International Journal of Security and Its Applications
Vol. 10, No. 6 (2016) pp.304-314
http://dx.doi.org/10.14257/ijsia.2016.10.6.29

An Intelligent Application Platform for Campus Life Based on Mobile Internet

Guanlin Chen[1,2], Peng Niu[1] and Hui Yao[1 *]

[1]*School of Computer and Computing Science, Zhejiang University City College, Hangzhou, 310015, P.R. China*
[2]*College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, P.R. China*
**Corresponding author email: yaoh@zucc.edu.cn*

Abstract

With the rapid development of mobile Internet, the smartphone has become an essential part of campus student everyday life. In this paper, an intelligent application platform for campus life based on mobile Internet (iCollege) is proposed. Using the collaborative filtering recommendation algorithm, iCollege aims at providing smart recommendations based on daily study demand and social entertainment of campus students.

Keywords*: Campus Information; Collaborative Filtering; Recommendation System; Mahout*

1. Introduction

Recently intelligent machine has an increasing market share, which is the highest among college students. That calls for a mobile terminal application of the college campus that could process college students' daily needs.

Some campus mobile terminal developers release mobile applications based on the different campus. For example, a campus information release platform based on Android framework is designed and realized [1]. Based on hybrid mobile application technology running on Android smartphone and Blackberry, a mobile learning application is introduced [2]. Another mobile digital campus middleware is designed based on the digital campus applications to obtain information from the original information system [3]. To provide a practical solution to access School Dean, libraries, *etc.*, a wireless digital campus system based on Android is developed [4].

However, after analyzing the campus applications, we found that existing research mainly focused on the construction of the campus network platform and virtualization building. These APPs mainly focus on certain school and can't connect with other universities. Designers just take classmate convenience into consideration, without thinking about national college students' needs. What's more, those applications' functions are relatively simple, which can't extend their market share through the users' stickiness.

To solve above problems, the campus application should be more customer kindly and effective, giving students more practical information services taken into account their needs.

Based on mobile Internet, this paper designs a campus application platform, iCollege, which is consisted by a distributed cloud server and intelligent mobile terminal (Android system).

[*] Corresponding Author

ISSN: 1738-9976 IJSIA

The iCollege uses Java as mobile client development and server-side language, MySQL as a distributed database. Combing with the collaborative filtering recommendation algorithm, iCollege could realize recommendations based on daily study demand and social entertainment.

2. Overview of iCollege

2.1. System Requirements

Gathering school information is the basic function of iCollege, which can rapidly crawl on the campus network to collect published information, school announcement, campus emails and so on. Three kinds of technology are utilized to construct our system: Internet information gathering, intelligent information processing and full-text indexing technology. According to users' behavior, different original log summaries are recorded in the session log, and each of them stands for a user request and the corresponding service.

By utilizing open source Mahout framework, we design and realize a college campus activity recommendation system.

The system mainly consists of four modules, which are shown in Figure 1.

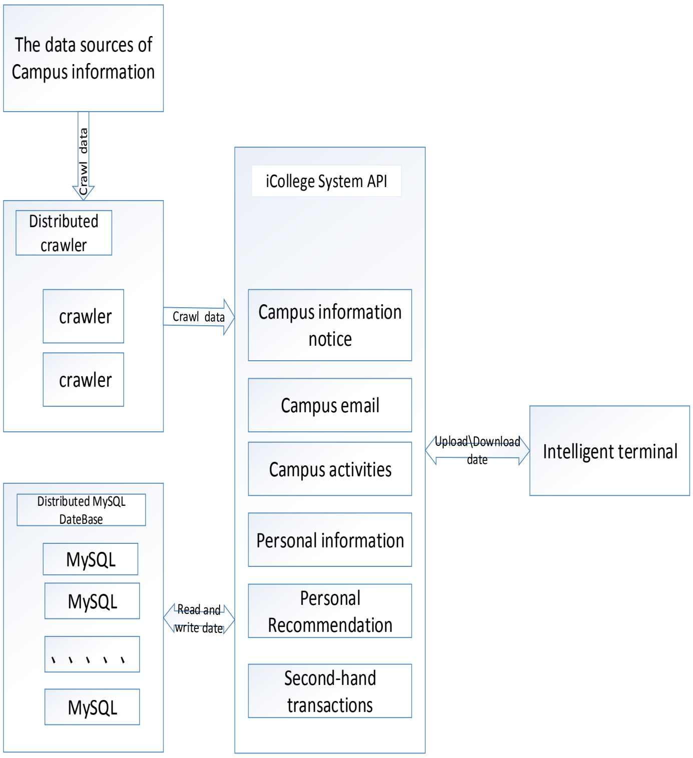


Figure 1. The Basic Framework of the System Functions

• Distributed Web Crawler module: this crawler is vertical [5]. Each crawler grabs a peculiar kind of campus information, following a uniform interface, which is beneficial for crawler system's unification and the expansion.

• System API (Application Programming Interface) module: this module applies API with the format of JSON, which is based on C/S model. System use API to execute communication between server and mobile terminal. Based on API, each module can read and process data easily.

- Distributed MySQL (non-relational database) database module. Considering that system’s target groups are Chinese college students, which are very huge. We adopt MySQL, a distributed file storage system, as the background database.

- Intelligent terminal service module. By adopting API interface module, system can display server stored contents and upload relevant information, providing campus information and responding recommendation to users.

2.2. Process Design of the System

At first, the system judges whether an intelligent terminal has connected. When the user begins to download and install the application from different platforms, an initialization navigation page will be displayed on the terminal. After registration, the system will assign a SessionId to the corresponding terminal, which would be saved on the server for five days.

After log-in, Speed Dial standing different functional category would be shown on the initialization page. When the user sends an order, for example, reading school information and campus announcement, the crawler will grab information from each data source directly; when the users want to know information about activities, data will be conveyed by the MySQL database.

Figure 2 shows the guiding Speed Dial page.

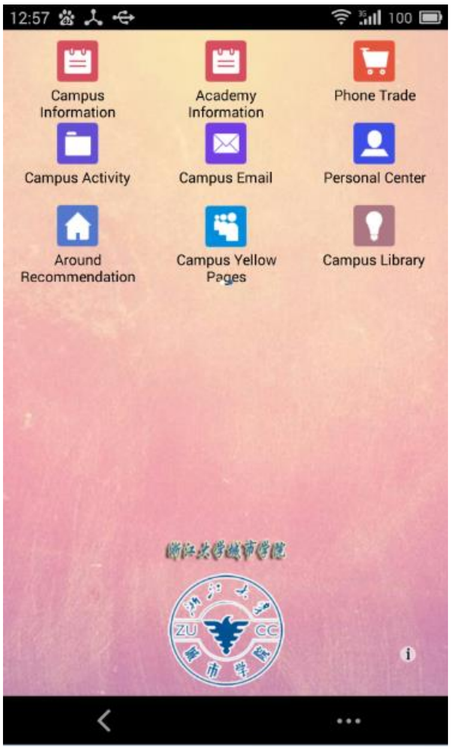


Figure 2. The Guiding Speed Dial Page

3. System Design and Realization

This system adopts distributed designing, Java as the development language for background and mobile terminal system. As for part of the campus information crawler, we use HttpClient to gather campus information and transfer them to the mobile terminal by JSON. What’s more, MySQL data storage has been built to storage campus activities and trade information. Due to the rapidly expanding users, a recommendation system built by explicit feedback data model with user behavior feedback data structures has been introduced.

3.1. Design of Distributed Data Crawlers

In order to increase scalability and maintainability, all web crawlers have been modularized and core work (such as a source of information classification, crawl content into a JSON format, intelligent terminal and server data transmission, etc.) has been encapsulated. Developers can add new modules to crawler content easily. In order to enhance flexibility, HttpClient offered by Apache has been imported which could capture content directional according to different campus web structure.

The concrete realization code is as follows.

```
    String url = (String) request.getParameter("category");
    String i_artid = (String) request.getParameter("i_artid");
    url="http://jsxy.zucc.edu.cn/module/permissionread/"+category+"&i_artid="+i_ar
tid+"&ptype=1";
    String[] pageInfo= catchPage(url);
    String errmsg= pageInfo[0];
    if( errmsg == null || errmsg.equals("")){
    String content= pageInfo[1];
    Document doc = Jsoup.parse(content);
```

As shown above, a standard campus information crawler module consists of two parts, fetching the contents of the specified directory and converting them into JSON format. Before generalization of this platform, almost each campus has established partly information platform. So this program just analyses existing data sources page structure and fetches a particular part.

In this case, category stands for type of data source when the user visits the intelligent mobile terminal, i_artid represent information number under a specific category, and ptype is on behalf of data access. URL stands URL parameter, and is a combination of category, i_artid, ptype and some fixed content combination.

After content has been grabbed from URL website, it should be packaged into JSON version. Detailed package method is 3rd party libraries, which mean the developer only needs to switch content grabbed by parse() method.

What should be pointed out that, key part of crawler is analyzing web structure and fetching data. Thus this process might fail after campus web page structure updated.

3.2. Background System Design

The iCollege platform is designed based on B/S model, MVC structure [6] and JavaEE framework. The main functions include data transmission and processing between intelligent terminal and the server.

The background system integrates data processing and data analysis. All operation should be done through interaction with the server background. Crawlers are not exposed to the database, which is completed by interaction between the server and terminals. Crawlers only need to submit the data to each mobile terminal. Crawlers don't know which data users concern, and which data have been read. All these works are executed by the End-user. This design could allow divisions of responsibilities, and each module only focuses on their own job.

To facilitate post-expansion, the system uses the API in popular JSON format for interaction between different parts. For security, system use MD5 [7] to encrypt passwords when users login. Passwords stored by server-side database are character that encrypted by MD5. After verification, system will assign a session for the user, which could stores user’s ID and name. Meanwhile, a SessionId would be sent to the client, which is defaulted by the system as the SessionId. The server background will save the session for 5 days. By holding this SessionId and related information, clients are able to call the server-side API.

The registration and log-in code is as follows.

```
.#Mobile terminal using MD5 encrypted transmission mode
netUtil.add("uname", name);
netUtil.add("upassword", MD5.digest(password));
http = netUtil.getHttp();
netUtil.post();
#If the password is incorrect, the system returns the following JSON data:
{
 "response": error
 }
#If the password is correct, the system will return to the user the correct
representation and a SessionId, as follows:
{
 "response": login,
"SessionId": "3011240"
}
```

3.3. Terminal System Design

The iCollege terminal is one of the most important parts of the whole system, which is service system for users at various universities. Through different mobile devices, users are able to run over campus news and announcements, campus Yellow Pages, the latest information about campus events, secondary trading, as well as send and receive emails.

The platform uses C/S architecture based on application platform API and adopts SOAP-based Web Service to interact between server-side API. Using some third-party tools such as network modules xUtils, the system could execute web model transmission and image cache mechanism between intelligent terminal and the server.

(1) Campus notices information

Each campus has the own original data source, so the intelligent terminal display for each campus crawlers crawl different data source. This data section is divided into two blocks: the school notice and branch notice. Instead of reaching the database, crawlers only need to submit data to respective intelligent terminals.

For example, to get international education information (we assume that users have logged in), through calling the API, mobile terminal should submit acquired account "SessionID" value and branch category "category" as parameters.

(2) Monitor mechanism

To maintain receiving and sending message between terminals and the server, the system adopts SOAP protocol based on Web Service [8]. Each registered intelligent terminal will have a thread, which connects to the server by a SessionId. The client could execute a communication with the server regularly. If the user has unread messages, this message will be carried out in terminal push notifications.

Mobile terminal may be either offline or online, so when messages transferred between users, the message is transmitted to ServletContext, listener trigger, by which to judge whether users online. If online, the message is written to the client and deposited into the unread message database. If offline, then the message is deposited into the library unread messages, and will be displayed at the terminal when users get online.

(3) Campus activities recommendation

With the iterative development of the system and increase of user group, the system constructs activities recommending mechanism based on collaborative filtering algorithm in the background. Campus activities recommendation mainly uses session logs which collect user history behavior. The system also uses the explicit behavior feedback mechanism, which would be one of the most important features in the future. Users can view and participate in activities recommended by the terminal based on personal activities record log.

Recommendation engine is divided into two parts, offline data analysis (main part) and online recommendation. In order to process huge amounts of data log, this part is built based on Hadoop platform. We use an algorithm library offered by Apache open source Mahout [9], which includes a complete set of recommendation engines: the package of "collaborative filtering" algorithm, and implemented parallelization and a simpler API interface. Completely encapsulates the "collaborative filtering" algorithm, and implements the parallelization, provides a relatively simple API interface.

The construction of recommendation system is shown in Figure 3.

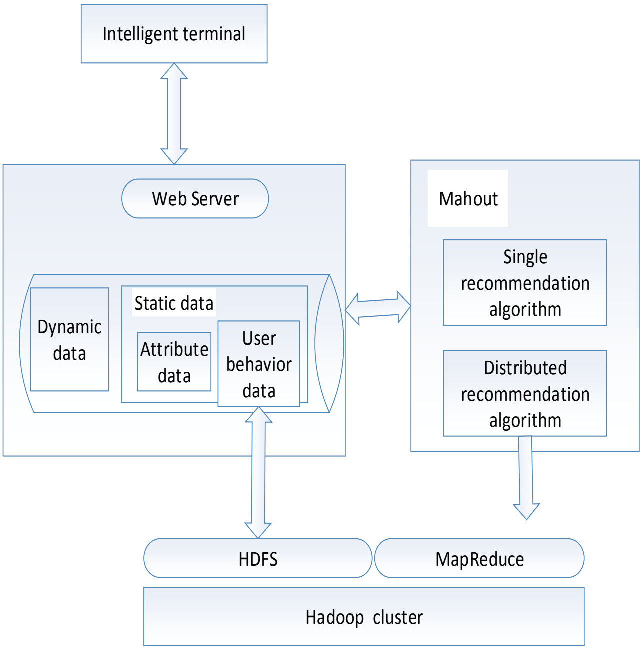


Figure 3. Construction of Recommendation System

Mahout divides recommendation algorithms into four steps: construct user vector, construct a synergistic matrix, matrix multiplication, resulting effect is recommended. Each stage of the algorithm is derived from a series of Mapper and Reducer's derivate. Development work is to achieve the Map function and Reduce functions subclass Mapper and Reducer classes.

When using the distributed algorithm, the system uses the Euclidean distance to measure similarity of the user. Each user's preference is stand by an n-dimensional vector, and value of each dimension represents preference of an activity: if value equals to 0, means this user doesn't have interest on this project; 1 stands for user is like this activities and participated into that. This vector is sparse, because in reality, users are only interested in few projects, so most of vector value is zero.

For example, a user i vector is $[\mathrm{r}_{i1}\ r_{i2}\ \cdots\ \mathrm{r}_{in}]^T$. In the implementation process, this stage takes example by traditional non-collaborative filtering, calculating and predicting ideology. Based on the algorithm implemented on Mahout, the user's participation and rating in all activities have been scanned firstly. And then compute the average score of each activity, which is $R=[\bar{R}_1 \ \bar{R}_2 \cdots \bar{R}_n]^T$, where $\bar{R}_{\mathrm{i}}$ is the i item (i = 1, 2, ..., n) average score. For each activity, subtract the user's score by an average score, and write that data into user score vector. Users vector format is User ID [Activities ID1: score difference, activities ID2: score difference, ..., event ID n: score difference].

In realization of distributed implementation of the project based on collaborative filtering algorithm in Mahout, synergistic matrix should be used, of which rows and columns of the matrix are active.

Values in the matrix are Synergistic Factors of rows value and the respective columns. Each of them stands for the number of simultaneous users who like the respective activities. Thus we can form a collaborative matrix. Assume there are n projects, similarity table is shown as Table 1.

Table 1. The Synergy Matrix

	Activity 1	Activity 2	···	Activity n-1	Activity n
Activity 1	8	3	···	2	0
Activity 2	3	3	···	5	6
···	···	···	···	···	···
Activity n-1	2	5	···	4	2
Activity n	0	6	···	2	2

In Table 1, the Synergistic Factor of activities 1 and the activities 2 is 3, showing that Item 1 and Item 2 are simultaneously liked by three users. If the Synergistic Factor is 0, means that there aren't any people like those activities at the same time.

The implementation code is as follows.

```
# Build map function of collaborative matrix
Iterator<Vector.Element> uItemPre =u Vector.get().
iterateNonZero();
while(uItemPre.hasNext()){
int iIndex1 = uItemPre.next().index();
Iterator<Vector.Element> iPre = uVector.get().
iterateNonZero();
while (iPre.hasNext()) {
 int iIndex2 = iPre.next().index();
context.write(new IntWritable(iIndex1),
new IntWritable(iIndex2));
}
 }
```

First and second stages construct all data needed for recommendation, and the next stage is a matrix of multiple users' vector and synergistic matrix, just like shown in Equation 1.

$$\begin{pmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{m1} & \cdots & t_{mn} \end{pmatrix} \bullet \begin{pmatrix} r_{i1} \\ \vdots \\ r_{in} \end{pmatrix} = \begin{pmatrix} R_{i1} \\ \vdots \\ R_{in} \end{pmatrix} \tag{1}$$

Equation 1 stands for recommendation result for user 1, giving first k recommendation through rank R_{i1} and R_{in}.

In real application, the system uses Equation 2 as calculating method to enhance efficiency and accuracy.

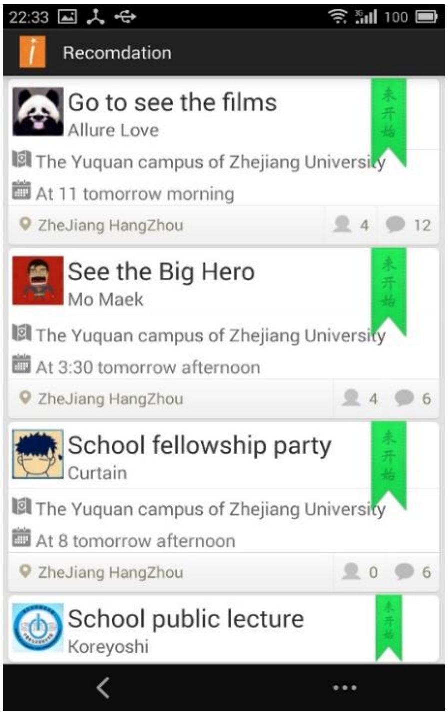
$$\begin{pmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & \ddots & \vdots \\ t_{n1} & \cdots & t_{nn} \end{pmatrix} \bullet \begin{pmatrix} r_{i1} - \bar{R}_1 \\ \vdots \\ r_{in} - \bar{R}_n \end{pmatrix} \bullet \frac{1}{\sum_{k=1}^{n} \mathrm{t}_{j,k}} + \begin{pmatrix} \bar{R}_1 \\ \vdots \\ \bar{R}_n \end{pmatrix} = \begin{pmatrix} R_{i1} \\ \vdots \\ R_{in} \end{pmatrix} \tag{2}$$

The j stands for the number of lines in synergistic matrix. Map function line method is needed here: input for the matrix is the out of map function packed by collaborative matrix column and map function for user preference value decomposing.

The implementation code is as follows.

```
# Realization of user's score divided map function:
long uID = key.get();
  Vector uVector = value.get();
  Iterator<Vector.Element> it = uVector.iterateNonZero();
    IntWritable iIndexWritable = new IntWritable();
    while (it.hasNext()) {
     Vector.Element e = it.next();
      int iIndex = e.index();
      float preferenceValue = (float) e.get();
     iIndexWritable.set(iIndex);
    context.write(iIndexWritable,
    new VectorOrPrefWritable(uID, preferenceValue));
  }
# Realization of Map function for synergistic matrix multiplication
  Vector coocColumn = vAndPrefsWritable.getVector();
  List<Long> uIDs = vAndPrefsWritable.getUserIDs();
  List<Float> prefValues = vAndPrefsWritable.getValues();
  for (int i = 0; i < uIDs.size(); i++) {
   long uID = uIDs.get(i);
   float prefValue = prefValues.get(i);
   Vector partProduct = coocColumn.times(prefValue);
   context.write(new VarLongWritable(uID),
     new VectorWritable(partProduct));
    }
# Realization of Reduce function for synergistic matrix multiplication
Vector partVector= null;
for (VectorWritable vectorWritable : values) {
partVector = partVector == null ? vWritable.get() :
partVector.plus(vWritable.get());
}
 context.write(iId, new VectorWritable(partVector));
```

After we got preference data, and rank this preference, the last recommendation page is shown as Figure 4.



Go to see the films
Allure Love
The Yuquan campus of Zhejiang University
At 11 tomorrow morning
ZheJiang HangZhou 4 12

See the Big Hero
Mo Maek
The Yuquan campus of Zhejiang University
At 3:30 tomorrow afternoon
ZheJiang HangZhou 4 6

School fellowship party
Curtain
The Yuquan campus of Zhejiang University
At 8 tomorrow afternoon
ZheJiang HangZhou 0 6

School public lecture
Koreyoshi

Figure 4. The Recommended Effect of Campus Activities

4. Conclusions

In this paper, an intelligent campus application platform, iCollege, is constructed. The iCollege supplies the information channel for different undergraduate students in mobile terminal, including campus notification, receiving and sending emails, trading and activities recommendation based on the user's preference. However, since this program still stays in iteration development and extension stage, the number of session log is relatively small. What's more, Mahout simplifies distributed collaborative filtering algorithm, which is simpler than non-distributed algorithm to calculate the similarity, as well as the final score prediction processing method. Thus accuracy of recommendation of local campus activities and trading still has a great space to improve, as well as the user's satisfaction.

In the future, with expansion of register and activity logs, and realization of Mahout distributed collaborative filtering algorithm, this limitation could be broken through and accuracy of recommendation could be further improved, which increase value of this system and the degree of customer's satisfaction.

Acknowledgements

This work is partially supported by the 2014 National Undergraduate Training Programs for Innovation and Entrepreneurship, China (No. 201413021015) and the Hangzhou Science & Technology Development Project of China (No. 20140533B13).

References

[1] J. Wang, X. Yu, Y. Zeng and D. Yang, "The design and realization of campus information release platform based on android framework", Proceeding of the 16th National Conference on Computer Engineering and Technology, Shanghai, China, **(2013)** August 17–19.

[2] D. H. Setiabudi, L. J. Tjahyana and Winsen, "Mobile learning application based on hybrid mobile application technology running on Android smartphone and Blackberry", Proceeding of the International Conference on ICT for Smart Society 2013: "Think Ecosystem Act Convergence", Jakarta, Indonesia, **(2013)** June 13–14.

[3] Y. Zhao and L. Cheng, “Design of data middleware for mobile campus based on OSGi and RESTful”, Journal of Huazhong University of Science and Technology (Natural Science Edition), 41, SUPPL.2, **(2013)**.

[4] L. Ma, Y. Zhou and K. Liao, S. Liu, J. Qiao, Z. Han and J. Wang, “Development and research of digital campus system based on android”, International Journal of Smart Home, vol. 8, no. 4, **(2014)**.

[5] E. Uzun, E. S. Güner and Y. Kiliçaslan, “An effective and efficient web content extractor for optimizing the crawling process”, Software - Practice and Experience, vol. 44, no. 10, **(2014)**.

[6] C. Liyan and G. Qing, “Research on framework developing technology based on MVC”, Advances in Information Sciences and Service Sciences, vol. 3, no. 3, **(2011)**.

[7] S. Chen and C. Jin, “Research on the multi-message modification techniques on MD5”, Journal on Communications, vol. 30, no. 8, **(2009)**.

[8] Y. Wu, H. Dong and H. Wang, “Software project march objective real-time monitor mechanism design”, Information Technology Journal, vol. 12, no. 20, **(2013)**.

[9] S. K. Thangavel and P. Swati, “Customization of recommendation system using collaborative filtering algorithm on cloud using mahout”, Advances in Intelligent Systems and Computing, vol. 321, **(2015)**.

Authors

Guanlin Chen,

Born in 1978, Ph.D., professor, chenguanlin@zucc.edu.cn. His main research interests include E-government, computer networks and information security.

Hui Yao,

Born in 1981, B.E., lecturer, yaoh@zucc.edu.cn. Her main research interests include communication engineering and student management.