A Personal Ubiquitous Distributed Knowledge Pool System

Shyh-Chang Liu[1], Tsung-Hung Chen[1], Peng-Shiang Wang[2]

[1]Dept. of Information Eng. I-Shou University, [2]Delta Electronics, Inc. Taiwan

Abstract

Recently, owing to the fast-developing of the Internet technology, the popularity of the digital devices and the enhancement of the people's computer ability, therefore the demand of ubiquitous personal knowledge services has gradually been mature. People can use these services to collect information, solve problems and proceed e-learning. Therefore, how to provide such service becomes a worth subject.

In this paper, a personal ubiquitous Distributed Knowledge Pool System (DKPS) based on the distributed system architecture and Rich Site Summary (RSS) technology is proposed here. In addition, a collection of Cluster Tree in Vector Space Model (VSM) is used to describe the information. In particular, users can build the personalized knowledge base derived from their preference and requirements through personal information recommendation service that is the aim of this article.

Keywords: Ubiquitous mechanism, Distributed Knowledge Pool System, Rich Site Summary, Personal information recommendation service

1. Introduction

1.1. Ubiquitous Knowledge Services

Since Rob Acksyn and Don McCracken developed knowledge management system (KMS) in late 1980s, the key research of KMS is focused on either how to assist the enterprise operations, or to provide the resources for academic research [1]. In recent years, users hoped to filter the useful part from overwhelming information in the Internet, organize to a personalize knowledge base and store it in digital devices. In addition, it can be used for reference or study any time and anywhere. Because the requirements of such service are greatly increased, this research namely aims at such subjects.

We design a distributed knowledge pool system based on the advantage of ubiquitous mechanism here, aggregate information from the Internet, to provide a more effective knowledge services to launch user's query, subscription and recommend information resources. It allows users to construct their own knowledge base for personal purpose.

1.2. Knowledge Social Software and RSS Technology

In order to aggregate information resources effectively, we must select suitable content (information) providers seriously. Beside professional web sites, a noteworthy aspect of knowledge social software that achieves significant improvements of such information service in institutional network is worth considered. It can be used to foster new social patterns, to enhance the community, and to expand the critical audience for people joining. Knowledge social software includes knowledge Blogs, knowledge forums and profession web sites, and most of above utilizes RSS technology to help users to browse reference materials [2].

There are three reasons why we chose to use the characteristics of RSS mechanism to develop DKPS. The first is that RSS mechanism can concurrent aggregate information from multiple resources. Second, users can subscribe the wanted information through "RSS-push" service ubiquitously. Finally, the communication format of RSS has become a standard and accepted by many systems.

1.3. Distributed Knowledge Pool System

DKPS developed here expected to provide the following capabilities:

1) DKPS use RSS mechanism to aggregate information resources from different Internet content provider (ex. social software).
2) DKPS only saves the description of information objects; it means fewer loads for storage devices.
3) The system recommends suitable reference. [3]
4) Users query what they needs from DKPS, select the needed response, place and organize into personal knowledge base.

This paper is organized as follows: Section 2 introduces relevant information structure and algorithms. The related system structure and coordination process is illustrated in Section 3. Section 4 presents the experiment results. In Session 5, we conclude our paper.

2. Infrastructure Analysis

2.1. System Architecture

The DKPS system architecture is in the Fig. 1. Each content provider can offer a RSS service and share content via Internet. Content contains both RSS feeds and information resource. RSS feed is a kind of descriptive file used to describe information resource. The RSS Aggregator of DKPS aggregates RSS feeds, and provides the services to satisfy users' needs through the client tool. The system will analyze the users' past access records and provides suitable recommendations.

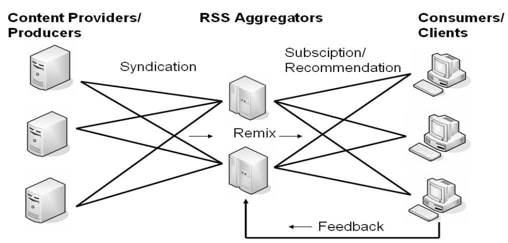

Fig. 1: DKPS Architecture

2.2. Model Analysis

The DKPS involves three services: "Subscript", "Query", and "Recommend". "Subscript" means that DKPS provides "Push" service. The RSS feed of the information resources can be integrated by the RSS Aggregators. The "Subscript" service requires content provider to supply users with RSS services followed the RSS definition. But for "Query" and "Recommend" services, the system needs a more efficient data structure to transfer relevant information to a model that can be analyzed and calculated. VSM is used to represent the relevant information. According to RSS standard, an information resource may have more than one category and the system can develop its RSS Feature Vector (RFV) based on categories:

$$\overrightarrow{R_{S_i}} = a_{i1}C_1 + a_{i2}C_2 \cdots + a_{it}C_t; ?@a_{ij} \subset \{0,1\}, C: \text{Category}$$

R_s is the set of all RSS in RSS Aggregators.

$R_s = \{\overrightarrow{R_{s1}}, \overrightarrow{R_{s2}}, \ldots, \overrightarrow{R_{sn}}\}$, $\overrightarrow{R_{si}} \subset R_s$

To inquire efficiently, the system integrates Cosine R-Tree and database to classify RFV in R_s [3]. The characteristics of Cosine R-Tree can be concluded as follows:

1) It is used to describe and save spatial objects.
2) All vectors $\overrightarrow{R_{si}}$ must be normalized.
3) All leaf nodes p forms surface S_k,
   $S_k : X_{i1}^2 + X_{i2}^2 + \ldots + X_{ik}^2 = 1$, and come into a set P, $P = \{p \in S_k\}$.
4) In Cosine R-Tree, the distance between two nodes is calculated by cosine distance:

$$Sim(X_a, X_b) = \frac{\sum_{i=1}^{k} X_{ai}X_{bi}}{\sqrt{\sum_{i=1}^{k} X_{ai}^2 \sum_{i=1}^{k} X_{bi}^2}}$$

In client site, there are two kinds of local data structure:

1) Cosine R-Tree that consists of :
   - Set of leaf nodes : includes RFV, users have accessed, and Recommend Vector (RV) described in section 2.4.
   - Set of non-leaf nodes： represents spatial areas.
2) Dictionary, a Nx3 matrix (N is system defined.) consists RSS ID which accessed by users, RFV that the RSS belongs to, and Keywords recorded in RSS (includes used by users).

The Cosine R-Tree of client dynamically records RFV then carries out the vectors clustering process. The system will select a proper number of marked subtrees' centroids of the Cosine R-Tree to calculate User's Feature Vector (UFV).

2.3. Query Method

There are two Query ways (i.) by choosing category (ii.) not choosing category.

1) Query message consists of keywords are entered along with one of the following (i.) feature vector of selected category or (ii.) user's UFV. Then the information will be sent to aggregator for query.
2) RSS Aggregator sends information to Cosine R-Tree after receiving message and integrates the searching procedure of query keywords into database. The system will record the RFVs, and calculate the cosine distance between these RFVs and user's UFV (or the feature vector of the category), and store the result in priority queue.
3) The system selects some of RFVs from the shortest distance in the priority queue as the result.

2.4. Recommendation Procedures

The system selects proper number of RFVs nearest the UFV as the RVs, but some RFVs are not representative vectors and far from others which will affect the accuracy of calculation for the UFV. To provide personalized recommendation service, we adopt Genetic Algorithms to filter out improper vectors [4].

The client side builds user's dictionary and Cosine R-Tree data structure to record the information of recent user access. The system will accord this information to amend the user's UFV and calculate the RV as follow (refer to Fig.3.) [5.6]:

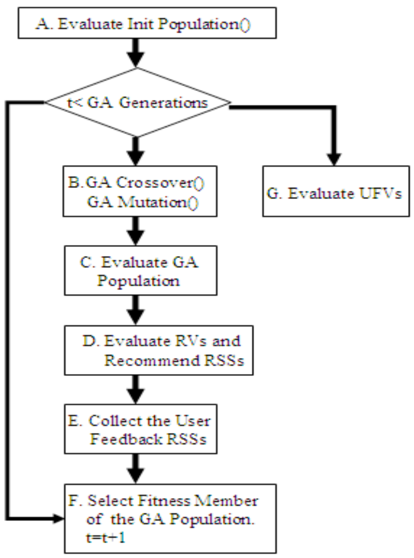

Fig. 3: The Genetic Algorithm Procedures

1) Selection: The system randomly selects keywords from the dictionary for initial population.
2) Reproduction: The system randomly selects an operator: AND, OR, NOT and query alphabetic strings which are combined by keywords in initial population. These alphabetic strings must be different and the key words in the same group cannot appeared repeatedly.
3) Crossover: These alphabetic strings are in pairs. An operator is a delimiter to mate and generate new query alphabetic strings.
4) Mutation: An operator is randomly selected among the query alphabetic string turns into another operator and creates a new query alphabetic string.
5) Fitness Function: The system sends the RVs and its RSS ID back to client, and adds the RV into client's Cosine R-Tree to calculate all $\delta(\overrightarrow{v_t})$ of subtrees in Root :

$$\delta(\overrightarrow{v_{t_i}}) = \frac{\sum_{j=1}^{m} sim(\overrightarrow{v_j}, \overrightarrow{v_{t_i}})}{m},$$

$\overrightarrow{v_t}$: centroid of subtree, m: all RV in one subtree.

As result, there will be n values:

$\{\delta(\overrightarrow{v_1}), \delta(\overrightarrow{v_2}), \ldots \delta(\overrightarrow{v_n})\}$. n: all subtrees

For those subtrees which have the max $\delta(\overrightarrow{v_t})$, the system will eliminate all of its RVs. Repeat the above steps until it fulfill the stopping criterion.

6) User's feedback: The RSS feed generated by the above steps will be recommended to users as the personalized information recommendations. Others RVs will be abandoned.
7) The system will select a number of centroids based on the $\delta(\overrightarrow{v_n})$ as the basis of UFV amendment after completing calculation.

3. Deployment

This paper precludes the server architecture of the content providers. The basic functions of RSS Aggregator are as follows: (refer to Fig.4.).

1) RSS collector responses for collecting RSS offered by content provider.
2) Users must register before using the services.
3) Cluster Service is responsible for classifying RSS feed collected by RSS collector, establishing RFV and dealing with user queries from web service.
4) VSM is saved in memory for quicker response, and RSS information is saved in knowledge pool.
5) RSS service supplies users with RSS information.

The coordination process of DKPS is as followings (refer to Fig. 5.):

1) Content provider supplies suitable resources to RSS Aggregator via RSS mechanism.
2) Through client tool, users can handle information query at RSS Aggregator.

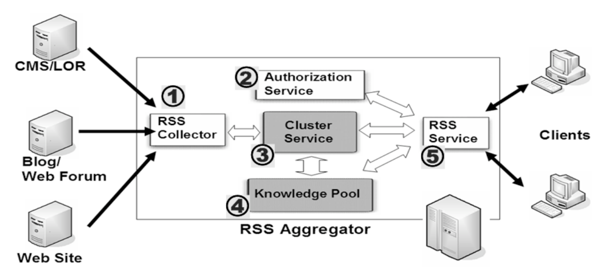

Fig. 4: Server Side Components

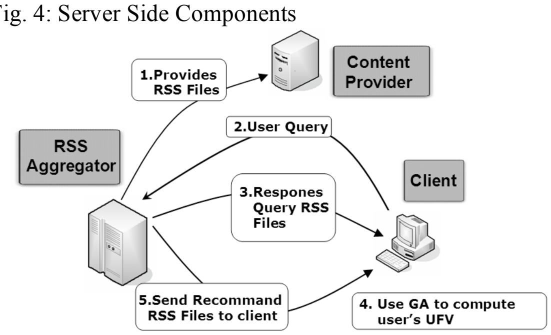

Fig. 5: Coordination Process of DKPS

3) RSS Aggregator computes the result and responds to the users for reference.
4) Client constructs UFVs to describe the user's feature by Genetic Algorithms based on the local data structure.

5) RSS Aggregator can receive some RVs from client, and return the RSS feeds of RVs to client as the recommendation.

4. Experiment

In this section, we evaluate the performance of our system through experiments with our prototype. Our experiments use 1000 RSS feeds in RSS Aggregator with all users within the range of 40. In all experiments, we calculate the information and repeat each calculation every five trials. The system develops 10 recommendations and send to the clients for users' referencing and evaluating, 1 for very interesting, 0.75 for interesting, 0.5 for indifferent, 0.25 for irrelevant, and 0 for very irrelevant.

In the first simulations, we use 2 servers to precede two tests in LAN environment. The first server uses linear scan method to pick up records which users deposit and withdraw (divide 5 time slots, each time increases 200 records). The second server uses Cosine R-Tree but without Genetic Algorithms to analyze RFV and UFV. Fig. 6. shows the effective rate of the two query methods as well as conducts comparative analysis. As expected, when the system uses the method of Cosine R-Tree (with non GA), it is more effective than the linear scan method when using the keywords to inquire information, and the better effectiveness is considerable while content increased.

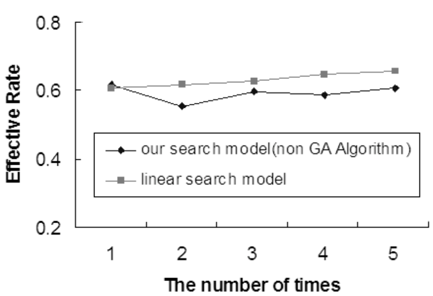

Fig.6: Comparison of Liner Search Model and Our Search (Non GA) Model.

In the second simulations, we only use the second server to precede tests. Users query information at their own choices in the same field and read 5 RSS feeds as their initial profiles. After finishing the first calculation (without Genetic Algorithm), we insert additional RSS feeds into the aggregator and generate 10 recommendations, and repeat this process for four more times. Subsequently, under the same condition with Genetic Algorithm, the system calculates and repeats each calculation for five times, and also produces 10 recommendations. We compare the effective rate using the two methods after the fifth calculation (refer to Fig. 7.), and find that Genetic method is more effective than non Genetic method while information resources are increased.

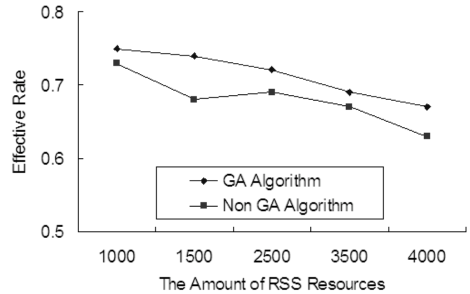

Fig. 7: Comparison of GA Recommendations Model and Non GA Recommendations Model.

5. Conclusion

In this paper, we have presented a model and system to help people construct their own knowledge base to satisfy their ubiquitous needs. The guiding principles for our solution are to integrate RSS mechanism to construct a distributed knowledge pool system. It allows users to query, subscript and recommend information resources. From the experiences we gathered in building and evaluating our system, and discover a number of performance optimization such as improving users' needs. We plan to design and integrate other methodologies to further improve the system's performance in the future.

6. References

[1] Alavi. M., and Leidner. D., "Review: knowledge management and knowledge management systems: conceptual foundations and research issues," *MIS Quarterly*, 25(1), pp. 107-136, 2001.

[2] Stephen Downes, An Introduction to RSS for Ed ucational Designers," *http://www.downes.ca/files /RSS_Educ.htm*,November 2002

[3] Tsung-Hung Chen and Yung-Hao Huang, "Research on the Application of Incremental kNN algorithm to P2P e-learning system", *Journal of Internet Technology*, pp 157-162,2005

[4] Tiffany Y. Tang and Gordon McCalla, "Smart Recommendation for an Evolving E-Learning System," *In 11th International Conference on Artificial Intelligence in Education (AIED' 2003)*. July 20-24, Australia. 2003. pp. 699-710.

[5] Robert Entriken, Siegfried Vössner, "Genetic Algorithms with cluster analysis for production simulation," *Proceedings of the 1997 Winter Simulation Conference*, USA , 1997

[6] Z. Z. Nick and P. Themis, "Web Search Using a Genetic Algorithm," *Journal of IEEE Internet Computing*, pp. 18-26, 2001.