WolfPal: Feedback and recommender system for course selection

Neel Kapadia
North Carolina State University
Raleigh, NC, US
ntkapadi@ncsu.edu

Rohan Chandavarkar
North Carolina State University
Raleigh, NC, US
rgchanda@ncsu.edu

Sainag Shetty
North Carolina State University
Raleigh, NC, US
sgshetty@ncsu.edu

Rohit Naik
North Carolina State University
Raleigh, NC, US
rtnaik@ncsu.edu

ABSTRACT

In today's competitive world students need to take certain decisions in their academic life which seem to be easy or trivial but have a major impact on their future. Choosing the right courses in their degree program at the right time is one such decision making task which is of great relevance to a student's academic and professional future. Apart from contributing to the grade point average and shaping up the resume, the course selection also affects the qualitative factors like the student's interest in the degree program. Since the decision of coming out with a proper course plan directly affects the career of the student, this decision should be a well informed one. The students rely on various sources and use multiple techniques to find out the best courses for them which fit in their credit-based study. We thought of contributing to one such platform which brings together all such resources and help the students in making this course plan. The platform has the functionalities of interacting with peers and discuss important points like professor reviews, average grades, course structure. We plan to add new functionalities in the platform using latest techniques like natural language processing so that the factor of student's interests and skill-sets is considered more strongly in the course planning. A student's resume will also be considered for suggesting the right courses.

Keywords

Course Feedback, Course Recommendation, Keyword Extraction

1. INTRODUCTION

The task of planning the courses is one of the most important tasks in a student's life. Students rely on a lot of resources which are scattered all around and not all resources give the students a perfect strategy to choose the right courses for them. The task of course selection is taken seriously by students all over the globe and it is necessary that all the resources are utilized in conjunction to give a course plan to the students which will be beneficial for them.

There are hundreds of universities in North America and huge number of students pursuing advanced degrees in these universities. During the application phase, the students select certain universities based on the area of specialization and the courses offered in that field. So it would be unfair for the students if they do not get the right courses after taking admission due to lack of resources or lack of communication. Wolfpal, a course planning assistant, is a platform which aims to resolve these shortcomings and come up with a good course plan for the students. We plan to add new functionalities to this novel approach of course planning by applying latest techniques available.

1.1 Existing Functionalities

Wolfpal is a highly efficient platform which has important functionalities like chatting with peers about the course reviews, average grades of previous batches, course structure. These parameters are judged accurately by experience of similar people and a chatting platform for the same serves the purpose. Moreover, the platform also has a bot functionality where in the students can get answers to certain queries related to planning of courses. The system also provides a list to the students for the course plan and also displays course information from the course catalog to the students.

1.2 New Functionalities

The main aim of our new functionalities is to capture maximum student's interests and skills to suggest them a course plan. We are planning Natural Language Processing techniques for the same.

We will ask the student to describe interests about the courses and the area of interest in a textual summarized form. Using NLP techniques, we will extract relevant keywords from the description of interests and map it keywords of courses so that these courses are suggested in the plan.

Similarly, a student's resume can also be used to extract keywords and skills and map it to course keywords to find a best match for that student.

We will also ask students certain questions regarding their course choices and the interests will be extracted from those answers.

2. MOTIVATION

The dilemma of deciding which course to take is a major

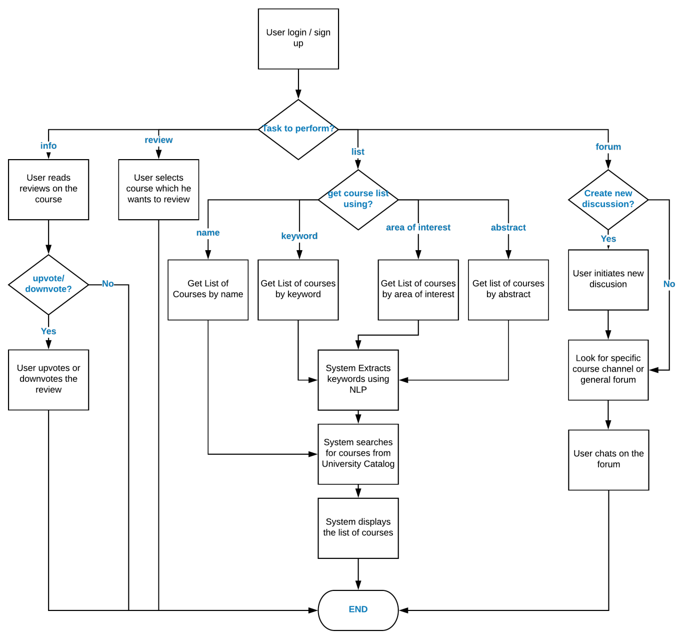
User login / sign up
task to perform?
info
review
list
forum
User reads reviews on the course
User selects course which he wants to review
get course list using?
Create new discussion?
name
keyword
area of interest
abstract
Yes
No
upvote/ downvote?
No
Yes
Get List of Courses by name
Get List of courses by keyword
Get List of courses by area of interest
Get list of courses by abstract
User initiates new discusion
User upvotes or downvotes the review
System Extracts keywords using NLP
Look for specific course channel or general forum
System searches for courses from University Catalog
User chats on the forum
System displays the list of courses
END

Figure 1: Activity Diagram

problem faced by every student. With a plethora of options and not a very reliable source of information, this task only becomes more difficult. Take the case of North Carolina State University, specifically the Computer Science stream, there are around 125 different courses. A student needs to study and find out which amongst these courses best suits his/her interest, which is a lot of work. Assuming it takes about 15 minutes to find out what the course has to offer, doing that for all of the possible courses takes 31.25 hours. Also, this is based on the condition that, all of the information about that course is readily available.

For a student, all of this is a lot of work given there is hardly any guidance. The aim of WolfPal is to reduce the effort put in by the student to do this work. The application integrates the resources provided by the university with the remarks of the students on a course, who had taken the course earlier, which will be ideal for the tackling the problem of lack of information.

3. LITERATURE REVIEW

Taking good courses is a very important necessity in every student's career. Despite steady work done related to course recommendation and feedback system, course selection remains a challenging process for any graduate student. In the following section we will discuss the previous works related to course recommendation and feedback system along with their shortcomings.

3.1 WolfPal

"WolfPal"[1] is an interactive platform that aims toward helping students with selecting courses for their graduation. It tries to integrate the resources provided by the university with the remarks of the students on a course, who had taken the course earlier which will be ideal for the tackling the problem of lack of information. Thus, it tries to give the student an idea of how and what to expect from a particular course. The system will provide a way for the student to connect to other students for sharing any information or queries about courses. The system also gives a chatbot for course recommendation.

The shortcomings from this current version of the application can be described as follows:

- Chatbot integrated into a Web GUI isn't intuitive:

  The application helps in course recommendation using a chatbot that recommends courses. But the problem here is that whilst using a Web Based GUI having a bot which does a QA is very intuitive.

- The keyword filtering of the courses is not effective:

  While the system does have the feature of filtering out courses by putting in keywords, this filtering technique is not guaranteed to produce ideal results as also noticed during testing.

- Course recommendation methodology:

  The chatbot used in the system asks the student a series of questions which helps find his/her interest. Although this method may yield student's interests, it will take a lot of the student's time

3.2 Course feedback system

"Course feedback system"[2] provides three features. One is a forum that can share short reviews for each course with like and dislike button that can increase its reliability. The second feature is a feedback system that senior students could answer eight questions for each course, such as numbers of project, number of assignment, course knowledge and so on.The third feature is a suggestion system that helps student search courses by 8 filters based on the data from second feature. We think this system is a more reliable and a more feedback friendly system than the previous one, because of its separate review and feedback, as well as grading criteria (questions) by multiple choice to prevent typing errors.

3.3 An Automated Recommender System for Course Selection[3]

This paper presents a collaborative recommender system that recommends university elective courses to students by exploiting courses that other similar students had taken. The system employs an association rules mining algorithm as an underlying technique to discover patterns between courses. The system tries to recommend elective courses to students based on what other similar students have taken. It finds similar students and then apply association rule mining algorithm on their courses to create courses association rules. Discovered courses association rules are used to get recommendation.

4. CHANGES FROM EXISTING SYSTEM

We are planning to add the following new features to the application

4.1 Community oriented rather than centralized system

We plan to add syllabus and reviews option for registered users (current or ex users). This move is towards eliminating the need of a central organization responsible for uploading and updating all the data continuously and rather having the users themselves as maintainers. We will obviously ensure certain checks such as avoiding multiple syllabus for the same year and adding features such as reputation of users (based on their reviews and number of upvotes, etc.)

4.2 Course suggestions based on keyword mapping

We plan to use NLP to extract keywords from the course summary, the syllabus and reviews. This is so because sometimes the name of the course is misleading or under representative of what the course actually is. Also, we plan to extract keywords from the user's interest description. Then we plan to match those keywords and display the corresponding matching courses with all the details of those courses. This will ensure all round information for the user.

4.3 Fetching relevant courses using topic modeling

We plan to identify topics for all the courses using the course descriptions and/or syllabus. Topic models can help to organize and offer insights for us to understand large collections of unstructured text bodies. This will make it more efficient in fetching relevant courses when the student searches for courses.

5. IMPLEMENTATION

5.1 Use Cases

1. login/signup

   This is the authentication module where the users create a new account or login to their existing accounts.

2. get list of courses by name

   The users search for a particular course by the name of the course or the course number

3. get list of courses by keyword

   The users search for courses by keyword. For example, if the user enters the keyword ”computer science” then all courses related to that keyword will be displayed

4. get list of courses according to area of interest

   The users search for courses by their area of interest. For example, if the user enters the keyword ”i am interested in data science” then related courses to the domain of data science will be displayed

5. get list of courses by abstract

   The user enters a paragraph of his past achievements, projects, courses and his interests. The system will use NLP to extract keywords from the text and display courses accordingly

6. get/give information about courses on discussion forum

   The users can provide information on the courses they attended or get information on the courses they wish to attend on the discussion forum. Basically it is a chat-based application where the users can interact and share information

7. enter details / reviews about a course

   Students can enter reviews and suggestions about a course which will be displayed along with the course information

8. extract keywords from NLP

   The system extracts keywords from the abstract the user provides

9. get courses from university course catalog

   The system get the list of courses from the university course catalog

10. display list of courses

    System displays the list of courses after mapping the keywords with the courses

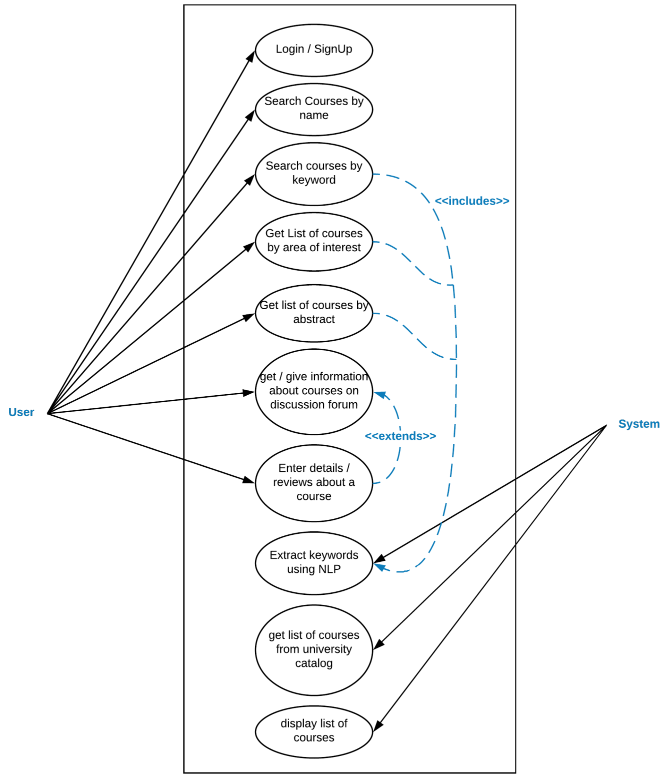
Login / SignUp
Search Courses by name
Search courses by keyword
<<includes>>
Get List of courses by area of interest
Get list of courses by abstract
get / give information about courses on discussion forum
User
<<extends>>
System
Enter details / reviews about a course
Extract keywords using NLP
get list of courses from university catalog
display list of courses

Figure 2: Use Case Diagram

5.2 Technology Stack

1. **MongoDB**

   - MongoDB[4] is a NoSQL database platform. It uses JSON-like documents with schemas. We used MongoDB for user information, course catalog, reviews of courses, chats of discussion forum etc. We used MongoDB so that we get the advantages of document oriented storage and indexing on any attribute.

- One document for each user help us cleanly maintain all the information about the users in one place without the need of joins to get the data.
- Using such a database gave us the advantage of easily adding entries from NodeJS and accessing that entry using a Python script. Schema-less JSON-like documents made it easy for us define relationships between users, courses and tasks.

2. **Mlab.com**

- mLab is a fully connected, managed cloud database that hosts MongoDB databases.
- It has partnered with platform-as-a-service providers and uses AWS, Google and Azure. As we have stored all users' data on MongoDB, we wanted a platform which is reliable and secure. mLab.com took care of the security of the details of the users. Using a platform like mLab.com ensures BASE properties of a NoSQL database, secures data on MongoDB and makes the database available online.
- Using such a database gave us the advantage of easily adding entries from NodeJS and accessing that entry using a Python script. Schema-less JSON-like documents made it easy for us define relationships between users, courses and tasks.

3. **Python(nltk)**

- Python is powerful and fast, plays well with others, runs everywhere, is friendly & easy to learn, is open source[5]. Python provide plethora of libraries which can be used to solve almost every problem. Also, it is an easy to use and easy to learn language.
- The primary reasons for selecting python to develop the core algorithm were:
  (a) suitable libraries available
  (b) team comfort with python
  (c) easy compatibility with Node.js.
- We used this language for natural language processing using the python nltk library. The Natural Language Toolkit (NLTK) is an open source Python library for Natural Language Processing.[6]

4. **Javascript**

- We are using javascript as a scripting language between the front-end and back-end.
- Javascript is used for validation of forms.
- JavaScript, often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content engineering.[7]

5. **Ruby on Rails**

- Our system uses Ruby on Rails under the hood which provides ease of development using MVC architecture. The vast available list of Ruby gems (similar to Java packages) helps with the development of a lot of features such as Authentication and Forum, with ease.

6. **Bootstrap**

- Bootstrap is a free front-end framework for faster and easier web development.
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins[8]
- Bootstrap also gives you the ability to easily create responsive designs.
- We are using bootstrap templates for styling our webpage. Bootstrap templates provide many easy-to-use style for various functionalities such as forms, buttons, tables etc.

5.3 Software Engineering Methodologies

5.3.1 Agile Methodology

We will use Agile methodology for development of our application.So the features will be built initially as small incremental features, and then the final one was built on top of those. We are making use of the storyboard feature provided by github to form issues and then resolve them.

5.3.2 Code Review

Code Review practices are followed by creating different branches in github other than master and working on those before pushing on master. In this way, we will try to reduce bugs and error in our code. Code will be reviewed by the all team members before committing any file to the master.

5.3.3 Pair Programming

Pair programming is an agile software development technique in which two programmers work together at one workstation. One, the driver, writes code while the other, the observer or navigator,[9] reviews each line of code as it is typed in. The two programmers switch roles frequently.

While reviewing, the observer also considers the "strategic" direction of the work, coming up with ideas for improvements and likely future problems to address. This is intended to free the driver to focus all of their attention on the "tactical" aspects of completing the current task, using the observer as a safety net and guide.

We, being a group of 4, formed 2 groups who will pair program the core functionality of the system. This will help debug errors faster and more than 1 member will know how the module functions. Thus, if 1 of the programmers is not available, then there is always a backup for urgent changes or issues.

6. SURVEY

We took a survey to find out how people are tackling issues with the current version of the system and regarding the changes we propose to implement. The questions which we asked are as follows:

1. **Would you be willing to add reviews for courses to improve your reputation within the application (similar to stackoverflow)?**

   Since, the application uses reviews from previous takers of the course, building a reputation will enable to promote a review i.e. more helpful ones.

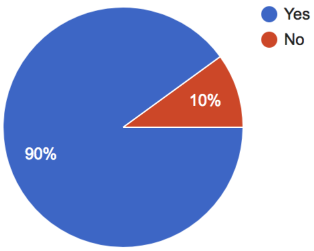

Figure 3: Response for Question 1

2. **Would you add the syllabus of the courses you have taken to the portal?**

   The purpose of this question so that we could have a way to make the syllabus available for future interested candidates.

   The results suggest that most of the students will add the syllabus of the course that they had taken.

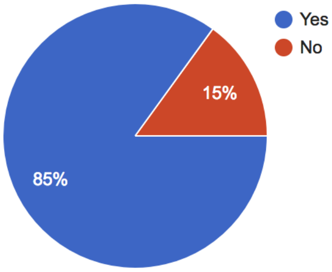

Figure 4: Response for Question 2

3. **For adding your interests (for subsequent course suggestion), what will you be comfortable doing?**

   We needed to find a way to fetch the student's interests which will be used for course suggestions. This question was aimed at finding out which out of the following methods would be an ideal way to get the student's interests.

   This result tell us that most students prefer answering a questionnaire over attaching a Resume or writing a summary. A considerable number also mentioned they wouldn't mind giving a summary stating their interests and projects.

6.1 Survey Conclusion

Based on the responses we got from our survey, we can conclude that the functionalities which we plan to add in original WolfPal are welcome by the users. The students will like to use the review platform like StackOverflow where reputation is earned by adding more helpful reviews. This would help students filter out the irrelevant comments from the useful ones. The idea that the students will be willing to add the syllabus of courses is very critical to our project and we got a positive response regarding the same.

Mapping the keywords of courses with interests is an idea which was appreciated by the respondents. We had asked the users 3 suitable ways for collecting the interests. As all 3 ways were acceptable by the survey respondents, we are planning to implement a combination of all 3: resume, questionnaire and a textual summary of interests.

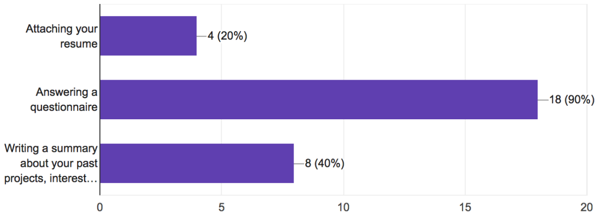

Figure 5: Response for Question 3

7. EVALUATION

7.1 Evaluation Method

After the prototype is ready, we plan to do Beta testing [1] and User acceptance testing. Apart from the features already present, the particular areas where we would like to test are

1. Adding content to the system and effects

   (a) Only a user with a healthy reputation can add syllabus:

   Users with reputation less than a threshold cannot add syllabus to the courses just to ensure reliability of data.

   (b) Everyone can add and view reviews for various courses

   Users can check courses based on their reviews to make a decision about which course to select.

   (c) Only a user with a healthy reputation can upvote or downvote a review

   For ensuring review quality, we plan to allow only users with a healthy reputation to upvote/downvote a review. This will ensure that actually good and helpful reviews do not get missed by users because some user decided to create multiple accounts and downvote the review.

   (d) The reputation points of the owner of a review/post get updated after an upvote or downvote

   The user whose post gets upvoted gets reputation points which should be reflected in his profile.

   (e) Users are getting reputation points after adding reviews

   The user who adds a review gets reputation points which should be reflected in his profile.

2. Course recommendation

(a) Correct topic identification

For each course summary/syllabus, we plan to do topic modeling. We plan to set the topic labels for around 20 courses by ourselves. Then we plan to test the topic modeling algorithm against those labels and retrieve metrics such as accuracy, precision, recall, f-score.

(b) Correct keywords are identified from the course summary, syllabus and reviews

For each course, the keywords that are extracted from the summary should be relevant to the course description.

(c) Correct matching of keywords is taking place to show recommendations

Keywords generated from each userâĂŹs interest should be mapped properly to the keywords corresponding to the courses to ensure that the recommendations are correct.

Apart from these areas, we would also like to evaluate the performance of all the modules working together especially after the new modules have been added.

8. CONCLUSIONS

WolfPal is a single go-to application for student who want to get information on courses offered by NC State University. WolfPal provides a variety of options for the student to decide which courses he/she should take. Student can get course information based on name, keyword search. Additionally, if the student does not know the courses offered by the university, he can mention his area of interest or an abstract of his achievements, and WolfPal will give a list of courses which will fit his area of interest and also satisfy the breadth requirements for the major. Apart from that, WolfPal provides a unique feature where the student can upload resume and the system will use natural language processing to extract keywords and map them to courses for the student. Thus, WolfPal is a resourceful stand-alone application for students who face difficulties selecting courses.

9. REFERENCES

[1] WolfPal Report *WolfPal CSC 520* Retrieved from: `https://github.com/ragarwa7/WolfPal/blob/master/Reports/team-k_wolfpal_mar_report.pdf` [Accessed: 21-Mar-2018].

[2] R. Bhatt, D. Desai, L. Shi, and C. Zhao. *Course feedback system. Software Engineering, 2017.* Retrieved from: `https://github.com/Rushi-Bhatt/SE17-Team-K-CourseFeedbackSystem/blob/master/Final_Report_MAY.pdf` [Accessed: 21-Mar-2018].

[3] Al-Badarenah, Amer, and Jamal Alsakran. *An Automated Recommender System for Course Selection.* International Journal of Advanced Computer Science and Applications 7.3 (2016): 1166-175.

[4] MongoDB Documentation, Retrieved from: `https://docs.mongodb.com/` [Accessed:26-Mar-2018]

[5] Python Documentation *Python* Retrieved from: `https://www.python.org/about/` [Accessed: 26-Mar-2018]

[6] NLTK Documentation *NLTK* Retrieved from: `https://www.nltk.org/install.html` [Accessed: 26-Mar-2018]

[7] JavaScript Documentation *JavaScript Documentation* Retrieved from: `https://en.wikipedia.org/wiki/JavaScript` [Accessed: 26-Mar-2018]

[8] Bootstrap Tutorial *Bootstrap W3Schools* Retrieved from: `https://www.w3schools.com/bootstrap/bootstrap_get_started.asp`[Accessed:26-Mar-2018]

[9] Pair Programming *Pair Programming - Wikipedia* Retrieved from: `https://en.wikipedia.org/wiki/Pair_programming` [Accessed: 21-Mar-2018]

[10] TechTarget *What is Beta Testing?* Retrieved from: `http://whatis.techtarget.com/definition/beta-test` [Accessed: 21-Mar-2018]