Final Report

Course: IS2560

Group Members: Rui Bi, Wentao Jiang

Instructor: Michael B. Spring

Abstract

Nowadays social network has been extremely popular in varies of areas and its application bring tremendous convenience to the real life. Our final project focused on developing a website that can help looking for group project partners and selecting projects of interest. For those who have had initial ideas, the system can serve as a helper gathering classmates. On the other hand, for those who have difficulty finding the proper projects, the website can provide suggestions and recommendations. Web technologies, including html, javascript, ajex, serverlet, css, database, and so on has been applied in this project, as well some elementary concepts and algorithms of information retrieval and machine learning.

Introduction

The nature of complexity of information technology projects decides that teamwork is inevitable and of great significance. iSchool at University of Pittsburgh has always been training students with the ability of cooperation by the form of team projects. However, since the size of class has been continuously expanding, the traditional way of grouping suffers from inefficiency. There exist severe difficulties to get familiar with everyone in the class, which lead to the problem when choosing partners. How can one get involved into the project he is most interested in? How can one find partners who are the best fit for different projects? Bearing this in mind, we decided to build a website on which students can either post their ideas of projects or see the projects recommended to them according to their interests.

External Functions

1. Register and Login

Since the website can provide customized recommendations of projects, the used need to create an account before fulfilling of functions. We designed a neat login interface with user name and password. Servlet and database are used here to verify the identity of user who already has an account. For new users, a registration form needs to be filled and submitted to create an account. Ajax elements have been used for the users to choose the interests.

2. Post Project Proposals

After logging into the system, users can submit proposals of projects. Till now, only text is supported in the input system. These proposals are stored and will be recommended to users according to their personal preference and interests.

3. Project Recommendations

On the homepage, there are three dropdown selection boxes for users to choose their interested technologies or areas, which will serve as the basis of recommendation. Three boxes are designed to hold three keywords of interest with descending order. After filling these boxes, recommended projects will be displayed on the page. The recommended documents can be previewed on the web page.

Infrastructures

JSP, servlet and database are combined to form the infrastructures of the system. All user information is saved in database once submitted by users. When user submits a proposal, the information associated with the proposal will first be stored into the database. Meanwhile, java programs will be triggered to process the proposal, extract the key characteristics of the proposal and save it separately for later use. Furthermore, since the recommendation of projects is based on scoring principles, the addition of a proposal will also lead to the update of data in the database that is used to calculate the matching scores.

When a user would like to see recommended projects, the system reads in the three keywords from the dropdown selection box, passes the keywords to the database which will trigger the matching process of the keywords to each proposal that exists in the database. After calculation, every proposal will be assigned a score based on the user inputs, and the proposals will be sorted with the scoring from high to low and then displayed on the webpage to the user.

During the process, the database can collect data and update automatically according to the data received from users, and adjust the recommendation accordingly. There are mainly four tables to record all data. They respectively are:

1. UserInfo: holds the users' information. It is composed of userID, username, password, first name, last name, gender and three tags chosen (in order).
2. ProjectInfo: stores the list of projects containing projectID, project name, date posted, date expired, user who posted the project, and the proposal content.
3. RecmdInfo: has the recommendation information for users. It project multiple pid to each user id, and store the grade for each proposal for specific user. It has four column including userID, projectID, matching score.
4. TechTagInfo: the storage of all the keywords, including document frequency, term total frequency and calculated matching score for each keyword. It is the bases for matching score calculation.
5. KeywordInfo: has a list of keywords found in proposals. Multiple keywords may project to the same proposalID. Also, the frequency of keywords showed in each proposal is recorded in this table.

Algorithms and Scoring Model

1. Parsing method

When proposal is submitted and taken into the database, it will be parsed first. The parsing process starts from converting letters to lower case and breaking the paragraphs into words. The words parsed will be saved in a hash table with words as the key and frequencies of words appeared in the proposal as value. After the entire proposal document is parsed, a hash table containing all wording information is generated accordingly. Since we've saved all keywords as a list in the database, the key in the hash table of proposal will be compared with the keywords list. If a word is contained in the list, it will be selected and saved separately including the frequency. In this way we can extract the relevant information. On the other hand, the frequencies of keywords in all proposals and frequencies of documents that contain certain keywords are also updated and saved for later use of matching calculation.

2. Scoring

For the matching of user interests to proposal, we used a basic scoring model --- Tf-idf weighing model. tf here stands for term frequency, which in our case is the total frequency of a keyword showed in all proposals. df represents the document frequency, which is the number of proposals in the collection that contains a specific keywords[1]. In order to decrease the overwhelming effect of high frequency keywords when counting the scores, the weight for different keywords needs to be adjusted. As the model of tf-idf, idf is defined as

$$\mathrm{idf} = \log \frac{N}{df}$$

where N is the number of proposal in the collection, and weighting scheme assigns a weight to t as

$$\mathrm{tf} - \mathrm{idf} = \mathrm{tf} \times \mathrm{idf}$$

We also taking the 1st keyword selected by user as a higher priority, and has greater weight than the 2nd, and the 2nd than the 3rd (with a coefficient of 5 for 1st, 3 for the 2nd and 1 for the 3rd). The assignment was tentative and could be easily adjusted.

Future Work

1. Get rid of remaining bugs. The proposal upload process sometimes will generate an error. The reason has not been fund due to time reason. There are other bugs that need to be fixed.
2. Add more elements for the scoring process. We are considering adding course history, resume, etc. to improve the accuracy of scoring. With longer descriptions of user, the vector model could be used to calculate the angle between the proposal and users' resume or other documents. Furthermore, more machine learning technique could be used such as filtering stop words, training words and so on.
3. Improve the webpages. Add more functions to the webpage to improve the user experience. Establish friendship between users, found a group and a lot of other improvements can be done.
4. Allow free input of keywords by users, and update the keyword list accordingly. In that case, the database can complete all the collection process as users submitting information, and automatically include the new information for later recommendation.

Reference

[1] Introduction to Information Retrieval, C. D. Maning, P. Raghavan, H. Schütze. Chambridge University Press. 2008.