vectors. This technique was researched by two independent groups of researchers, one by Lang [3], and the other by Balabanovic and Shoham [9].

Besides the traditional heuristics (which are based mostly on information retrieval methods), there are other techniques such as Bayesian classifiers and various machine learning techniques, including clustering, decision trees, and artificial neural networks [8]. These techniques differ from information retrieval — based approaches in that they calculate utility predictions based not on a heuristic formula (such as the cosine similarity measure), but rather are based on a *model* learned from the data using statistical learning and machine learning techniques. As mentioned in [8], [9], content-based recommender system has several limitations as the following:

Limited item content: Content-based techniques are limited by the features that are associated with the objects that are recommended by these systems. Therefore, in order to have a sufficient set of features, the content must either be in a form that can be parsed automatically by a computer (e.g., text), or the features should be assigned in some way. Whereas information retrieval techniques work well in extracting features from text documents, some types of data have a problem with automatic feature extraction (such as multimedia data, e.g., graphical images, audio streams, and video streams).

Overspecialization: When the system can only recommend items that score highly against a user profile, the user is limited to being recommended items that are similar to those already rated. This problem is often

addressed by introducing randomness to the system. The use of genetic algorithm has been proposed to solve this problem in the research conducted by Sheth and Maes in 1993 [14]. It is important to note that in certain cases, items should not be recommended if they are too similar to something the user has already known (such as news stories). Therefore, some content-based recommender systems like DailyLearner [15] filter out items not only if they are too different from the user's preferences, but also if they are too similar to something the user has read before. In conclusion, *diversity* is the desirable feature in recommender systems.

New user problem: The user has to rate a sufficient number of items before a content-based recommender system can really understand the user's preferences and present the user with reliable recommendations. Therefore, a new user would not be able to get accurate recommendations.

3. xenoNews — Vietnamese Recommender System

3.1. System Model

xenoNews is constructed based on the traditional *multi-tier architecture*. Specifically, there are three tiers corresponding with three main parts of the system:

- The core processor (back-end): is the part taking the primary responsible for processing, undertaking most tasks, such as: crawling news stories from online newspapers, processing the content of news stories, updating user models, calculating and recommending news stories.
- The user interface (front-end): is the web page that the user surfs to read news stories. The main functions of the front - end include: displaying news and results of the processing by back-end to the user; receiving requests from the user and collecting feedback.
- The intermediary part (middle-level): contains a database and a unit processing requests from the front-end, and undertake the following tasks: updating the user's interaction into the database, transfer data calculated by the back-end to the user through a web interface. This is considered the communication channel between the user (via the front-end) and the back-end, there is no direct interaction between the front-end and the back-end.

3.2. System Architecture

As mentioned in 3.1., xenoNews consists of three parts: *front-end* (website), *middle-level* and core processor *back-end*. Besides, it is critical to mention two vital data items in any recommender systems that are: *item profile* and *user profile*. The following part is the detailed presentation on item profile, user profile and the back-end.

3.2.1. Item profile

xenoNews takes the content-based approach. Therefore, in order to decide whether to recommend a news story to the user or not, it is important to create a user profile.

Based on the classification of such online newspapers as VietnamNet.vn, VNExpress.vn, Dantri.com.vn, etc; each story is put into one of the following news categories: News, Market, Sports, Technology, Style, Education, Health, Real estate. The system, in this case, the back-end, frequently crawls new stories from popular online newspapers such as: VietnamNet.vn, dantri.com.vn, tinhte.vn, etc. These news stories, which include basic information such as: titles, brief descriptions, images, publishing date, link to full articles (for online newspapers), are crawled through RSS feeds provided by each newspaper. Each news story is automatically classified into corresponding categories based on the RSS feeds. This process is called *basic information retrieval of news stories*.

3.2.2. User profile

The main task of such an intelligent information agent like xenoNews is automatically adapting to individual users. Therefore, the development of appropriate user modeling techniques is of central

importance. Most information retrieval systems assume that the user has specific and stable attributes. However, the news recommender system which needs to be built is not one of them. The user's query could be phrased as:" What news that I do not know yet, but will want to know?". Computing satisfactory results for such a query is not simple. The difficulty results from the number of topics that could interest the user, the increasing number of new information, the user's changing interest in these topics, etc. Moreover, the user will not be interested in information they have heard before.

Therefore, xenoNews recommender system, which adopts *hybrid user profile* that consists of separate models, is built automatically for *short-term interest* and *long-term interest*. A similar user model has been used in the Daily Learner [16]. Apart from automatically synthesized information, xenoNews contains *self-described* user interests, which are also part of the user profile.

Short-term model: is constructed based on recent observations of the user, which helps the model adjust more quickly to news stories that the user has read recently. This model uses the n most recently rated stories. The short-term model has two tasks. Firstly, it should contain information about recent events that the user has been interested in, so that stories which belong to the same thread of events can be identified. Secondly, it should allow for identification of news that the user has already known. The nearest neighbor algorithm is used to achieve the above mentioned functionality.

Long-term model: users have different general news preferences. Modeling these general preferences may prove useful for deciding if a new story, which is not related to a recent rated event, would interest the user. Based on the history of rated stories and the user's feedback, long-term interest of the user is updated periodically, encoded in binary numbers and stored on hard disk. This model's function is to learn the features of the user's general preferences for news. To achieve this goal, the *naïve Bayesian classifier* — a statistical learning algorithm by Duda *et al.* [17] has been chosen.

3.2.3. Core processor back-end

Back-end consists of four main modules which are illustrated in Fig. 1.

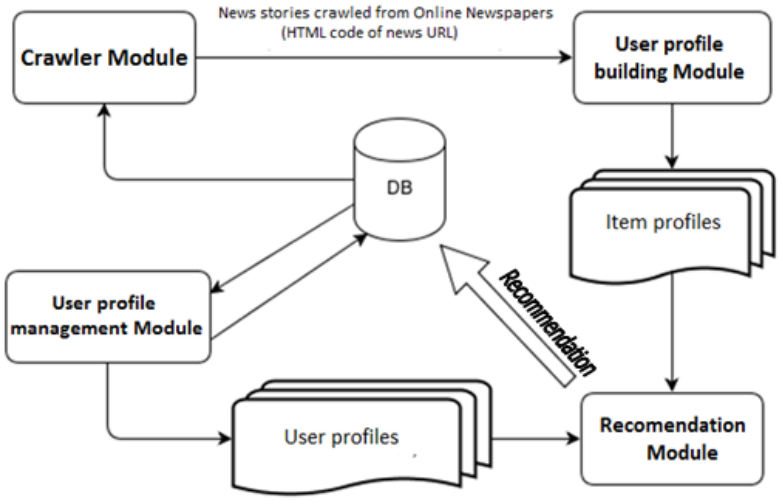


Fig. 1. Back-end main modules.

1) Crawler module: update newest stories from online newspaper into the system: Input: RSS sources stored in the database; Output: basic information of crawled news stories stored in the database; and the HTML code of online newspapers, move to user profile building module. 2) User profile building module: pre-processes the content and creates representatives for news stories: Input: HTML code of URL which contains news stories; Output: representative of news stories. 3) User profile management module: this is a special Module in the system because it is placed before the Back-end and Front-end. The functions of this

module include: Based on the history of interaction of the user with news stories in the system, update the user's long-term model periodically. Provide information about the user profile for *Recommendation Module*. 4) *Recommendation module*: Input: user profile + item profile; Output: recommended news stories, stored in the database. Fig. 2 presents more details about the tasks of modules as well as their interaction.

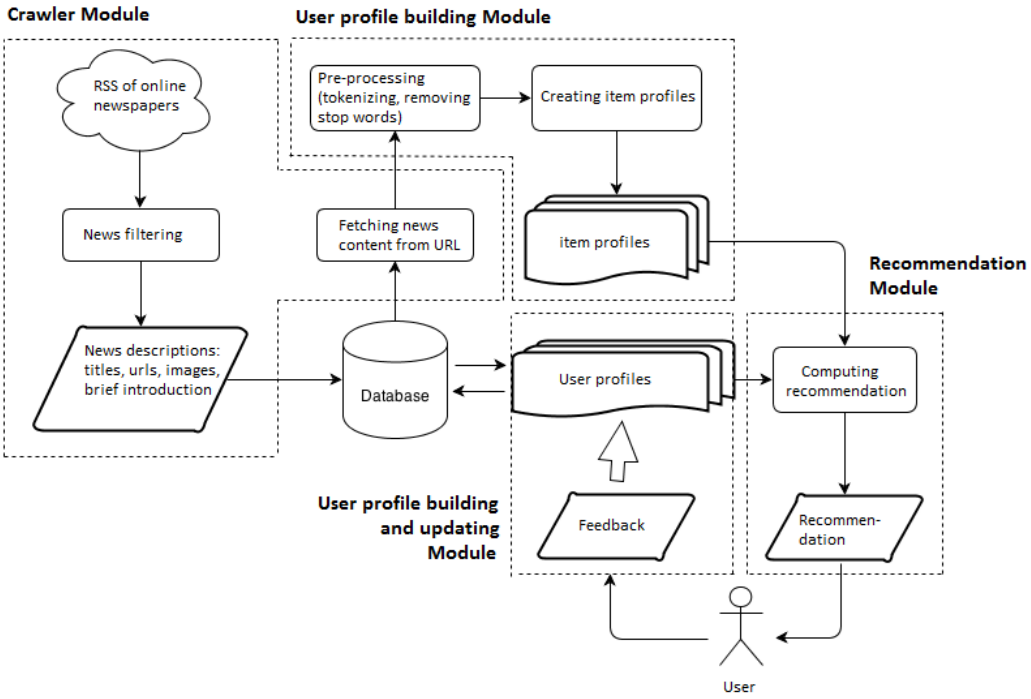


Fig. 2. Detailed interaction among back-end modules.

4. Experiments and Evaluation

4.1. Experiment Process

The experiment process consists of two main phases: 1) Firstly, *crawler module* (in the Back-end) of the system is built. 2) *Classification Preparation* (Recommendation Module in the Back-end): the crawled information is used for training in the development of the system; and used to learn parameters of short-term models, as well as of long-term models in the user profile. It is notable that data used in later experiments is not included in the training data.

Ten people participated in experiments in eight days. Every day, each user was allowed to read all new stories that have been crawled by the system and rated the news stories they had read (after reading details, or just skimming titles, brief description in the news lists). After this training session, more than 4000 feedback were collected. Each user rated in average 50 news stories per day. This amount of data may not allow correct evaluation of the overall performance of the system in case there are more users and the distribution of news changes daily. However, it is still helpful in evaluating whether hybrid user profile works as effectively as expected, and measure the performance contribution of the short-term model and long-term model to the overall performance.

4.2. Data Collection

Prior to conducting experiments, this module crawled over 40000 news stories of 8 categories from various online newspapers such as: VietnamNet.vn, dantri.com.vn, tinhte.vn, genk.vn, ndl.com.vn, tienphong.vn, news.zing.vn, ione.vnexpress.net, etc.

To transform character strings which contain single words in stories into tokens (meaningful compound words) in Vietnamese, the following tools and data have been used: 1) The tokenizer vnTokenizer version 4.1.1. According to the author's statistics in the website, this software has a high accuracy of about 97%, and it is constructed based on the combination of the Vietnamese dictionary (from VLSP project) and n-gram model, in which the n-gram model is trained using Viettreebank (a Project annotates naturally-occurring text for linguistic structure for Vietnamese) in Vietnamese (70000 sentences are tokenized). 2) A list of 570 stopwords in Vietnamese, taken from the web page on Process natural language in Vietnamese. Stopwords are words that appear too frequently in a language; therefore they have little value in computation and comparison.

4.3. Data Analysis

Evaluation process: each user's data was divided into separate *training sessions*; one training session corresponded to one day. Firstly, the system was trained with all rated examples from the first training session, and compared its recommendation with for class labels of stories from the second training session to the user's ratings. Then the training set was incremented session by session, and the system's performance was measured in the following training sessions.

Finally, the results were averaged over all users. This methodology models the way the system is used realistically. The results obtained from this experiment were presented in Table 1.

Table 1. Average Data over All Users, after Each Training Session

No	Precision			Recall			F1		
	S-T	L-T	Hybrid	S-T	L-T	Hybrid	S-T	L-T	Hybrid
1	72.6	28.8	51.9	26.2	17.0	32.1	38.5	21.4	39.7
2	70.1	32.7	72.1	41.5	36.7	48.8	52.1	34.6	58.2
3	74.6	41.2	75.5	53.0	58.3	57.7	62.0	48.3	65.4
4	80.2	40.6	83.0	59.6	66.9	61.1	68.4	50.5	70.4
5	76.9	38.2	78.1	58.6	71.4	61.3	66.5	49.8	68.7
6	83.1	66.8	86.8	56.3	63.9	61.4	67.1	65.3	71.9
7	85.8	67.7	87.7	54.8	72.0	61.5	66.9	69.8	72.3

Fig. 3 was built based on data in Table 1. Fig. 3 shows the positive change of Precision and Recall measure after training sessions; therefore, F1 measure is also improved.

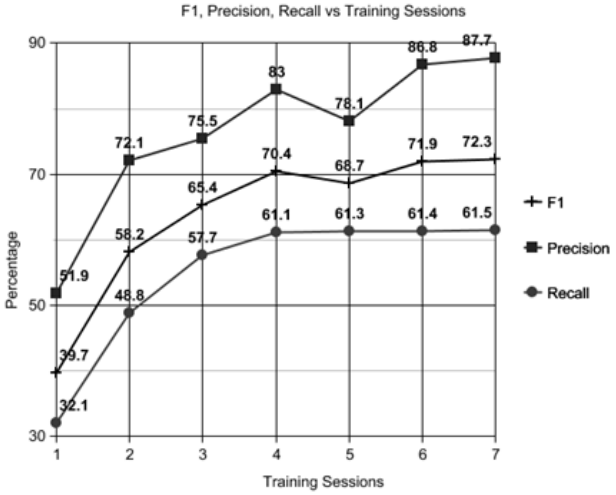


Fig. 3. The change of precision, recall and F1 measures after each training session.

In Fig. 4, the F1 score measure is used to present the performance of the system in the form of functions of training sessions. The graph shows a rapid increase of classification during the first few training sessions, and then starts to fluctuate as a result of changing distributions of daily news stories. The figure also shows the relative performance of the two user model components. As expected, the hybrid approach combining a short-term and long-term user model outperforms each individual approach with respect to the F1 measure.

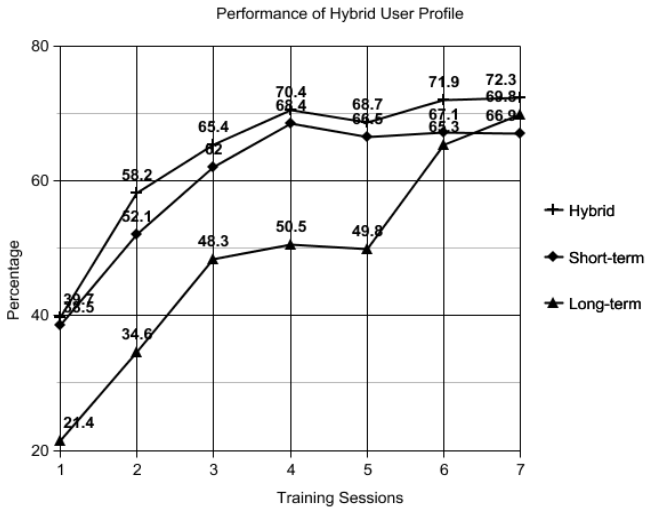


Fig. 4. The system's performance presented in F1 measure.

Time-coded is the implicit feedback collection mechanism, which plays a critical role in xenoNews because it allows the system to retrieve much more users' feedback automatically. In xenoNews, two time thresholds are chosen. If the time spent reading news stories exceeds these thresholds, the system automatically determines that the user is interested in the news stories: For short stories with under 700 characters, the threshold is 22 seconds; for other stories: the threshold is 35 seconds.

To measure the performance of this mechanism, a small experiment was conducted. The system display successively news stories that the user rated in the previous experiment. The system shows users how they rated that news stories before (interesting, not interesting). (1) If they rated "not interesting" previously, they are recommended to click "Move to next news stories" immediately. (2) If they rated "interesting" previously, they are recommended to read the stories as usual. However, in case they have not finished the stories but the threshold is reached, the system automatically moves to the next stories. The user can move to the next stories even before the time threshold is reached. The result of the experiment is presented in Table 2.

Table 2. Automatic Measure of Time-Coded Implicit Feedback Mechanism

Cases	Predicted Negative	Predicted Positive
Negative Cases	TN: user:"not interesting" + xenoNew:"Irrelevant" 2245	FP: user:"not interesting" + xenoNew:"Relevant" 5
Positive Cases	FN: user:"interesting" + xenoNew:"Irrelevant" 19	TP: user:"interesting" + xenoNew:"Relevant" 2089

From the table, it can be calculated that:

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{2089}{2089 + 5} \approx 99.8\% \quad ; \quad \text{Recall} = \frac{TP}{TP + FN} = \frac{2089}{2089 + 19} \approx 99.1\%$$

4.4. Discussion

It is difficult to measure the overall performance of the xenoNews system due to the following reasons: 1) There is no standard data to measure performance and compare algorithm. 2) User profiles try to approximate model of users' interests.

However, their "interests" are only relative. Users' interests are neither *static* nor *consistent*. Therefore, a user going through the same list of stories at different times might have different ratings. 3) Standard evaluation methodologies in machine learning, such as n-fold cross-validation are not applicable to this scenario. This is mainly due to the chronological order of news stories. Therefore, applying such methodologies can cause skewing results. 4) Distributions of news stories are uneven, for example: everyday, the number of stories of different categories varies greatly. Experiment in implicit feedback yielded good results is easy to understand, because the number of "super short" news (news which has less than 700 characters, equivalent to 4-5 sentences in Vietnamese) is limited. The users may have finished the stories but the reading time is not enough for the Time-coded to decide to put it into the "Relevant" class or not. Sometimes, the user may not be interested in a news story but it is still labeled "Relevant" - meaning related to the user's interest by the system, but this is quite rare. The reason for this is that the user may become neglectful whereas reading and does not close the story, or move to the next story immediately.

5. Conclusion

The paper has presented the functions, design as well as algorithms of an adaptive system — xenoNews, which can "learn" users' interest from news stories rated by users everyday; then xenoNews recommends relevant stories in the next days. xenoNews uses the *content-based*, *multi-strategic approach* to model short-term and long-term information of users separately, and combines with self-described user rules to create a hybrid user profile.

Even though the sample content-based news selection system worked quite well, it is believed that researching on the following issues and integrate them in xenoNews will help the system perform better: 1*) Increase the use of cooperative information*: cooperation-based recommendation approach seems to be promising and can be combined with the current system to create a hybrid system. 2) *Feedback collection mechanism*: as mentioned earlier, users' feedback play a vital role in the system. Research on *implicit feedback*, especially the "*irrelevant*" feedback will be carried out more carefully in the future.

References

[1] Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

[2] Hanani, U., Shapira, B., & Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Model. User-Adapt. Interact., 11(3)*, 203–259.

[3] Lang, K. (1995). NewsWeeder: Learning to filter news. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 331–339).

[4] Balabanovic, M. (1998). Learning to Surf: Multiagent systems for adaptive web page recommendation.

[5] Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Mach. Learn*., *27*, 313–331.

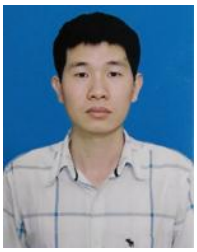
[6] Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. *Proceedings of CHI*.

[7] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. *Proceedings of 1994 Comput. Support. Coop. Work Conf*.

[8] Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating 'word of

mouth. *Proceedings of Conf. Hum. Factors Comput. Syst.*.
[9] Balabanovic, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Commun. ACM*, *40(3)*, 66–72.
[10] Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley.
[11] Salton, G. (1989). *Automatic Text Processing*. Addison-Wesley.
[12] Belkin, N. J., & Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, *35(12)*, 29–38.
[13] Rocchio, J. J. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing,* 313–323.
[14] Sheth, B., & Maes, P. (1993). Evolving agents for personalized information filtering. *Proceedings of 9th IEEE Conf. Artif. Intell. Appl*.
[15] Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access. *User Model. User-Adapted Interact.*, *10(2–3)*, 147–180.
[16] Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*.
[17] Phuong, L. H., Huyên, N. T. M., Roussanaly, A., & Vinh, H. T. (2008). A hybrid approach to word segmentation of Vietnamese texts. *Lecture Notes in Computer Science*, *5196*, 240–249.

Nguyen Thac Huy is a graduate student with VNU-University of Engineering and Technology. He is interested in information retrieval and natural language processing research.



Do The Chuan was born in Hung Yen province of Vietnam, in 1986. He received the B.S degree in networking engineering from University of Engineering and Technology, Vietnam National University, Hanoi. He is currently pursuing the M.S. degree in computer science at University of Engineering and Technology. His research interests include network security and its applications, e-learning, and recommender system.

Viet Anh Nguyen joined VNU-University of Engineering and Technology in 2000. He defended his PhD dissertation at University of Engineering and Technology in 2010, after four years of study about adaptive hypermedia in e-learning. His research interests include e-learning, m-learning, user modeling, adaptive system, recommender system. He has more than 15 year of experiences in software development, for mobile, Linux, Windows, and Web platforms using C/C++, Php, Python, among others.