Course Recommender System Using Keyword Classification and Content-based Filtering

Heejin Kang[a,e.1], Daeyoung In[b.2], Jeemin Seo[c.3], Jinwoo Ryu[d.4]

[a]*Economics, Underwood International College, Yonsei University*
[b]*Statistics and Data Science, College of Commerce and Economics, Yonsei University*
[c]*Chemical and Biomolecular Engineering, College of Engineering, Yonsei University*
[d]*Chinese Language and Literature, College of Liberal Arts, Yonsei University*
[e]*Computer Science, College of Computing, Yonsei University*
[1]*heejin.kang@yonsei.ac.kr,* [2]*daniel_in@yonsei.ac.kr*
[3]*jeeminsally05@yonsei.ac.kr,* [4]*bedro10312@yonsei.ac.kr*

Abstract
This paper aims to implement a robust hybrid recommender system that utilizes keyword classification, sentimental analysis, and content-based filtering to make recommendations for new courses based on a student's course enrollment history. We implemented a web crawling algorithm for obtaining raw data of general electives and their corresponding reviews. We then apply keyword classification to find variables that can be used for recommendation based on the review data. The course categorical data and extracted keyword data are used as the variables for our recommender system which employs the content-based filtering method. Widely used TF-IDF vectorizer and cosine similarity functions were chosen for content-based filtering. The weight setting method for our algorithm was inspired by previous research on a hybrid recommender system which used fixed weights for non-scalable binary data. The top scoring courses are recommended to the user. Investigation of methods for evaluation are ongoing in order to improve the accuracy of our recommender system.

Keywords: Datum Academia, Data Science, Keyword Classification, Sentimental Analysis, Content-based Filtering, Recommender System, Web Crawling

I. Introduction

1.1 Problem Identification

'Everytime' is a mobile application that is widely used by university students in Korea for finding a suitable timetable for upcoming semesters. In order to fulfill their graduation requirements, students have to manually go through ratings and course reviews for each course they are interested in enrolling. Though 'Everytime' provides a good

amount of information on the course such as rating, reviews, and information on the number of tests and assignments, this whole process seems to be an unreliable way for determining whether the selected course is an adequate fit for the user. We identified this method of course selection to be not only tedious and complicated but also inaccurate. This problem can be highlighted in the fact that students must switch between different tabs in order to look at general electives from different categories. Whereas there are strict graduation requirements and a small range of courses for major requirements and major electives, there is a great number of general electives to choose from. Since selecting general electives requires the user to manually go through all the courses for every single category, it can be said that there is a great need and demand for alleviating this time-consuming process.

1.2 Problem Prioritization

There are a few areas that need further inspection to solve this course selection problem. First and most importantly, we need to identify the factors that can be utilized for accurately recommending satisfactory courses. Secondly, we need to determine the size of the data set utilized throughout the whole recommendation system. We decided to obtain the input data via web crawling. From our input data, we had to question and scale whether the dataset was effective in providing accurate recommendations.

1.3 Analysis

The identified problems have led us to think about the analysis of the ‘Everytime’ course selection process. As previously addressed, it was clear that there needed to be a better system to overcome the tiring process of individually selecting general elective courses out of a long list of categories. However, the most important problem was that there were no specific comparisons and criteria that we chose to select these general elective courses. For instance, a newly registered course with no rating/ review may be a better fit for a certain user than a course that has a rating of 2 stars. Another example is choosing a course based on recommendations from a friend and ending up being dissatisfied with the course due to different preferences.

1.4 Justification

There are a number of precedent researches related to this topic that aim to provide an aid for students to come up with time schedules for their upcoming semesters. There are projects that are constructed to aid students look for courses of their taste through keywords extracted from course reviews and those analyzing the similarities of courses in the system. However, few precedent projects focus on directly providing a set of courses that match the needs of individual users. Therefore, our team proposes a model which provides a direct set of courses that minimizes the possibility of leaving out courses that match the user’s preference due to human error.

1.5 Research Question

Can we accurately recommend general elective courses according to a student's enrollment history?

II. Literature Review

2.1 Case Study: *How to improve the accuracy of recommendation systems: Combining ratings and review texts sentiment scores*

The literature proposes the usage of 'sentiment analysis' as an attempt to improve the accuracy of recommendation systems. There exists a limitation in a sense that recommendations were mostly based on quantitative information such as users' ratings, which made the accuracy be lowered. To solve these problems, many studies have been actively attempted to improve the performance of the recommendation system by using other information besides the quantitative information. The literature suggests sentiment analysis on review text data as a means to incorporate both quantitative (rating score) and qualitative data in the recommendation process. The study shows that the collaborative filtering that reflects sentiment scores of user review is superior to the traditional method that only considers the existing rating. It further reaches a conclusion that the proposed model is superior to the traditional type of recommendation via evaluation processes. The research adopts the mean absolute error (MAE) and the root mean square error (RMSE) as an evaluation index and further attempted pair t-test validation to ensure the superiority. The literature provides justification for us to adopt sentiment analysis as a method to strengthen recommendation accuracy. In addition, it provides insight on the process of evaluating and verifying the constructed model.

2.2 Case Study: *TasteWeights: A Visual Interactive Hybrid Recommender System*

This paper employs a combination of different recommender systems from different sources of media to develop a hybrid recommender system for recommending music based on user preferences. The method introduced here combines models for different platforms, namely Wikipedia, Facebook, and Twitter to improve the accuracy of their recommender system. The section we want to focus on from this paper is the weighting method adopted for the Wikipedia model which is a type of content-based filtering system. When no scaled preference rating is available, the profile item was assigned a score of 0.5. This method can be employed in our recommender system where portions of the input data are non-scalable. In addition, the weight for each Wikipedia item is calculated as "the sum of the individual user-provided weights of the profile items it shares links with." This gives us insight on how to adjust the weights for the variables belonging to our items. We should also consider the prioritization approach applied in

this hybrid system. The authors considered a recommendation generated from more than one source to be of higher importance. Thus, they added a variable for the number of content sources a recommendation was generated by when calculating the recommendation score. We learn from this case study, one that has been cited almost 300 times, that the settings for a hybrid recommendation system are based on intuition as well as trial-and-error. This paper provides a new perspective on how one should consider justifying the weights and evaluation scores in a recommender system.

III. Research Objectives

3.1 General Objectives

Since the needs of every individual student vary, our aim is to provide an output that satisfies the specific preference of each user. In this way, users will be able to access relevant and customized information of general electives in an economical method. Our goal is to not only save a great deal of time and effort for users to look up these courses but to recommend courses that fit their standards but weren't found by the biased mind for different reasons.

3.2 Specific Objectives

To improve the accuracy of our model, first all the data should be collected on several values beforehand by extracting review data from the 'Everytime' website. Second, to establish standards thoroughly reflecting user preference, keywords are extracted from the reviews. This way we can add extra standards to the predetermined ones on the 'Everytime' system in a method minimizing bias. Next, sentimental analysis on the reviews will also fortify the accuracy, by verifying the extracted keywords. Then the final step is to implement recommendation using a content-based filtering method.

3.3 Limitations

There exist certain limitations regarding the relatively small size of the dataset compared to other studies and credibility issues coming from this factor. The size of review data is limited to a number under 15,000 reviews. In addition, we don't have access to user data regarding course preferences in advance of users' voluntary submittment. Therefore, there are limits in employing 'collaborative filtering' techniques that are frequently used as a recommendation method in related industries.

We aim to balance these accuracy issues by gaining profound, thorough insight over the given dataset in a diverse perspective and actively utilizing the information that was pre-analyzed by the 'Everytime' website such as rating scores etc. We further plan to improve the accuracy of the proposed model by employing qualitative information in the recommendation process.

3.4 Hypotheses

We believe the result of our project will successfully provide adequate choices for students who have a hard time digging up and further choosing the general electives they would like to enroll in. General elective courses can be recommended based on enrollment history and course reviews via keyword extraction and classification coupled with content-based filtering.

IV. Research Methodology

4.1 Variables

The independent variables of the research are the data from the reviews in 'Everytime' including user ratings, reviews, classifiers, etc. This data is used in the construction of the recommendation model. During the construction and testing process, confounding variables required to take into consideration are predicted to be manual labeling processes based on subjective bias, and careless reviews. As a result, dependent variables include user customized sets of recommended courses.

4.2 Types of Study *(Quantitative & Qualitative)*

This project incorporates both quantitative and qualitative study. Quantitative data in the form of course ratings is used to set a universal standard of recommendation and qualitative data such as keywords extracted from reviews or the estimated sentiment of these reviews provide further insight on driving the recommendation to customize individual use preference.

4.3 Data Collection Techniques

Building a course recommendation system required a data set utilizing selenium to crawl information regarding courses from the Everytime schedule page. Web crawling is the process of indexing data on web pages by using a program or automated script. Web crawling allows us to copy pages for processing by a search engine, which then indexes the downloaded pages. This method allows us to retrieve any information that are on one or more pages.

4.4 Sampling

The final processed data set was crawled from the main page of each elective course, which included the name of the elective course, name of the professor, rating, assignment weight, group work weight, grade weight, attendance weight, test frequency, and review. Those without a rating or review were replaced with a number of -1 rather than putting them to neglect as shown in Figure 1.

	강의명	교수명	평점	과제	조모임	성적	출결	시험 횟수	강의평
0	러시아문학	심지은	-1	-1	-1	-1	-1	-1	-1
1	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	철학을 몰라서 수업듣는 내내 너무 어렵
2	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	성적 매우 후하지만 철학에 관심이 없다
3	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	리딩이 너무 어려웠지만 나름대로 얻어
4	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	우선 본인 에쁠 받음1. 철학에 큰 관심 없
5	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	학점 어떻게 나올지 감도 안잡히는데 철
6	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	수업내내 로드가 생각보다 빡쎄다. 판단
7	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	배운건 많이 없지만 학점 잘 주시고 에서
8	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	솔직히 비추. 지만 3점인 이유는 학점을
9	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	책 읽는게 핵심인 수업인데 책들이 어려
10	미학	김동규	3.8	보통	보통	너그러움	전자출결	한 번	일단 이번 학기는 코로나 때문에 비대면

Figure 1. Course information obtained by web crawling from the ‘Everytime’ platform.

4.5 Plan for Data Collection

The course information that is displayed on the ‘Everytime’ platform comes from the school portal data. Each course and their corresponding reviews are available on a single page and can be accessed by a few clicks. Each course has multiple categories that characterize certain aspects of the course, such as number of assignments, whether or not there is a group project, how lenient the grading is, etc. We believe that these categories as well as the semantic data from the course reviews can be utilized for our recommender system. Thus, the data provided from the ‘Everytime’ platform is adequate for web crawling. We used the ‘webdriver’ module from the ‘selenium’ Python package to perform web crawling on the ‘Everytime’ website. Courses for general electives are classified into eleven different categories. By iterating through each class from all eleven class categories, we were able to extract all of the aforementioned class data as well as their corresponding reviews. The crawled data was stored in a dataframe and imported as an excel file for further data processing.

4.6 Plan for Data Processing and Analysis

After the user reviews have been verified by sentimental analysis, the last step is to implement the actual recommendation algorithm. The method we have chosen is content-based filtering, because it matches with the data that we have collected. There are two main functions involved in this algorithm which are TF-IDF vectorizer and cosine similarity.

$$\text{tfidf}(t, d, D) = \text{tf}(t,d) \times \text{idf}(t, D) \qquad (1)$$

The equation for TF-IDF vectorizer is shown in equation (1). The TF-IDF vectorizer is calculated by multiplying the term frequency (tf) by the inverse document frequency (idf) which are shown in equations (2) and (3), respectively.

$$\mathrm{tf}(t,d) = 0.5 + \frac{0.5 \times \mathrm{f}(t, d)}{\max\{\mathrm{f}(w, d) : w \in d\}} \quad (2)$$

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (3)$$

The idea is to represent how relevant a certain word is in a given document by the use of vectors. Then documents with similar, relevant words will have similar vectors, which is what we need for our recommendation algorithm. The tf function is used to evaluate how frequently the word appears in a given document. In equation (2), the frequency of the target word is divided by the frequency of the word with highest frequency. In equation (3), the idf formula is calculated by taking the log of the total number of documents over the number of documents that contain the target word. Thus, the relevance of a target word depends on how frequently it appears in the document as well as how unique the word is in the given document relative to the other documents.

$$\cos(\theta) = \frac{A \cdot B}{|A|\,|B|} \quad (4)$$

Once we have vectorized our keywords using the TF-IDF vectorizer, we can calculate the similarity of different courses using a cosine similarity function as shown in equation (4). This is just a variation of the dot product. The cosine similarity allows us to quantitatively measure the similarity of different courses. The smaller the angle between two vectors, the larger the value of the cosine. Therefore, having a large cosine similarity means that the two TF-IDF vectors are similar. In this way, we can find courses that are similar to each other based on their categorical and review data. Although our data will have a much higher dimension than the subspace represented in Figure 2, this visualizes how the cosine similarity can help us. We can see from the figure that the more similar two vectors are, the smaller their angle will be. After applying the cosine similarity, we will have a matrix of courses with their respective cosine similarity values. By summing the cosine similarity values of the courses taken by the user, we can obtain the top ranking general electives that are similar to the courses that the user has previously taken.

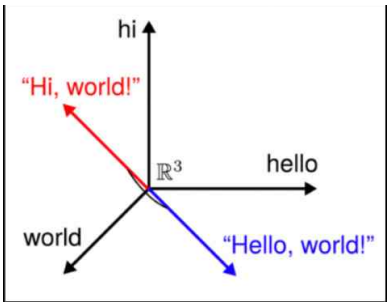

Figure 2. Visual representation of the cosine similarity function.

The data we have collected can be segmented into two types: categorical data, e.g. number of exams, assignments, leniency of grading, etc., and qualitative data, i.e. course reviews. The categorical data is encoded using numerical labels while taking into account the dissimilarity between different courses based on the distance between vectors. For example, a course with no exams will be considered to be the most dissimilar with a course with four exams (the maximum number). In this way, numerical encoding allows the distance between the vectors of different courses to be reflected. The second type of data, the course reviews, must undergo TF-IDF vectorization. The keywords are selected subjectively after filtering all words in the course reviews by frequency. A dictionary is used to count the frequency of each word to simplify the filtering process. The data is pre-processed such that the number of keywords as well as their respective counts are saved to each course. Thus, the frequency of keywords are taken into account for calculating dissimilarity as well. Once both types of data are transformed into vector form, the cosine similarity can be applied. The cosine similarity function as shown in equation (4) can be applied directly to the categorical data, but a special Python package is necessary to calculate the cosine similarity for the course reviews. Finally, a content-based recommendation function that calculates and returns the top ten highest rating courses for a given course is implemented. The result is shown in Figure 3.

	target_title	recom_title
0	논리회로설계	체력육성
1	논리회로설계	SW프로그래밍
2	논리회로설계	통계방법론
3	논리회로설계	한문(2)
4	논리회로설계	프랑스문화와예술
5	논리회로설계	철학과윤리
6	논리회로설계	시각예술의이해
7	논리회로설계	페미니즘의이해
8	논리회로설계	논리회로설계
9	논리회로설계	회계원리(1)

Figure 3. The top ten most similar courses can be calculated based on the proposed model.

4.7 Ethical Considerations

Web crawling is a simple and convenient method for gathering data for research purposes. However, one should be aware of the ethical considerations of this process. For example, if the data collected by web crawling was personal or restricted information, it is obvious in this case that requesting for permission would be the ethical choice of action. We were careful not to violate anyone's personal information and focused on obtaining information that is openly accessible. Another point to consider is that aggressive crawling can lead to functionality errors on the website. We were also careful not to put an excessive amount of strain on the website server when initiating our web crawling algorithm. There are limitations to the extent of research that can be conducted considering that without each user's data profile, it is impossible to employ recommender algorithms such as collaborative filtering. Nonetheless, it is important to respect the ethical boundaries of a website when utilizing such data collecting methods.

4.8 Pre-test or Pilot study

Content-based filtering is one of the most common methods used in recommender systems. Due to the nature of our collected data, we are limited from using other recommendation algorithms such as collaborative filtering. We employ the item-based algorithm on our data which has both discrete and continuous data. We looked into a similar research example that utilizes semantic data from Wikipedia articles to make content-based recommendations. In this algorithm, recommendations were made and evaluated from various platforms. Due to the variations in sources and recommendation methods, the researchers were flexible on determining the overall recommendation score of their model. When no preference scale was available, a fixed value of 0.5 was assigned for evaluation. The aforementioned study gives us insight on how to set the weights for our algorithm. We apply this method on non-scalable data of our courses. Furthermore, we set the weights of the review data based on keyword vectorization and cosine similarity. This is a common method used in recommender systems that considers the relevance of semantic data.

V. Research Plan

5.1 Outline of Research Paper

To summarize the outline of our project, we have four main steps. The first step is web crawling to collect data on the general electives and their corresponding reviews. Using the data we collected in the first step, keyword extraction is applied on the review data. Once the keyword data is ready for sentimental analysis, we use sentimental analysis to filter through and check for any misplaced keywords. After this step is

complete, we combine the course categorical data and keyword data to find any relationships between courses and make recommendations based on these relationships.

5.2 Schedule of Research

1. Discuss Research Topic **(week 3)**
2. Data Collection: web crawling on course and review data **(week 4-5)**
3. Data Cleaning: preprocess data for data analysis **(week 6, 9)**
4. Data Analysis: keyword extraction, sentimental analysis, content-based filtering **(week 10-12)**
5. Interpretation and Evaluation **(week 13)**
6. Data Visualization and Finalization **(week 14)**

On a concluding note, there are a few areas that we hope to further investigate. In case the recommendations from our algorithm are unsatisfactory, we may need to make adjustments to the weights of some of our variables, especially the course categorical data. We applied TF-IDF vectorization to the keywords in order to quantify their relevance in each document. However, we have yet to find a method for assigning weights to the course categorical data that reflect their importance. This remains one of the main challenges for improving the accuracy of our recommendation system. One solution for this is implementing machine learning techniques to our algorithm. Many recommendation systems combine different algorithms such as collaborative filtering, neural networks, and content-based filtering. We plan to look further into previous studies that can provide insight on how to add machine learning techniques to our algorithm. Lastly, taking into account the schedules of the recommended courses is a feature we believe is beneficial from the user's perspective.

Keyword extraction uses machine-learning artificial intelligence (AI) with natural language processing (NLP) to break down human language so that it can be understood and analyzed by machines. It is used to find keywords from various types of text: regular documents and business reports, social media comments, etc. We are going to be utilizing this technique to extract keywords from written course reviews from 'Everytime'. We will attempt to obtain important keywords such as (꿀강, 명강, 좋다… etc.), and those set of keywords will establish their own column. As opposed to the subjective selection of keywords implemented in this study, this method can include words that may be representative of the course but are not necessarily frequent. We expect to improve the accuracy of our current model by implementing keyword extraction via NLP. Another method that we will use in order to analyze the collected data is sentiment analysis. As previously mentioned we are aiming to extract keywords from the course reviews and basically use them as standards for data classification. However, we realized that there is room for error in this classification method.

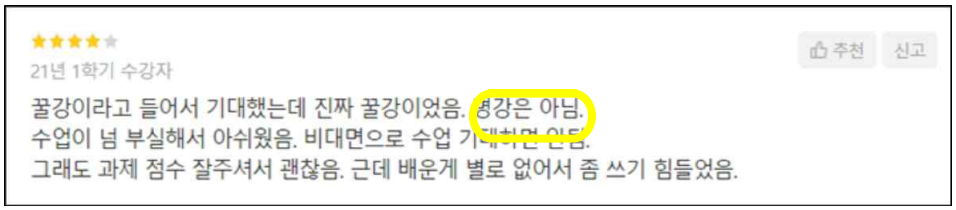

Figure 4. Course review example.

Let us look into a simple example shown in Figure 4. Suppose that a user was looking for a memorable course, easily known as a '명강'. According to our methodology, in the right condition, it is likely for this course to be classified as a '명강', just because of the fact that the review includes the keyword '명강'. In this example, keywords are used differently than originally anticipated. This type of error was estimated to occur quite frequently and therefore, we decided to use the result of the sentiment analysis of these reviews (positive or negative) in order to minimize error. Thus, as a means to improve accuracy, we are planning to implement sentiment analysis on keywords including reviews for the verification of keywords.

Figure 5. Sentiment analysis.

To build a sentiment analyzer for the process we plan to use machine learning techniques. First, pre-processing of text is necessary. For the preprocessing, we plan to use tools such as KONLPy, NLTK. We tokenize the given corpus based on morpheme units and then we implement normalization and cleaning process by stemming, i.e. extracting relevant morphemes that have meaning, and stopword removal. The next step is feature hashing. We plan to increase the performance of the model by vectorizing features using the TF-IDF vectorization. Then we split the data set into the training data and test data. Then we build a logistic regression algorithm using a model that results in either 'positive' or negative. The model is further evaluated with the test data that was previously split. Finally, we are able to analyze the sentiment of the reviews to verify keywords.

VI. Reference (Bibliography)

1. Hyun Jiyeon, Yoo Sangyi, Lee Sangyong. <How to improve the accuracy of recommendation systems: Combining ratings and review texts sentiment scores>. Journal of intelligence and information systems v.25 no.1. 2019. pp.219 - 239
2. Bostandjiev, Svetlin & O'Donovan, John & Höllerer, Tobias. (2012). TasteWeights: a visual interactive hybrid recommender system. 10.1145/2365952.2365964.
3. Manning, C. D.; Raghavan, P.; Schutze, H., <Introduction to Information Retrieval>. Cambridge University Press. p.100-123. “Scoring, term weighting, and the vector space model”
4. Selva Prabhakaran, “Cosine Similarity - Understanding the math and how it works (with python codes)”, Machine Learning Plus, October 22, 2018. https://www.machinelearningplus.com/nlp/cosine-similarity/
5. Atsumi Kyoko, “Recommender Systems: Content-based Recommendations & Collaborative Filtering [FULL]”, Medium, July 12, 2021. https://medium.com/mlearning-ai/recommender-systems-content-based-recommendations-collaborative-filtering-full-6483b6caa5eb#01c2