Available *online* at : http://jitce.fti.unand.ac.id/
JITCE (Journal of Information Technology and Computer Engineering)
| ISSN (Online) 2599-1663 |

Research Article

Advisor-Oriented Course Recommendation System Using Student Grades

Muftah Afrizal Pangestu [1], Oscar Karnalim [1]

[1] *Universitas Kristen Maranatha, Jl. Surya Sumantri No.65, Bandung, 40164, Indonesia*

ARTICLE INFORMATION

Received: February 00, 00
Revised: March 00, 00
Available online: April 00, 00

KEYWORDS

recommender system, student advisor, content-based filtering, collaborative filtering, computer science education

CORRESPONDENCE

Phone: +62 (0751) 12345678
E-mail: first_author@affiliation.xx.xx

ABSTRACT

In some universities, student advisors are often hired to enhance students' retention rate. Having some students in mind, these advisors may find some difficulties in guiding the students in terms of selecting relevant courses. This paper proposes an advisor-oriented course recommendation system. Using this system, the advisors may suggest relevant courses to their students easier and more accurate. This system relies on student grades and comprehensive course data. Further, it utilises content-based and collaborative filtering for predicting relevant courses. According to our evaluation, the system is considerably effective; the accuracy of content-based filtering is about 66% while the accuracy of collaborative filtering is about 58%. Further, some parameters may be potential for enhancing accuracy while the others may be not.

INTRODUCTION

Student retention is a crucial aspect in universities [1]; it does not only affect student's future but also the university's as well. Several strategies are therefore considered to keep the retention high [2]. These strategies are usually emerged from the analysis of student data [3] and student characteristics [4]. Apart from newly-introduced strategies to maintain high retention (e.g., Student Success Course [5], a persuasive social media [6], or the integration of educational technologies [7]), the existence of student advisors is a common practice in some universities, especially those that offer formal academic degree.

Student advisor is a person who is responsible to guide students in terms of their academic path. In some universities that offer formal academic degree, one advisor is allocated to some students for enhancing their' retention rate. Having some students in mind, the advisor may find some difficulties in guiding the students; students have their own unique skill while some of them are still not aware with it. In addition, the description of available courses may be not comprehensive enough for guidance.

These advisors are responsible to guide students so that they can take courses relevant to their skills. With such thing in mind, it is expected that they can complete the courses, which obviously enhance the student retention rate. However, providing such guidance takes a considerable amount of time based on three reasons. First, the skills of each student are unique and different to each other. Second, some students are not aware with their own potential, which means what they say may not reflect their skills. Third, the description of available courses may be not comprehensive enough to cover the courses' content, requiring more observation prior deciding whether some courses are suitable for a particular student.

To mitigate advisor' effort, this paper proposes a course recommendation system that only considers students' empirical skills (represented by their grades) and comprehensive course data (i.e., course syllabi or slides). The system does not only suggest relevant courses but also provides their predicted result. Further, it displays two recommended course lists instead of one to facilitate more comprehensive analysis; one of them is based on content-based filtering while another is based on collaborative filtering. Using this system, an advisor may suggest some courses to a particular student easier and more accurate.

Recommender system aims to help humans in terms of suggesting the most suitable items based on their interest [8]. The items itself vary from movie [9], books [10], e-commerce products [11],

https://doi.org/10.25077/7.02.63-72.2023

academic publication venues [12], to learning materials [13]. In general, the system can be categorised further either based on their technique [8] or application [14].

From technique perspective [8], recommender system can be classified into four categories: demographic, content-based, collaborative, and hybrid filtering. Demographic filtering [15] recommends items based on users' shared personal attributes. Content-based filtering [16] utilises user's previous choices. Collaborative filtering [17] relies on users' rating toward the items. Hybrid filtering [15] combines two or more aforementioned filtering mechanisms. Two examples of hybrid filtering are a work in [18] (that combines collaborative with content-based filtering) and a work in [19] (that combines collaborative with demographic filtering).

From application perspective [14], recommender system can be classified to eight categories. These categories are: 1) e-business recommender system that focuses on product for business owners [20]; 2) e-commerce recommender system that is like e-business system except that it focuses on commercial transaction [21]; 3) e-government recommender system that focuses on government-related information [22]; 4) e-learning recommender system that focuses on learning materials [23] or potential supervisors [24]; 5) e-library recommender system that focuses on knowledge sources such as e-books [25]; 6) e-tourism recommender system that focuses on tourist attractions [26]; 7) e-resource services recommender system that focuses on electronic files (e.g., video and movie) [27]; and 8) e-group activities recommender system that focuses on group preferences [28].

Among e-learning recommender systems, course recommender system is considerably common. Given a high number of available courses, the system will recommend some courses that match user's interest or skills [29]. This system is commonly used on Massive Open Online Courses, where a lot of courses are available to choose.

In terms of their applied technique, most course recommender systems use collaborative filtering. A course is recommended to a particular user if the user shares similar characteristics with enrolled users on that course. Such kind of filtering usually relies on three kinds of data: user rating [30], user interest via survey [31], [32], and user log history [31].

For higher accuracy, collaborative filtering is often combined with content-based filtering. A work in [33] utilises content-based filtering as a correction to the result of collaborative filtering. This work relies on four parts of data: user information, user learning behaviour, course information, and course rating. A work in [34] implements both filtering techniques separately. Collaborative filtering is used to recommend courses based on users' preferences while content-based filtering is used when a user wants to search courses with the same context (measured from the courses' descriptions). A work in [35] searches relevant courses based on users' implicit query (formed based on the users' previously-visited learning objects) through content-based filtering. Each relevant course is then featured with related courses (that are computed based on collaborative filtering toward users' web session logs). A work in [36] relies on Vector Space Model toward course materials to perform content-based filtering and good learners' average rating to perform collaborative filtering. A work in [37] utilises the result of content-based filtering as an input of collaborative filtering.

In addition to collaborative filtering and the combination of content-based and collaborative filtering, other filtering techniques are still used for recommending courses (even though they are uncommon). For instance, a work in [38] combines collaborative with demographic filtering.

It is true that some works have also proposed course recommendation systems. However, none of them are focused on helping student advisor; they are more focused on providing the recommendation directly to students. We would argue that such a direct manner is risky in the context of formal education; recommendation system's result is not 100% accurate while the result affects more on students' future. When students know that they will succeed on a particular course, some of them may not study seriously since they know that they will succeed (even though the fact will be in reverse if they do not study). When students know that they will not perform well on a course that they are interested in, they will be less motivated. Advisors' guidance is needed in both cases so that students can respond to the results more carefully.

Another differentiating factor between our proposed system with other recommendation systems is that our system relies on finer level of granularity of data. For example, we utilise course slides in addition to course syllabi. Further, we do not consider student interest in the process as that may lead to lower retention rate; some students may choose interesting courses with insufficient skills.

METHOD

This paper proposes an advisor-oriented course recommendation system. Unique to this system, it exclusively relies on student empirical skills (i.e., grades) and comprehensive course data, ignoring student interests. It also predicts student's result on suggested courses to enable more comprehensive analysis. From technique perspective, content-based and collaborative filtering (i.e., two common techniques on recommendation system) are applied to recommend relevant courses. In our context, relevant courses refer to courses where the given student should be able to pass.

Figure 1 shows how our proposed system works. It has three components namely course data preprocessing, content-based filtering, and collaborative filtering. Course data preprocessing converts all course syllabi and slides to terms (which will be used for content-based filtering). Content-based filtering suggests relevant courses to a student (whose ID is inputted) based on the student's grades of taken courses and course terms. Collaborative filtering suggests relevant courses based on other students whose grades are similar to the student's.

 https://doi.org/10.25077/7.02.63-72.2023

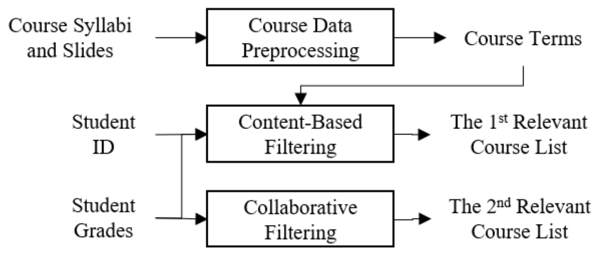

Figure 1. How our proposed system works.

Course Data Preprocessing

This phase converts course syllabi (as text files) and course slides (as PDF files) to course terms that will be used on content-based filtering. Figure 2 depicts how such conversion works. At first, it extracts all sentences from course slides with PDFMiner [39] and stores them as text files (one slide is stored in one text file). Secondly, along with course syllabi, the slides are fed to word splitter. This splitter returns sequences of lowercased words (one course corresponds to one sequence) by utilising whitespaces and punctuations as delimiters. Thirdly, stop words from resulted word sequences are removed based on stop word lists provided by Scikit-learn [40] (for English) and PySastrawi [41] (for Indonesian). We utilise those two lists since our course syllabi and slides are written in English and Indonesian. Fourthly, n-gram sequence for each word sequence is generated. n-gram is a technique which considers *n* adjacent words as one term. For instance, if {"this", "is", "a", "sample"} is a word sequence, its 2-gram sequence will be {{"this", "is"}, {"is", "a"}, {"a", "sample"}}. Finally, resulted n-gram sequences are then converted to course terms (as indexes). Index is a hash map containing key-value tuples wherein key refers to terms and value refers to their occurrence frequencies. If given n-gram sequence is {"this", "is", "another", "sample", "this", "is", "a", "sample"}, for example, its index will have 5 tuples: {"this":2}, {"is":2}, {"another":1}, {"sample":2}, and {"a":1}.

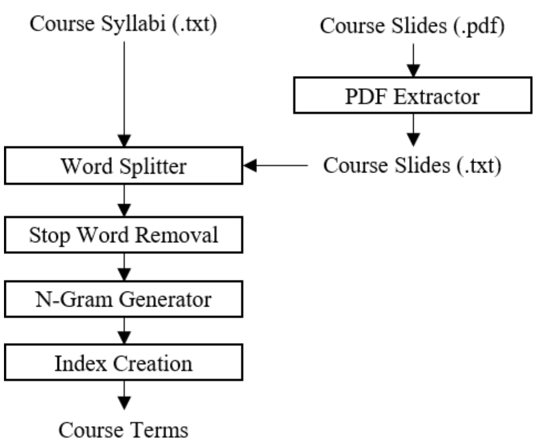

Figure 2. Course data preprocessing

Stemming is out of our consideration since course syllabi and slides are written in bilingual (English and Indonesian). English stemming may ruin the meaning of some Indonesian terms and vice-versa. It is true that we could detect the terms' language prior stemming. However, in addition to more processing time, it could be also ineffective considering some English terms are often used on Indonesian sentences.

Content-Based Filtering

This phase considers student grades and course terms (as indexes) to recommend relevant courses for a student whose ID is inputted (See Figure 3). Considering the grade for each course varies among students, it will be performed separately per student.

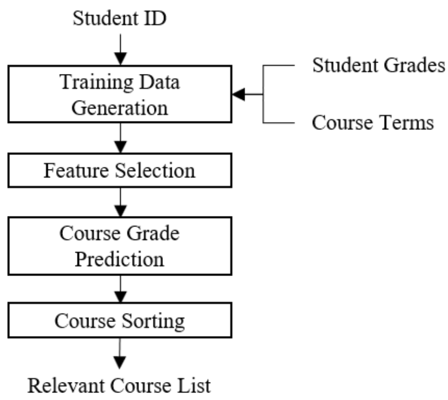

Figure 3. Content-based filtering that is conducted in fourfold.

First of all, it generates training data related to given student ID. It takes all student grades which correspond to the ID and correlates them with distinct course terms. The grade will become target class while the terms will become features with their frequencies as their feature values. To provide more uniform values on the target class, all grades are converted to three categories: "pass with excellence" (for grades higher or equal to B), "pass" (for values lower than B but higher or equal to C), and "fail" (for grades lower than C). If all of the student's grades fall on the same category, the remaining steps will be ignored (since it will be biased if a prediction is made only from single-valued target class) and content-based filtering will return no relevant courses. To illustrate this, let assume we have two courses (Introductory Programming and Information System) with their own course terms and student grade (see Fig. 4). Training data will then have two entries where each entry refers to one course. For each entry, target class refers to student grade while its features refer to distinct terms' frequencies. It is important to note that, if a particular distinct term is not found on a course's terms, its feature value will be assigned with zero (see features named "data" on row 1 and "function" on row 2).

Secondly, the dimension of training data's features will be reduced through feature selection. In our context, we will use TF-IDF weighting (a technique to favour rarely-occurred features) and X^2 feature selection (a technique to select top-N most representative terms toward given target classes [42]). Both of them are implemented with the help of Scikit-learn [40].

Thirdly, a model from the training data will be formed with Naive Bayes or K-Nearest-Neighbours algorithm [43]. The model will be used to predict the grades of remaining courses. Prior doing that, remaining courses will be converted to testing entries with empty target class.

Course:	Introductory Programming
Terms	{{program:2}, {function:3}, {feature:6}}
Student Grade	Pass with excellence

Course:	Information System
Terms	{{program:3}, {data:8}, {feature:4}}
Student Grade	Fail

program	function	feature	data	Result
2	3	6	0	Pass with excellence
3	0	4	8	Fail

Figure 4. An example of training data generation for content-based filtering.

Fourthly, all courses will be sorted based on their predicted grade from "pass with excellence" to "fail". Later, the courses which prediction is "pass with excellence" or "pass" will be displayed as relevant courses.

Collaborative Filtering

This phase recommends relevant courses based on student ID and grades. Figure 5 depicts how this phase works. It begins by converting student grades to training data. Each student (except the one whose ID is inputted) corresponds to one training data entry; wherein its features are all available courses and their features' values are the student's grades. Similar with content-based filtering, all grades on that class are converted to three categories—which are "pass with excellence" (for values higher or equal to B), "pass" (for values lower than B but higher or equal to C), and "fail" (for values lower than C)—to provide more uniform values. It is important to note that target class is still undefined at this stage. It will be defined later at course grade prediction.

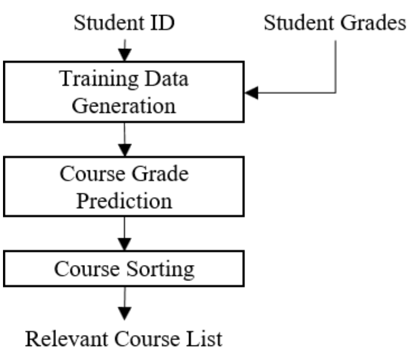

Figure 5. Collaborative filtering that is conducted in threefold.

Figure 6 shows an example of training data generation. Each student (as one column) will have N features where N refers to the number of available courses (in this example, we assume N is 5). These features will be assigned with their categorised grade for that student. For instance, student 1's feature corresponding to Introductory Programming is assigned with "pass with excellence" since they get an A for that course. It is important to note that features for untaken courses (e.g., Calculus for student 1 and Information System for student 2) are assigned with "undefined".

Student 1

Course	Grade
Introductory Programming	A
Information System	B+
Artificial Intelligence	C

Student 2

Course	Grade
Introductory Programming	D
Calculus	A
Discrete Mathematics	A

Training Data

Course	Student 1	Student 2
Introductory Programming	pass with excellence	fail
Information System	pass with excellence	undefined
Artificial Intelligence	pass	undefined
Calculus	undefined	pass with excellence
Discrete Mathematics	undefined	pass with excellence

Figure 6. An example of training data generation for collaborative filtering.

Course grade prediction will be performed for each course that has not been taken by the student with given ID. On each course, its corresponding feature will be promoted to target class. All training data entries which target class is "undefined" are removed from consideration; students corresponding to those entries have not taken the course.

The grade of a particular course will be predicted in twofold. All grades of the student with given ID (except the one that is being predicted) are mapped in similar format as training data entries'. Later, the course's grade will be predicted using either Naive Bayes or K-Nearest-Neighbours algorithm [43].

Upon predicting all courses' grades, these courses will be sorted based on their predicted grade from "pass with excellence" to "fail". Only courses which prediction is not "fail" will be displayed as relevant courses.

RESULT AND DISCUSSION

Our proposed methodology has two recommendation modules called content-based and collaborative filtering. Content-based filtering will be evaluated from nine perspectives: the impact of stop word removal on terms' descriptiveness, the effectiveness of stop word removal, the effectiveness of n-gram generator, the effectiveness of X^2 feature selection, the effectiveness of course data, the effectiveness of K-Nearest-Neighbours' number of neighbours, the effectiveness of prediction algorithm, the impact of academic semester, and the effectiveness variation among students' data. Collaborative filtering will be evaluated from two perspectives: the effectiveness of K-Nearest-Neighbours' number of neighbours and the effectiveness of prediction algorithm. It is

 https://doi.org/10.25077/7.02.63-72.2023

important to note that we put more evaluation on content-based filtering since it is more complex in terms of method. Further, it relies on course data in addition to student grades.

All evaluation will be performed on either balance or imbalance dataset; both of them are taken from our undergraduate students' data. Balance dataset represents a condition where the proportion of courses' topics and the proportion of students' academic merit are balance. It contains the data of 10 students who take the same 10 courses (3 programming courses, 2 mathematics courses, 3 hardware courses, and 2 general courses); these students are good at a particular topic while performing bad on others. Imbalance dataset represents a real condition on our faculty where the proportion of courses' topics and the proportion of students' academic merit may be not balance. It contains the data of 154 students as active students for eight semesters in our faculty. In other words, it covers most courses from our curriculum (some courses—e.g., Internship and Thesis—are removed since they are irrelevant to other courses). In total, there are 5945 non-distinct course entries and 58 distinct course entries. We consider all courses to be comparable one another without explicitly grouping them into categories.

In terms of evaluation metrics, human judgment, k-fold cross validation, real accuracy and McNemar's test will be used. Human judgment will be used for evaluating the impact of stop word removal on terms' descriptiveness. k-fold cross validation will be used to evaluate the effectiveness variation among students. Real accuracy will be used on remaining evaluation methods. McNemar's test will be used when real accuracy shows slight change among evaluated approaches.

Human judgment qualitatively measures terms' descriptiveness. For each evaluated approach, the descriptiveness of its extracted terms will be judged by the first author of this paper. If a term is descriptive toward given context, it will be assigned with 1. Otherwise, it will be assigned with 0. Approach A is considered to generate more descriptive terms than approach B if its total term score is higher than approach B's.

K-fold cross validation measures the effectiveness of an evaluated approach for each student in threefold [44]. At first, it partitions the student's data to K parts. Secondly, for each part, its local accuracy will be measured by treating that part as testing data and other remaining parts as training data. Thirdly, the effectiveness will be measured by averaging all local accuracy. In our context, K is assigned with 10 since that number is commonly used [44].

Real accuracy also measures the effectiveness of an evaluated approach. However, it portrays our real condition where the grade of a course can only be predicted through previously-taken courses (on k-fold, it assumes all courses have been taken). For example, as seen in Figure 7 which displays a student academic record, the grade prediction of the 2nd semester courses only rely on the 1st semester courses while the grade prediction of the 3rd semester courses rely on courses taken on previous two semesters. The grades of the 1st semester courses cannot be predicted since they have no previous semesters. The accuracy is simply measured by calculating how large is the proportion of correct prediction among all testing-training pairs for the whole students.

The 1st semester courses	The 2nd semester courses	The 3rd semester courses
Introductory Programming	Algorithms	Artificial Intelligence
Discrete Mathematics	Business Informatics	Data Structure
	Network and Security	

Testing Data	Training Data
Algorithms	Introductory Programming, Discrete Mathematics
Business Informatics	Introductory Programming, Discrete Mathematics
Network and Security	Introductory Programming, Discrete Mathematics
Artificial Intelligence	Introductory Programming, Discrete Mathematics, Algorithms, Business Informatics, Network and Security
Data Structure	Introductory Programming, Discrete Mathematics, Algorithms, Business Informatics, Network and Security

Figure 7. An example of mapping one student's academic history to training and testing data for real accuracy.

McNemar's test [45] measures whether two evaluated approaches generate different classification result. It relies on true positive, false positive, true negative, and false negative. Two classification results are considered as statistically different to each other if the test generates p-value lower or equal to 0.05.

Evaluating The Impact of Stop Word Removal on Content-Based Filtering: The Terms' Descriptiveness

Three approaches are considered in this evaluation. All of them are derived from content-based filtering with Naive Bayes as its prediction algorithm, top-10 1-gram features taken from X^2 feature selection as its features, and course syllabi & slides as its course data. They only differ in terms of stop word lists. The first one (referred as NO-STOP-WORDS) utilises no stop words. The second one (referred as DEFAULT) relies on English and Indonesian stop words provided by Scikit-learn [40] and PySastrawi [41]. The third one (referred as CUSTOM) combines DEFAULT's stop words with lecturer names and course IDs (course names are not considered as stop words since they still represent their course's context).

The impact of these approaches is measured on ten random courses from our imbalance dataset. For each course, ten terms with the highest X^2 value will be displayed and judged manually by the first author of this paper. In other words, the first author will judge the descriptiveness of 100 terms.

According to our evaluation, CUSTOM generates the largest proportion of descriptive terms (76 of 100), followed by DEFAULT (48 of 100) and NO-STOP-WORDS (46 of 100). Hence, two findings can be concluded. First, the existence of language-based stop words is not effective to increase the number of descriptive terms when TF-IDF weighting and X^2 feature selection has been applied. Second, lecturer names and course IDs (which are signature stop words for CUSTOM) are effective to generate more descriptive terms.

Evaluating The Impact of Stop Word Removal on Content-Based Filtering: The Effectiveness

This evaluation will rely on three approaches used on previous evaluation: NO-STOP-WORDS, DEFAULT, and CUSTOM. Their effectiveness will be measured based on real accuracy and McNemar's test toward all testing-training pairs from imbalance dataset.

Despite CUSTOM generates more descriptive terms (see previous evaluation), its accuracy (66.3%) is considerably similar to DEFAULT's (66.4%) and NO-STOP-WORDS's (66.3%). In other words, it can be stated that terms' descriptiveness may not affect accuracy. Another interesting finding is that the use of stop words may not lead to significant difference in terms of classification result. The p-value for NO-STOP-WORDS (an approach with no stop words) and CUSTOM (an approach with many stop words) is 0.93; it is far higher than 0.05 (the maximum threshold of significance).

Evaluating The Effectiveness of N-Gram Generator on Content-Based Filtering

The impact of n-gram generator can be measured by changing its n-gram constant (NG). Four unique NGs are used in this evaluation, starting from 1 to 4. Each NG corresponds to one evaluated approach that is derived from content-based filtering with Naive Bayes as its prediction algorithm, top-10 NG-gram features taken from X^2 feature selection as its features, and course syllabi & slides as its course data. These approaches will be compared to each other toward testing-training pairs from imbalance dataset with real accuracy and McNemar's test as their evaluation metrics.

According to our evaluation, no significant accuracy improvement occurs when NG is changed. All approaches generate considerably similar result: 66.5% for N=1, 66.4% for N=2, 66.3% for N=3, and 66.6% for N=4. When evaluated using McNemar's test, increasing NG does not significantly change classification result. Two approaches with extremely contradicting NGs (N=1 and N=4) are compared and their p-value (0.621) is still higher than the maximum threshold for significance (0.05).

Evaluating The Effectiveness of X^2 Feature Selection on Content-Based Filtering

The impact of X^2 Feature Selection can be measured by changing its number of features (NF). Four variants of such number are used in this evaluation: 5, 10, 20, and 50. The variants are applied to one approach each; wherein each approach is derived from content-based filtering with Naïve Bayes as its prediction algorithm, top-NF 1-gram features taken from X^2 feature selection as its features, and course syllabi & slides as its course data. These approaches will be evaluated toward our imbalance dataset by considering all testing-training pairs.

Figure 8 shows that higher NF leads to higher accuracy even though its improvement is not significant. The lowest accuracy occurs on NF=5 (which is 66.4%) while the highest occurs on NF=50 (which is 67.1%). Another important finding is that NF may not affect classification result. When two of these approaches with extremely contradicting NFs (NF=5 and NF=50) are compared with McNemar's test, their p-value (10.1%) shows no significant difference between those two's results.

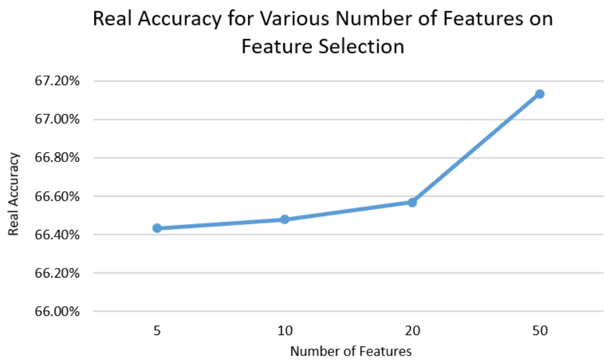

Figure 8. Real accuracy for various number of features (NF) on feature selection.

Evaluating The Effectiveness of Course Data on Content-Based Filtering

In our proposed content-based filtering, two kinds of course data are used: course syllabi and course slides. From those kinds, three approaches are generated for this evaluation. The first relies only on course syllabi and called SYL. The second relies only on course slides and called SLD. The third relies on both course syllabi and slides; it is called as SYL+SLD. All of them will be derived from a content-based filtering with Naive Bayes as its prediction algorithm and top-10 1-gram features taken from X^2 feature selection as its features.

Real accuracy and McNemar's test will be applied as our effectiveness metrics while all testing-training pairs for imbalance dataset is used as our evaluation data. According to our evaluation, SYL yields the highest accuracy (66.52%), followed by SYL+SLD (66.48%) and SLD (66.25%). In other words, course syllabi alone are sufficient for recommending relevant courses while the existence of course slides may slightly reduce the accuracy. When SYL and SYL+SLD are compared to each other using McNemar's test, they generate similar classification result; their resulted p-value is 0.08, which is still higher than the maximum threshold for significance.

Evaluating The Effectiveness of K-Nearest-Neighbours' Number of Neighbours on Content-Based Filtering

K-Nearest-Neighbours has a parameter called the number of neighbours (NN) which stands for how many closest neighbours that will be considered for prediction. This evaluation will measure how significant NN's impact on effectiveness (measured with real accuracy). For evaluation dataset, all testing-training pairs from imbalance dataset are used (10-fold cross validation on 5945 non-distinct course entries).

Four variants of NN are used: 1, 2, 3, and 4. Each of them refers to one approach that is derived from content-based filtering with NN-Nearest-Neighbours as its prediction algorithm, top-10 1-gram features taken from X^2 feature selection as its features, and course syllabi & slides as its course data. Figure 9 shows that a considerate improvement occurs when NN is increased. This

https://doi.org/10.25077/7.02.63-72.2023

finding is natural since more neighbours will lead to more considerate prediction.

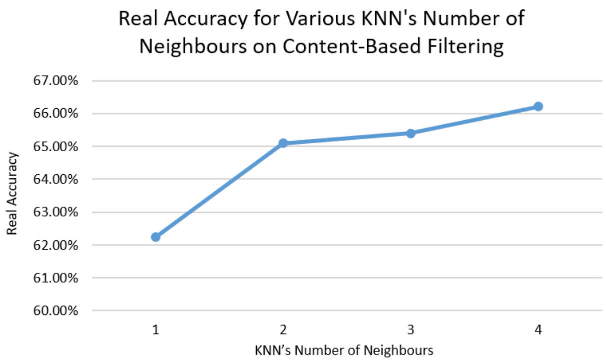

Figure 9. Real accuracy for various KNN's number of neighbours (NN) on content-based filtering.

Evaluating The Effectiveness of Prediction Algorithm on Content-Based Filtering

Two approaches derived from content-based filtering (with top-10 1-gram features taken from X^2 feature selection as its features and course syllabi & slides as its course data) are used in this evaluation. The former (which will be called NB) utilises Naive Bayes as its prediction algorithm while the latter (which will be called KNN) utilises K-Nearest-Neighbours with K=4 (the most effective KNN setting according to previous evaluation). They will be compared in terms of real accuracy and McNemar's test toward all testing-training pairs from imbalance dataset (10-fold cross validation on 5945 non-distinct course entries).

Two findings can be deducted from our evaluation. First, in terms of accuracy, NB (66.5%) is as effective as KNN (66.2%), even though it outperforms KNN for about 0.3%. Second, both NB and KNN generate similar classification result; their p-value for McNemar's test is 0.357, which is still higher than the maximum threshold for significance (0.05).

Evaluating The Impact of Academic Semester on Content-Based Filtering

Seven approaches derived from content-based filtering (with Naive Bayes as its prediction algorithm, top-10 1-gram features taken from X^2 feature selection as its features, and course syllabi & slides as its course data) are used in this evaluation. Each approach relies only on testing-training pairs from imbalance dataset for a particular semester, starting from the 2nd semester to the 8th semester (the 1st semester is not included since it has no training data for prediction). All of them will be compared based on real accuracy.

Academic semester is defined based on student enrolment. For example, if a student enrolled on the first semester of 2014, their 3rd semester will be different with the 3rd semester of a student enrolled on the first semester of 2015. The former will be the first semester of 2015 while the latter will be the first semester of 2016.

Figure 10 shows that the accuracy of our content-based filtering varies among academic semesters. A slight reduction occurs when the academic semester is later than the 5th semester. It could be caused by academic performance inconsistency from previous semesters. Given some years, it is possible that some students fail at a particular course due to non-academic factors such as motivation and environment. Another finding is that the 2nd semester yields a considerably low accuracy since its testing data relies only on a limited number of courses as its training data (about five courses taken on the 1st semester).

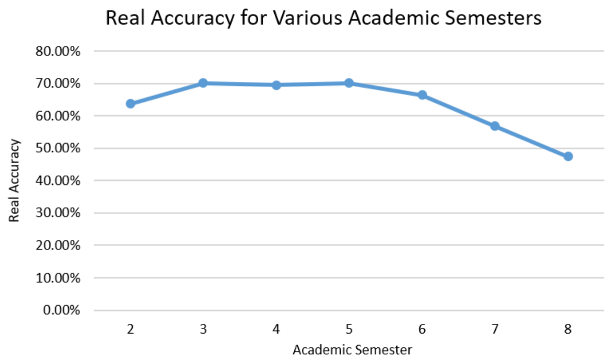

Figure 10. Real accuracy for various academic semesters.

Evaluating The Effectiveness Variation Among Students' Data on Content-Based Filtering

This evaluation measures whether the effectiveness of content-based filtering varies among students' data. To do so, 10-fold cross validation will be conducted for each student's data from imbalance dataset and all results will be displayed as a box-plot. Course recommendation will be performed by content-based filtering with Naive Bayes as its prediction algorithm, top-10 1-gram features taken from X^2 feature selection as its features, and course syllabi & slides as its course data.

Figure 11 shows that the effectiveness varies among students' data with 70% as its average value. The lowest accuracy is 10% while the highest accuracy is 100%. Such variation may occur due to non-academic factors on students or lecturers.

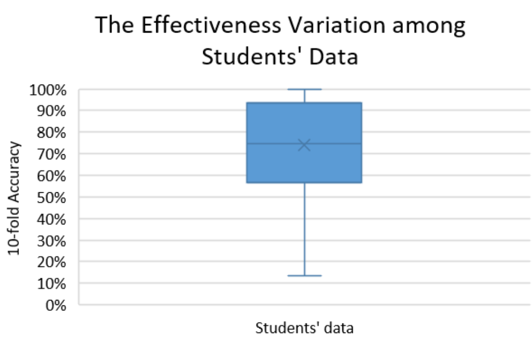

Figure 11. A box-plot depicting effectiveness variation among students' data. Horizontal axis refers to students' data and vertical axis refers to 10-fold accuracy.

Evaluating The Effectiveness of K-Nearest-Neighbours' Number of Neighbours on Collaborative Filtering

K-Nearest-Neighbours' number of neighbours (NN) determines how many closest neighbours used for prediction. Its impact on effectiveness will be measured in this evaluation with real accuracy as its evaluation metric and balance dataset as its evaluation dataset.

Nine first positive integers are used as NN constants; each constant corresponds to one approach. All approaches are derived from collaborative filtering with Naive Bayes as its prediction algorithm, top-10 1-gram features taken from X^2 feature selection as its features, and course syllabi & slides as its course data.

As seen in Figure 12, a slight improvement occurs when NN is increased. This finding strengthens our previous finding from content-based filtering that states more considered neighbours will lead to more considerate prediction.

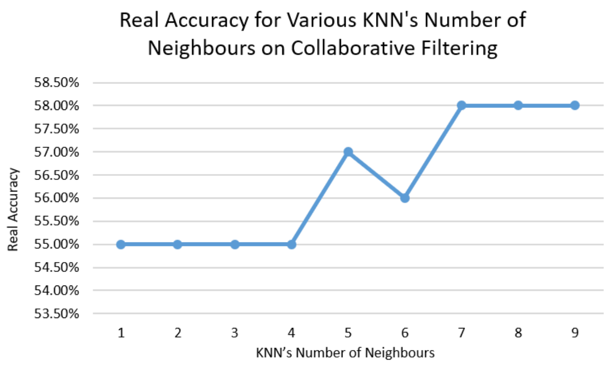

Figure 12. Real accuracy for various KNN's number of neighbours (NN) on collaborative filtering.

Evaluating The Effectiveness of Prediction Algorithm on Collaborative Filtering

Two algorithms used in collaborative filtering will be compared. Each one of them corresponds to an approach that utilises top-10 1-gram features taken from X^2 feature selection as its features and course syllabi & slides as its course data. The first one (called as NB) relies on Naive Bayes as its prediction algorithm while the second one (called KNN) relies on K-Nearest-Neighbours with K=9 (the most effective KNN setting according to previous evaluation). They will be evaluated toward balance dataset with real accuracy as their evaluation metric.

According to our evaluation, NB (63%) performs better than KNN (58%). It generates 5% higher accuracy. Hence, it can be stated that, in our context, Naive Bayes is preferred for collaborative filtering.

According to our evaluation, several findings can be deducted regarding our proposed course recommendation system. First, both content-based and collaborative filtering are considerably effective; their resulted accuracies are about 66% and 58% respectively. Second, X^2 feature selection and K-Nearest-Neighbours' number of neighbours may affect the effectiveness of content-based filtering while stop word removal, prediction algorithm, and n-gram generator may not. Third, term descriptiveness may be not related to accuracy. Fourth, course syllabi alone are sufficient to act as course data for content-based filtering. Fifth, the effectiveness of content-based filtering depends on academic semesters and students' data. Sixth, Naive Bayes is preferred for collaborative filtering. Seventh, K-Nearest-Neighbours' number of neighbours may affect the effectiveness of collaborative filtering.

CONCLUSIONS

This paper presents a course recommendation system dedicated for a student advisor. Using this system, the advisor can guide their students to take courses based on their real skills (captured from the students' grades). It relies on two recommendation techniques: content-based and collaborative filtering. The system is quite effective based on our evaluation.

Our study has a number of limitations that can entail to future work. First, the system is only evaluated at technical level. We plan to use the system in real academic environment and collect feedback from both the advisors and the students. Second, it is unclear whether our system is more effective than existing course recommendation systems in terms of the recommendation technique. We plan to compare those under the same data set. Third, while the effectiveness is relatively high, there is a possibility to improve it further. We plan to integrate course data to collaborative filtering and check whether its effectiveness is improved. Another direction is to extract keyphrases from students' relevant courses using a technique adapted from [46].

REFERENCES

[1] L. Wild and L. Ebbers, "Rethinking Student Retention in Community Colleges," *Community College Journal of Research and Practice*, vol. 26, no. 6, pp. 503–519, Jul. 2002, doi: 10.1080/2776770290041864.

[2] K. S. Hone and G. R. El Said, "Exploring the factors affecting MOOC retention: A survey study," *Computers & Education*, vol. 98, pp. 157–168, Jul. 2016, doi: 10.1016/J.COMPEDU.2016.03.016.

[3] S. de Freitas *et al.*, "Foundations of dynamic learning analytics: Using university student data to increase retention," *British Journal of Educational Technology*, vol. 46, no. 6, pp. 1175–1188, Nov. 2015, doi: 10.1111/bjet.12212.

[4] J. D. Cochran, S. M. Campbell, H. M. Baker, and E. M. Leeds, "The Role of Student Characteristics in Predicting Retention in Online Courses," *Research in Higher Education*, vol. 55, no. 1, pp. 27–48, Feb. 2014, doi: 10.1007/s11162-013-9305-8.

[5] K. Kimbark, M. L. Peters, and T. Richardson, "Effectiveness of the Student Success Course on Persistence, Retention, Academic Achievement, and Student Engagement," *Community College Journal of Research and Practice*, vol. 41, no. 2, pp. 124–138, Feb. 2017, doi: 10.1080/10668926.2016.1166352.

[6] S. Zheng, K. Han, M. B. Rosson, and J. M. Carroll, "The Role of Social Media in MOOCs: How to Use Social Media to Enhance Student Retention," in *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*

 https://doi.org/10.25077/7.02.63-72.2023

- *L@S '16*, 2016, pp. 419–428, doi: 10.1145/2876034.2876047.

[7] E. Elvina, O. Karnalim, M. Ayub, and M. C. Wijanto, "Combining program visualization with programming workspace to assist students for completing programming laboratory task," *Journal of Technology and Science Education*, vol. 8, no. 4, p. 268, Jun. 2018, doi: 10.3926/jotse.420.

[8] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, Jul. 2013, doi: 10.1016/J.KNOSYS.2013.03.012.

[9] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, and F. García-Sánchez, "Social knowledge-based recommender system. Application to the movies domain," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10990–11000, Sep. 2012, doi: 10.1016/J.ESWA.2012.03.025.

[10] E. R. Núñez-Valdéz, J. M. Cueva Lovelle, O. Sanjuán Martínez, V. García-Díaz, P. Ordoñez de Pablos, and C. E. Montenegro Marín, "Implicit feedback techniques on recommender systems applied to electronic books," *Computers in Human Behavior*, vol. 28, no. 4, pp. 1186–1193, Jul. 2012, doi: 10.1016/J.CHB.2012.02.001.

[11] J. J. Castro-Schez, R. Miguel, D. Vallejo, and L. M. López-López, "A highly adaptive recommender system based on fuzzy logic for B2C e-commerce portals," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2441–2454, Mar. 2011, doi: 10.1016/J.ESWA.2010.08.033.

[12] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowledge-Based Systems*, vol. 157, pp. 1–9, Oct. 2018, doi: 10.1016/J.KNOSYS.2018.05.001.

[13] J. Bobadilla, F. Serradilla, and A. Hernando, "Collaborative filtering adapted to recommender systems of e-learning," *Knowledge-Based Systems*, vol. 22, no. 4, pp. 261–265, May 2009, doi: 10.1016/J.KNOSYS.2009.01.008.

[14] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, pp. 12–32, Jun. 2015, doi: 10.1016/J.DSS.2015.03.008.

[15] C. Porcel, A. Tejeda-Lorente, M. A. Martínez, and E. Herrera-Viedma, "A hybrid recommender system for the selective dissemination of research resources in a Technology Transfer Office," *Information Sciences*, vol. 184, no. 1, pp. 1–19, Feb. 2012, doi: 10.1016/J.INS.2011.08.026.

[16] J. Salter and N. Antonopoulos, "CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering," *IEEE Intelligent Systems*, vol. 21, no. 1, pp. 35–41, Jan. 2006, doi: 10.1109/MIS.2006.4.

[17] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, vol. 2009, pp. 1–19, Oct. 2009, doi: 10.1155/2009/421425.

[18] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Information Sciences*, vol. 180, no. 22, pp. 4290–4311, Nov. 2010, doi: 10.1016/J.INS.2010.07.024.

[19] M. G. Vozalis and K. G. Margaritis, "Using SVD and demographic data for the enhancement of generalized Collaborative Filtering," *Information Sciences*, vol. 177, no. 15, pp. 3017–3037, Aug. 2007, doi: 10.1016/J.INS.2007.02.036.

[20] T. Lee, J. Chun, J. Shim, and S. Lee, "An Ontology-Based Product Recommender System for B2B Marketplaces," *International Journal of Electronic Commerce*, vol. 11, no. 2, pp. 125–155, Dec. 2006, doi: 10.2753/JEC1086-4415110206.

[21] A. Nanopoulos, D. Rafailidis, P. Symeonidis, and Y. Manolopoulos, "MusicBox: Personalized Music Recommendation Based on Cubic Analysis of Social Tags," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 407–412, Feb. 2010, doi: 10.1109/TASL.2009.2033973.

[22] L. Terán and A. Meier, "A Fuzzy Recommender System for eElections," in *International Conference on Electronic Government and the Information Systems Perspective*, 2010, pp. 62–76, doi: 10.1007/978-3-642-15172-9_6.

[23] C. Cobos *et al.*, "A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes," *Information Processing & Management*, vol. 49, no. 3, pp. 607–625, May 2013, doi: 10.1016/J.IPM.2012.12.002.

[24] M. C. Wijanto, R. Rachmadiany, and O. Karnalim, "Thesis Supervisor Recommendation with Representative Content and Information Retrieval," *Journal of Information Systems Engineering and Business Intelligence*, vol. 6, no. 2, 2020, doi: http://dx.doi.org/10.20473/jisebi.6.2.143-150.

[25] C. Porcel and E. Herrera-Viedma, "Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries," *Knowledge-Based Systems*, vol. 23, no. 1, pp. 32–39, Feb. 2010, doi: 10.1016/J.KNOSYS.2009.07.007.

[26] S. Schiaffino and A. Amandi, "Building an expert travel agent as a software agent," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1291–1299, Mar. 2009, doi: 10.1016/J.ESWA.2007.11.032.

[27] T. T. S. Nguyen, Hai Yan Lu, and Jie Lu, "Web-Page Recommendation Based on Web Usage and Domain Knowledge," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2574–2587, Oct. 2014, doi: 10.1109/TKDE.2013.78.

[28] J. A. Recio-Garcia, G. Jimenez-Diaz, A. A. Sanchez-Ruiz, and B. Diaz-Agudo, "Personality aware recommendations to groups," in *Proceedings of the third ACM conference on Recommender systems - RecSys '09*, 2009, p. 325, doi: 10.1145/1639714.1639779.

[29] R. Farzan and P. Brusilovsky, "Encouraging user participation in a course recommender system: An impact on user behavior," *Computers in Human Behavior*, vol. 27, no. 1, pp. 276–284, Jan. 2011, doi: 10.1016/J.CHB.2010.08.005.

[30] R. Farzan and P. Brusilovsky, "Social Navigation Support in a Course Recommendation System," in *International Conference on Adaptive Hypermedia and*

Adaptive Web-Based Systems*, 2006, pp. 91–100, doi: 10.1007/11768012_11.

[31] C.-M. Chen, H.-M. Lee, and Y.-H. Chen, "Personalized e-learning system using Item Response Theory," *Computers & Education*, vol. 44, no. 3, pp. 237–255, Apr. 2005, doi: 10.1016/J.COMPEDU.2004.01.006.

[32] X. Li, T. Wang, H. Wang, and J. Tang, "Understanding User Interests Acquisition in Personalized Online Course Recommendation," in *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, Jul. 2018, pp. 230–242, doi: 10.1007/978-3-030-01298-4_20.

[33] J. Xiao, M. Wang, B. Jiang, and J. Li, "A personalized recommendation system with combinational algorithm for online learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 3, pp. 667–677, Jun. 2018, doi: 10.1007/s12652-017-0466-8.

[34] M. P. O'Mahony and B. Smyth, "A recommender system for on-line course enrolment: an initial study," in *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07*, 2007, p. 133, doi: 10.1145/1297231.1297254.

[35] M. K. Khribi, M. Jemni, and O. Nasraoui, "Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval," in *2008 Eighth IEEE International Conference on Advanced Learning Technologies*, 2008, pp. 241–245, doi: 10.1109/ICALT.2008.198.

[36] K. I. Bin Ghauth and N. A. Abdullah, "Building an E-learning Recommender System Using Vector Space Model and Good Learners Average Rating," in *2009 Ninth IEEE International Conference on Advanced Learning Technologies*, Jul. 2009, pp. 194–196, doi: 10.1109/ICALT.2009.161.

[37] A. Klašnja-Milićević, B. Vesin, M. Ivanović, and Z. Budimac, "E-Learning personalization based on hybrid recommendation strategy and learning style identification," *Computers & Education*, vol. 56, no. 3, pp. 885–899, Apr. 2011, doi: 10.1016/J.COMPEDU.2010.11.001.

[38] D. Wen-Shung Tai, H. Wu, and P. Li, "Effective e-learning recommendation system based on self-organizing maps and association mining," *The Electronic Library*, vol. 26, no. 3, pp. 329–344, Jun. 2008, doi: 10.1108/02640470810879482.

[39] Y. Shinyama, "PDF Miner," 2021. https://github.com/euske/pdfminer.

[40] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[41] H. A. Robbani, "PySastrawi," 2021. https://github.com/har07/PySastrawi.

[42] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[43] T. Mitchell, *Machine Learning*. McGraw-Hill Education, 1997.

[44] G. J. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing Microarray Gene Expression Data*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004.

[45] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[46] F. C. Jonathan and O. Karnalim, "Semi-Supervised Keyphrase Extraction on Scientific Article using Fact-based Sentiment," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 4, pp. 1771–1778, Aug. 2018, doi: 10.12928/TELKOMNIKA.V16I3.5473.

AUTHORS BIOGRAPHY

Muftah Afrizal Pangestu

Muftah completed his bachelor's degree at Maranatha Christian University, Indonesia and his master's degree at The University of Newcastle, Australia. He currently lives in Fukuoka, Japan and works in there.

Oscar Karnalim

Oscar completed his bachelor's degree at Parahyangan Catholic University, Indonesia, his master's degree at Bandung Institute of Technology, Indonesia, and his Ph.D. at The University of Newcastle, Australia. He is now the head of Master Program in Computer Science at Maranatha Christian University, Indonesia. His research interests are educational computing and artificial intelligence.

 https://doi.org/10.25077/7.02.63-72.2023