User-centered Adaptive Push-type P2P Information Delivery Model

Hiroyuki Ito Atsuo Hazeyama
Graduate School of Information Education
Tokyo Gakugei University
hazeyama@u-gakugei.ac.jp

Abstract

Various forms of information delivery services have been provided with propagation and development of the Internet technology. Information push has a facility that does not need to have trouble of choosing information because of being able to get information automatically that matches to a user profile. However, in existing information push services, there is necessity to register the taste information in advance, and it is not considered that the taste information may change dynamically. Therefore there is a problem that it is difficult to acquire information widening the field of one's interest for a user and deliver information adapted user's taste for the delivery side. The authors propose a means to build a user profile dynamically by using content-based filtering method. They also propose the network model that connects users that have similar taste information mutually in pure peer-to-peer network and allow to trigger information push mutually. In addition, they design the network protocol to build the network model and make a prototype to show effectiveness of the proposal.

1. Introduction

By the progress of the Internet and Web technology, the amount of the contents circulated on the Internet increase rapidly. As a result, one has more opportunity to search for information for his/her purpose. A variety of information retrieval service and/or information distribution service have emerged. Current ways of acquiring information on the Internet are classified into pull-type and push-type. In existing push-type information distribution systems, users are required to register their request for information distribution in advance. This action comes upon them as well as limits the information they can acquire. It is difficult for users to get unforeseen information that may lead to discover new areas of interests. Interests of a user may change momentarily in itself. However because existing systems do not deal with information of interests users have in a dynamic manner, it is difficult for systems to change the information that is distributed according to the change of interests by users.

Collaborative filtering is a method for evaluating tastes of users in a dynamic manner. This method requires the number of users in proportion to the quantity of the information distributed. If contents do not have feedback from users, they will not be recommended [6]. Therefore collaborative filtering is not appropriate for distributing web contents.

This paper proposes adaptive push-type information distribution model, which enables to distribute information by deriving information of interests of users in a dynamic manner even when they change their interests and/or tastes.

Gnasa et al. made a study on dynamic push-type information delivery [1]. They proposed to configure a network group of users whose related information is similar via peer-to-peer. The related information is derived from query information the user made in the past, the contents browsed as the results of queries and the evaluations for the contents. If a user wants to receive contents by push-type information delivery, correlations of evaluation value between the user and other users in the same group are calculated. Contents that were determined to be useful for her/him are delivered. Gnasa et al. adopt collaborative filtering based on evaluation score, therefore they do not deal with contents themselves. It is not realistic to hold evaluation score for web contents in a local computer because the amount of web contents is huge and changeable chronologically.

This paper is organized as follows: section 2 describes a user profile generation method. Section 3 proposes a network model for push-type information delivery and section 4 describes a prototype system to realize the proposed model. Finally section 5 concludes this paper.

2. User Profile

In order to provide information services that include recommendation like push-type information delivery, the system must recognize interests and/or tastes of users in the form of profile. However, in traditional push-type information delivery systems, a user profile is created as static information, i.e., channel selection and/or check boxes. These methods do not take chronological change of interests and/or tastes of users into consideration. Task of inputting gets on top of users. This section describes representation of a user profile and its construction method.

2.1. User profile representation scheme

We aim at constructing a network model that enables to change delivered information according to the transition of interests and/or tastes of users. For that purpose, the user profile must represent chronological transition of a user's interests and/or tastes. Applying collaborative filtering is not appropriate for generating a user profile in push-type contents delivery on the Web because the amount of Web contents is now quite huge and it will increase day by day, thus it is not realistic to manage evaluations for all of them. Therefore this study tries to derive the user profile by contents analysis of Web pages.

(1) User profile

Users usually browse the information that corresponds with their interests and/or tastes. We assume browsing history of Web contents reflects on users' tastes and/or contexts. We consider to construct a user profile based on the browsing history automatically. It is not appropriate to simply analyze all the contents a user browsed, because the contents that a user browsed for making sure do not always reflect on their tastes although (s)he iterates search operations till reaching the required contents. On the other hand, it is overloaded that a user has to do some operations in order to clarify the target of analysis. Thus we determine whether contents should be the target of analysis according to duration of browsing them.

(2) Representation by a vector space model

Contents that match a user's tastes can include some sort of useful terms. Therefore we assume a user's interests and/or tastes can be represented by some useful terms that stand for them. We will represent a user profile by a vector space model, which is a kind of text mining technique. The vector space model was proposed by Salton et al. [5]. It represents arbitrary text information with vector and determines the contents by their direction. Terms are represented as an axis and degree of importance is represented as the element. It is general to calculate the degree of importance for terms according to statistics for them.

2.2. Document vectors

We call a vector composed of terms extracted from the contents, which were determined to match with a user's taste by a text mining technique as document vector. We show a derivation procedure of document vectors in Figure 1 and describe the details in the following.

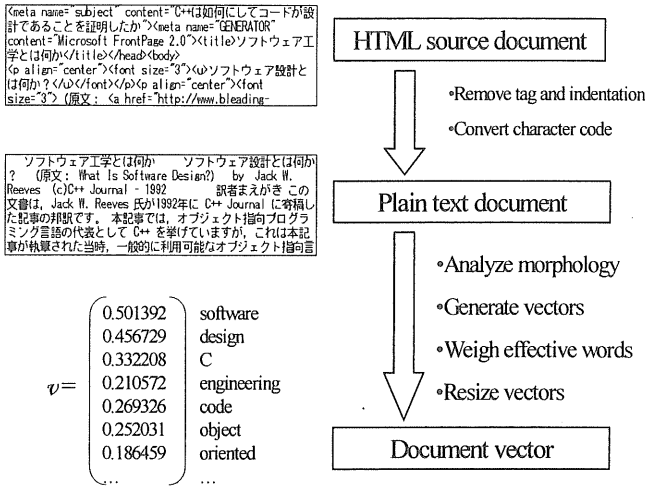

Figure 1. Document vector derivation procedure

(1) Extraction of a plain text document from a HTML document

A document vector is generated from a HTML document that corresponds to the Web contents a user browsed. As the first step, a plain text file is extracted from a HTML document. Since various character code sets are used to create web contents, the character code set of them is converted into a specific one.

(2) Processing to a plain text file

The morphological analysis is conducted to a plain text file that was extracted by the previous step. Here, noun terms and unknown terms are extracted (we call the terms useful terms). Term Frequency (TF) is recorded for each extracted term. A document vector space is generated by term frequency of all useful terms a HTML document contains. As the weight of importance for terms in the vector space model, TF, Inverse Document Frequency (IDF), and TF*IDF that multiples with IDF, Chi-square measure to Frequent terms (CF) are known. We use only TF

because the purpose of this study is not information retrieval, so IDF is not needed.

(3) Processing to a document vector

When a document vector was generated, dimension of the vector corresponds with the number of useful terms in the target document. Most contents contain hundreds or thousands of useful terms. This means not only increase of calculation cost but also inclusion of noisy terms, which do not represent the contents. To solve this problem, we resize the document vector so that more important terms are included in it. This study sets sixty to the provisional value.

2.3 Lifetime of vectors

Tastes of a user may change as time goes by. While contents analysis methods can generate a user's profile, they do not consider chronological transition. In this study, we set lifetime for each document vector that was derived in the previous section. Document vectors that passed away their lifetime will be breached. When the contents that correspond to a derived document vector were browsed, the document vector will be extended its lifetime during a pre-defined period unless the contents are changed. If the contents are changed, a new document vector is generated and lifetime for it is set. We expect our method can adapt transition of users' interests and/or tastes flexibly by introduction of lifetime.

2.4 Classification of users' tastes

Generally speaking, a user may have a variety of tastes. Therefore a set of document vectors includes various elements that were derived from various contents based on users' interests. If only one user profile is constructed from all document vectors, document vectors with little similarity are dealt together. This means that characteristics each document vector has are lost in a user profile. Therefore we apply a clustering technique for set of document vectors and classify set of document vectors with high similarity into some clusters. Then we calculate the center from all vectors included in each classified cluster. In this study, we define vectors which represent a user profile "user vectors". Clustering techniques are classified into hierarchical clustering and non-hierarchical clustering. Hierarchical clustering consists of nested clusters. On the other hand, non-hierarchical clustering generates pre-defined number of clusters. We adopt non-hierarchical clustering because document vectors obtained from browsing histories by a user are targets of our study and we do not require the generation process of clusters.

2.5 User taste vectors

It is said for web contents, which include various topics like portal sites or news sites, not to be suitable to apply a method based on statistical data on terms [2]. It is desirable to reflect a user's intents and/or contexts on his/her profile as much as possible.

From this background, we provide an editing function for a user profile that can append and delete any vectors. We call the vectors directly appended by a user taste vector. Lifetime is applicable to user taste vectors. Although user taste vectors do not affect user vectors directly, supplementary effects for more appropriate clustering results by appending a user's taste vectors for clustering are expected. Figure 2 shows relationships among several types of vectors of a vector space in this study.

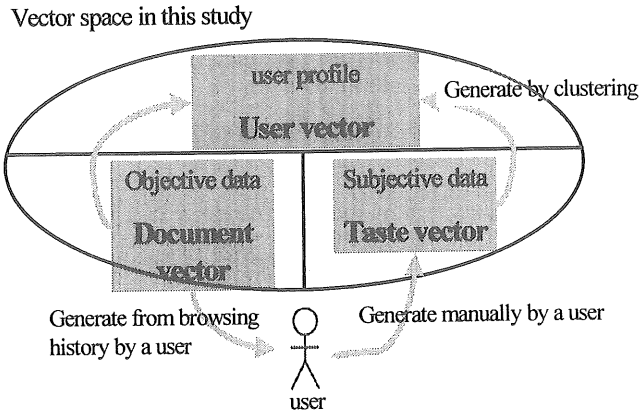

Figure 2. Relationships among several types of vectors

3. Network Model for Push-based Information Delivery

3.1. Concept

We adopt pure push-type peer-to-peer model [7] as the basis of an information delivery network model. The reasons we adopted this model are as follows:

- Bi-directional delivery and reception services of web contents are realized by pure peer-to-peer model
- Anonymity is guaranteed in delivering a private user profile

(1) Network model

This study aims at constructing a mesh network. The number of links from a node to other nodes limits to the number of user vectors of the node. It is general to construct a network by nodes collaborating in the pure peer-to-peer network model. That is, nodes in a network we suppose have links that

correspond to the number of final clusters. The number was determined in clustering the document vectors and taste vectors. In the network, a node corresponds to a user. As the network is the pure peer-to-peer model, it is necessary to have a mechanism to discover initial nodes. To solve this problem, this study introduces a coordination node that has a fixed address. In this way, all nodes that participate in the network connect to this coordination node first, and then they know other nodes via the coordination node and switch connections to appropriate nodes.

(2) Information flow

The pure peer-to-peer network enables to provide bi-directional services between nodes. By using this mechanism, we implement push-type information delivery. When a user takes an action associated with information delivery during browsing contents, push-type information delivery is done. At that time, within the node the user operated, the system generates the meta-data for the contents and broadcasts the meta-data with TTL (Time To Live: it means the number of node the packet can pass through) to neighboring nodes. The node that received the message presents it to the user, decreases one from the TTL value and broadcasts the message to its neighboring nodes. When a node updates the TTL value and the value becomes to zero, the node destroys the message. Figure 3 shows a model of information delivery among nodes.

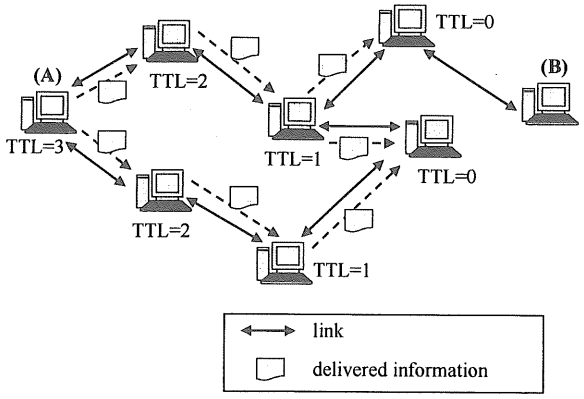

Figure 3. Information delivery model among nodes

(3) Node clustering

Under an assumption that a user will have interests in web contents which other people who have similar tastes to him/her browsed, a node connects to the nodes which have similar user vectors with higher priority. By this mechanism, push information that was delivered by the procedure of the previous section will be delivered to the nodes that have more similar tastes to the source node.

In our pure peer-to-peer network, a node constantly delivers its user vectors to the currently connecting nodes by the protocol of "request for user vectors similarity" that will be described later. The nodes that received the request calculate the degree of similarity between the source node and them. Then they send back the result to the source node by the protocol of "response to the request for user vectors similarity". The source node switches the connection to the nodes with higher similarity based on the result.

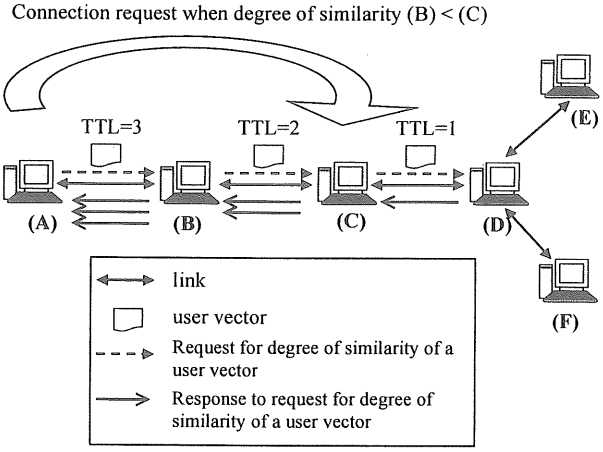

Figure4. Flow of node clustering

Figure 4 shows node A sends request for user vectors similarity to the connecting nodes. In this example, we assume the initial value of TTL is three. This means the user vector of node A reaches node D via node B and C. At this time, as node C receives a packet from node B, node C does not know the source node of the packet. This guarantees anonymity for user vectors. Node B, C, and D that received user vectors from node A calculate the degree of similarity between the received user vector and the user vector of their node respectively. They send back the result to the source node as a response to the request. Response from node D is sent to node A through node B and C. Response from node C is also sent to node A through node B. Node A compares the results and switches the connection to a node with higher degree of similarity. For example, when the highest degree of similarity is between node C and node A, node A cuts off connection with node B and then establishes connection with node C. By this exchange of connection, when node A sends a message of "request for user vectors similarity", it can reach node E and F.

Changing nodes for connection represents change of a user's tastes and/or interests. This also brings change of delivered information. Therefore it may contribute to discover new areas of interests for a user. By iterating the abovementioned requests and responses among nodes, the more difference of user

vectors, the more the logical distance among nodes. This realizes node clustering. It is expected the information is delivered to users with more similar tastes.

(4) Reliability of delivered information

Although the nodes with higher degree of similarity of their user vectors may have more similar tastes, it is not guaranteed the information delivered to a node matches to the user's tastes because all nodes can deliver information without restrictions. Therefore delivered information has some problems with respect to reliability. In order to solve this problem, we provide a filtering facility for nodes in a peer-to-peer network.

Each node stores evaluation for each information delivered from other nodes. Evaluation is done by a user of the receiver node from the viewpoint of usefulness ranging from zero to five. The receiver node records the source node ID and score of the evaluation. The receiver node filters nodes by setting threshold to the score. On the contrary, source nodes need to know how much useful the information they delivered is for the receiver nodes.

The score evaluated by receiver nodes is sent back to the source node (protocol of response to browsing).

3.2. Proposed protocol

This sub-section describes a protocol required to implement the abovementioned push-type peer-to-peer network. It runs on the TCP layer and as follows:

- Request for connection establishment
- Request for cutting off connection
- Request for degree of similarity to a user vector
- Response to request for degree of similarity to a user vector
- Request for information providing
- Response to browsing
- Ping
- Pong

Hereafter we describe the information structure delivered as protocol. We also show two messages "Request for degree of similarity to a user vector" and "Response to request for degree of similarity to a user vector" because of space limitation.

(1) Header

The header consists of message type, message ID for identifying each message, relay node information, and TTL (Time To Live) that represents the maximum number of relay nodes. The following messages inherit this header information.

(2) "Request for degree of similarity to a user vector" message

The "request for degree of similarity to a user vector" message is one, which asks for calculation of the degree of similarity between the node itself and other nodes. In order to ensure anonymity, user ID is not included in payload. Payload is body of data except for the header. A node that received this message calculates the degree of similarity between all user vectors the node has and the user vector it received. It determines a user vector that brings max degree of similarity and sends a "response to request for degree of similarity to a user vector" message to the source node.

(3) "Response to request for degree of similarity to a user vector" message

When a node receives a "request for degree of similarity to a user vector" message, this message will be generated as its reply. The message ID of this message is identical to that of "request for degree of similarity to a user vector". The nodes that received this message retrieve a message with the same message ID out of all received "request for degree of similarity to a user vector" message in them. Then value of the relay node attribute in the "request for degree of similarity to a user vector" message is extracted and the received message is sent to the relay node. This processing is iterated. Finally the reply message is sent back to the source node. The number of hop in the "request for degree of similarity to a user vector" message is set to the TTL attribute in the "response to request for degree of similarity to a user vector " message.

4. Prototype Development

This section describes a system to realize the mechanism shown in the previous section. We used C++ as programming language, Win32API as Graphical User Interface (GUI) components and Winsock as a communication driver for TCP/IP communication. The prototype system is composed of the following modules:

- Dynamic Data Exchange module
- HTTP client module
- Vector editing module
- Peer-to-Peer communication module

Figure 5 shows a system architecture. DDE module, HTTP client module, and vector editing module run in a collaborative manner to derive a user profile. The Peer-To-Peer module manages routing on the Peer-To-Peer network, connections among nodes, and deliver and/or receive Web contents according to the user profile and by using the proposed protocol.

- **Dynamic Data Exchange (DDE) Module**

DDE is a communication means to exchange data among different applications and to invoke commands on MS Windows environments. Since this study considers to create a user profile from browsing history of web contents by users, the system needs to grasp what contents a user browses in real time through DDE communication with web browsers. The system inserts the URL information, which corresponds to the contents that was judged a user browsed into a URL queue.

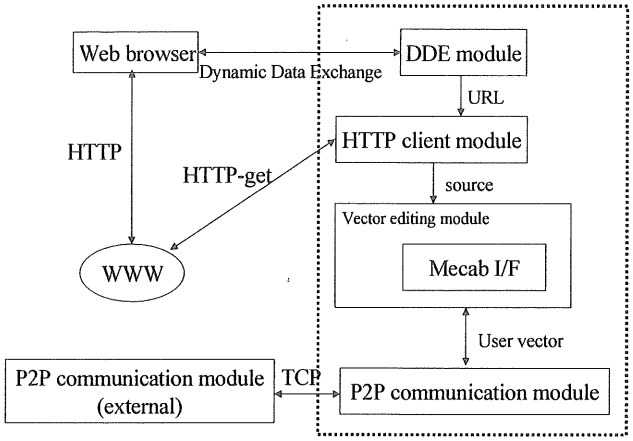

Figure 5. System Architecture

- **HTTP Client Module**

This module extracts a URL from the URL queue and executes the get request according to the URL information. Then it stores a HTML source file of the contents into a buffer of the vector editing module.

- **Vector Editing Module**

This module checks the buffer at a constant interval. If some contents are found in the buffer, this module convert their character code to Shift-JIS code, analyzes morphology of them, and then generates a document vector. This module is also responsible for appending and/or deleting a taste vectors according to a user's requests. We used Mecab [4] for morphology analysis.

- **Peer-to-Peer Communication Module**

This module manages a network with the protocol we described in the previous section and delivers and/or receives the information. When a user takes some specified actions during browsing contents, the module delivers them to neighboring nodes with the "request for information providing" message.

When the Peer-to-Peer communication module of receiver nodes receive the message, it presents the contents on the screen in the form of popup. When a user enters score of evaluation from the popup window, it is sent back to the source node with "response to browsing" message.

5. Conclusion

We have proposed a method to build a user profile dynamically by using content-based filtering technique. We have also proposed the network model that connects users that have similar taste information mutually in pure peer-to-peer network and allows to cause information push mutually. In addition, we have designed the network protocol to build the network model and made system architecture.

In the future, we will complete to implement the system (we have a plan to use JXTA [3]) and evaluate usefulness of our proposal.

References

[1] M. Gnasa, S. Alda, N. Gul, J. Grigull, and A.B. Cremers, "Cooperative Pull-Push Cycle for Searching a Hybrid P2P Network," *Proceedings of the Fourth International Conference on Peer-to-Peer Computing*, 2004, pp. 192-199.

[2] Y. Hijikata, "User Profiling Technique for Information Reccomendation/Information Filtering," *JSAI Magazine*, Vol. 19, No. 3, The Japanese Society for Artificial Intelligence, 2004, pp.1-8 (In Japanese).

[3] JXTA, http://www.jxta.org/

[4] Mecab, http://chasen.org/~taku/software/mecab/

[5] G. Salton, and M. J. McGill, "Introduction to Modern Information Retrieval", McGraw-Hill, 1983.

[6] T. Terano, "Information Recommendation Systems on the Web," IPSJ Magazine, Vol.44, No.7,Information Processing Society of Japan, 2003, pp.696-701 (In Japanese).

[7] A. Tiwana, "Affinity to Infinity in Peer-to-Peer Knowledge Platforms," *Communications of the ACM,* Vol. 46, No. 5, pp. 77-80, May 2003.