SDCA Asia-Pacific Multidisciplinary Research Journal
Volume 5, Issue 1, June 2023

Structuring topics of Philippine universities' introductory programming courses using semi-supervised pairwise-constrained clustering to synthesize alternative course topic outlines

Kleb Dale G. Bayaras

Abstract

In course design, topic outline organization encompasses the structuring and sequencing of topics to be delivered in a learning environment. Recent studies in topic outline optimization revolve around massive open online courses (MOOCs) due to their abundance but not much has been studied on the traditional courses. This study investigates the organization of topic outlines in traditional introductory programming courses across Philippine higher education institutions (HEIs) and evaluates the viability of the synthesized alternative topic outlines. Course syllabi were collected from 16 HEIs. A topic precedence graph (TPG) model that provides a structured overview of the introductory programming was created via a semi-supervised pairwise constrained k-means (PCK-Means) clustering to structure the topics which produced 20 topic clusters with strong topic cohesion within the clusters. The TPG showed that HEIs tend to start the outline similarly, followed by core programming topics with varied sequences, and divergent ways of ending the outline. Two anomaly clusters were identified as having topic titles grouped that do not seem to have a unifying topic. Limitations of the clustering algorithm are identified where it cannot identify semantic meaning between words which may affect its applicability in situations where topic titles are named inconsistently. From the TPG, alternative optimal and comprehensive topic outlines were synthesized via greedy and DFS graph traversal algorithms. However, these alternative outlines performed very poorly when compared with the evaluators' (n=19) arrangement of topic outlines due to some prerequisite topics being discussed in the latter part already. Overall, this study introduces a method to incorporate computer science technologies in structuring topics across HEIs and aiding educators in topic outline design but more research is needed before it can be implemented in a real classroom setting.

Keywords: Alternative topic outlines; topic precedence graph; pairwise constrained clustering; traditional course sequencing.

Kleb Dale G. Bayaras*
Department of Information Systems and Computer Science, Ateneo de Manila University, Quezon City, Philippines
E-mail: kleb.bayaras@obf.ateneo.edu

***Corresponding author**

To cite this article:
Bayaras, K. D. G. (2023). Structuring topics of Philippine universities' introductory programming courses using semi-supervised pairwise-constrained clustering to synthesize alternative course topic outlines. *SDCA Asia-Pacific Multidisciplinary Research Journal*, 4(1), 27-35.

doi: 10.5281/zenodo.8031230

Introduction

The quality of education is a complex and challenging issue, but educators, researchers, and educational theorists agree that well-crafted and well-designed courses are a significant factor in improving education (Black et al., 2014; McNeil, 2014). Course design includes the organization of course resources, instructional methodologies, course schedule, structure, contents, topic organization, and the planning of the whole course. A well-designed course has a positive impact on students, motivating and satisfying them, and improving their learning outcomes (Almaiah & Alyoussef, 2019; Mtebe & Raisamo, 2014). It also benefits teachers, reducing their stress and promoting flexibility, confidence, and innovation (Kyriacou, 2019).

Course design is still a broad issue to address. This paper focuses on the sub-problem of topic outline organization (also referred to as a study plan or lesson plan organization) and investigates the topic outline organization of traditional Introductory Programming courses. Traditional courses, in this context, are educational courses offered by Philippine HEIs that are approved by their respective department or university heads and are delivered via face-to-face instruction or distance learning to students who enrolled in the said course in a scope of a semester. This is to differentiate traditional courses from other educational courses like MOOCs or custom learning plans by personal tutors.

In the context of the Philippines, Commission on Higher Education (CHED) oversees higher education programs and institutions in the Philippines, setting policies, standards, and guidelines for their creation. CHED Memorandum Order (CMO) No. 25, Series of 2015 (Commission on Higher Education, 2015) specifies the requirements for Bachelor of Science in Computer Science (BSCS), Bachelor of Science in Information Systems (BSIS), and Bachelor of Science in Information Technology (BSIT) programs. The CMO outlines the competencies, number of units, and learning outcomes for the courses, but leaves the creation and organization of topic outlines to the discretion of the HEIs and educators. Educators have the academic freedom to decide which topics to include in a course, as long as the course learning outcomes are achieved. It has also been shown that HEIs are also more likely to see an increase in student engagement and learning with well-designed and well-structured courses (Ekren, 2017) which is a goal that HEIs always strive for.

This paper acknowledges that most of the literature about optimization of topic sequences pertains to MOOCs and other online courses but argues that while the context of online education warrants different quality indicators compared to traditional ones due to people not trusting the quality it offers for now (Ekren, 2017), in terms of course content or topic organization, the optimization and evaluation methods can be adapted for traditional courses since its examination of how the topics are organized is independent of the learning context.

It draws from existing introductory programming courses' topic outlines from Philippine HEIs and considers similarity and precedence relationships between topics and subtopics to generate a model that shows the structure of the introductory programming knowledge domain and used to synthesize new course topic outlines.

Statement of the Problem

Designing an introductory programming course topic outline involves deciding which topics to include, grouping them based on similarity and interdependencies, and determining the sequence of topics to be taught based on prerequisite knowledge. This manual process can be time-consuming and can lead to random topic sequences. Despite technology's integration into education, course design remains a mixture of manual labor and art (Agrawal et al., 2016; Bain, 2020) that can lead to suboptimal outcomes. For example, some topics may be discussed before students have the prerequisite knowledge, which can increase their cognitive load and make the course seem more difficult (Vuong et al., 2011).

To address these challenges, the paper proposes automating the topic organization process by analyzing existing courses and identifying patterns of topic similarity and precedence. By generating a directed weighted graph model of the introductory programming knowledge domain, the authors aim to provide a method that can be incorporated into systems for educators to optimize their course outlines and improve student learning outcomes. The model can help in revealing structures that are not readily apparent in introductory programming courses across different HEIs by capturing similarities and variations in their topic outlines. It is also useful for analyzing and generating new topic outlines. Ultimately, the paper seeks to reduce the time and resource-intensive process of course design and help educators create well-structured and engaging courses for their students.

Research Questions

To address the problems stated above, this paper aims to answer the following questions:

RQ1. How do we construct a topic precedence model based on the topic outlines of traditional introductory programming courses?
RQ2. What do the topic outline structures of traditional introductory programming courses look like?
RQ3. Are the synthesized course topic outlines viable as an actual introductory programming course topic outline?

Scope and Limitations

This study focuses only on the organization of topic outlines in traditional introductory programming courses from Philippine HEIs. It assumes that the existing topic outlines are of good quality and the order of topics serves as the ground truth for the construction of the graph. The model presented will synthesize alternative course topic outlines by maximizing topic similarity and precedence relationships. However, external factors like student performance analysis and actual implementation of the synthesized topic outlines in a classroom setting are not included in this study.

Methodology

Dataset

A total of 16 course syllabi were collected from existing introductory programming courses of Philippine HEIs that were used in the school year 2018-2019 onwards. Out of the 16 course syllabi, six (6) were from private HEIs and 10 were from state universities and colleges. This number of collected syllabi was similar to (Alsaad et al., 2021) which gathered 10-21 courses to structure a robust and viable topic transition map from MOOCs for Machine Learning, Structured Query Language (SQL), and Python.

The study considered the course title, course description, and course topic outline for the analysis. The course title and description were used to verify if a course fits the criteria of an introductory programming course, while the topic outline was used to extract topic titles and sub-topics. Lecture transcripts were not available, so only the topic titles and sub-topics were used. The extracted information was encoded into a CSV file and used for preprocessing and clustering.

Topic Title Extraction and Vector Representation

The 129 extracted topic titles in the course syllabi were preprocessed by manual lemmatization by transforming different inflections of a word into a singular form (i.e., Algorithmic, Algorithms, Algorithmically into Algorithm) and removing stopwords. The preprocessed titles were then tokenized to form a vocabulary consisting of 338 unique preprocessed words.

Prior to topic clustering, topic titles must first be represented in a mathematical model to be used for calculations and quantifiable comparisons. Various approaches have been used for this but has shown that clustering algorithms performed better with bag-of-words (BoW) and term frequency-inverse document frequency (TF-IDF) sparse vector representations compared to dense vector representations (Alsaad & Alawini, 2020; Shen et al., 2015). In this study, the topic titles were represented by a sparse normalized bag-of-words vectors using the Scikit-learn library (Pedregosa et al., 2011).

Topic Title Clustering

Topic clustering was used to group similar topic titles into a single cluster that represents a single topic or a set of closely related topics. This was achieved by computing the cosine similarities of the topic titles' vector representation and using the Pairwise-Constraint K-Means (PCK-Means) clustering algorithm. The number of clusters was determined by maximizing the silhouette coefficients of varying similarity thresholds and it should be at least the length (in terms of number of topic titles) of the shortest course topic outline and at most 1.5 times the length of the longest one.

Cosine similarity, a measure of similarity between two vectors, was used to determine the similarity between topic titles. It was especially useful in this case since the cosine similarity does not depend on the magnitude or the length of the topic title. The PCK-Means constraints – Must-Link and Cannot-Link tuples – were formed using the cosine similarity to guide the clustering process.

The PCK-Means clustering algorithm is a modified K-means algorithm with added constraints (Basu et al., 2004). It considers the distance between points and the pairwise constraints to guide the clustering process. PCK-Means was used instead of the regular K-means since the aim was to cluster topics from different courses and avoid topics within the same course being clustered together as was used in Alsaad and Alawini (2020). A python implementation of the algorithm by Babaki (2017) was used in this study.

The Must-Link constraint tells which topic pairs should be clustered together, while the Cannot-Link constraint tells which topic pairs should not be clustered together. The adjacent topic titles and topic titles within the same course were under the Cannot-Link list by default to avoid clustering topic titles within just one course and find topic titles with similar topics from different courses. However, there may exist adjacent topic title tuples that surpass the similarity threshold and should be in the Must-Link list. The set closure of the Must-Link tuples was also removed from the Cannot-Link list. Topic titles in neither list were treated as they were. The output of this step was an empty graph with nodes as a clustering of topic titles with similar topics but no edges yet.

Construction of Topic Precedence Graph (TPG)

The resulting topic title clusters from the PCK-means clustering of the topic titles were considered as nodes and linked them with each other by leveraging the original precedence order of the topics in their respective original topic outlines as the directed edges. This method captured the likelihood of the transition of topics and ensured that no cluster remained unlinked. The researcher assumed that the existing courses' topic outline's precedence order maintained the prerequisite relationships between them since they were crafted by experts in their respective HEIs (Manrique et al., 2018). The utilization of the existing courses' precedence was also in line with the axioms of prerequisite compliance and locality of reference (Agrawal et al., 2016) to reduce students' comprehension burden. For the edge weights, it was the accumulated frequency of the links which was similarly used by Alsaad et al. (2021). The output of this step was a Topic Precedence Graph – a directed graph where nodes were connected, and the edges' directions and weights were leveraged from the original precedence graphs of the topic outlines.

Synthesis of Alternative Topic Outlines

The clustering of topic titles into nodes and linking them based on their precedence relationships has resulted in a topic precedence graph. The clusters themselves are a group of similar topic titles, and can be considered as a single unit with a unified topic, forming a basis for a topic title when synthesizing a topic outline. The topic outline can be represented as an ordered set of nodes in a path, and traversing the graph based on a chosen algorithm results in an ordered set of nodes and edges that form the synthesized topic outline. The traversal algorithms used in this study are greedy algorithm and depth-first search algorithm to synthesize optimal and all-topics alternative topic outlines, respectively.

To synthesize an optimal alternative topic outline, a graph traversal using a greedy algorithm was utilized. The root node will be chosen based on the node cluster with the highest number of starting topic titles, and the algorithm will pick the next unvisited node with the heaviest edge weight which corresponds to the most connections from the original topic outlines. In case of a tie, the next node will be randomly chosen. The algorithm picks the next node based on the edge with the highest weight, indicating the most likely topic to be discussed next. The traversal stops when it reaches a node with no more transitions.

To synthesize an all-topics alternative topic outline, a Depth-First Search algorithm starting from the root node with the highest frequency of starting topic titles was utilized. However, this traversal only considers the existence and direction of edges in between nodes and not their edge weights, and backtracks when it cannot find a transition to an unvisited node which may not produce the most optimal topic sequence with respect to the accumulated frequency of precedence. Nevertheless, it provides a comprehensive topic outline that covers all topics in the model created, although it may not fit a traditional topic outline due to time and length limitations.

Evaluation Designs

Educators who taught introductory programming and undergraduate students who recently completed an introductory programming course were recruited to evaluate the quality of the topic title clusters and synthesized alternative topic outlines. The evaluation aimed to assess the topic cohesion of each cluster, the quality of topic sequencing of the synthesized alternative topic outlines, and to identify similarities between the evaluators' own outlines and the synthesized outlines. These questions were asked to the evaluators:

Q1: In each topic cluster produced by the PCK-Means algorithm, identify the odd topics that you think should not be part of this topic cluster. If you think all topics are related in a topic cluster, choose the option "*No Odd Topics*."
Q2: Given the same topic clusters, in which order would you present these topics in an introductory programming course you believe is the most optimal and respect the prerequisite relationship of the topics?
Q3: For each topic outline shown, do you agree that the topic outline is a viable and helpful topic outline for an introductory programming course and can be used in a classroom setting? (1 = strongly disagree; 5 = strongly agree).

These evaluation questions were adapted from the studies of Agrawal et al. (2016), Rüdian and Pinkwart (2021), and Shen et al. (2015) that similarly evaluated the quality of their automated models that also produced topic clusters and topic cluster sequences.

Results and Analysis

Dataset Summary

Sixteen (16) introductory programming course syllabi from different Philippine HEIs were collected, with varying count of topic titles. Some syllabi contained as few as five (5) topic titles, while others had as many as 17. The level of detail also varied, with some outlines having only main topic titles while others had detailed subtopic lists.

The level of detail also varied, with some outlines having only main topic titles while others had detailed subtopic lists. Table 1 provides a statistical summary of the topic titles in the syllabi, with the length of an outline measured by the number of topic titles or clusters.

Table 1. Statistical summary of the collected introductory programming syllabi

Category	Number of Topic Titles
Minimum Length	5
Maximum Length	17
Mean Length	8.0625
Mode Length	6
Mode Count	6

Preprocessing Results

From the 16 collected course syllabi, 129 topic titles were extracted and preprocessed. The preprocessing comprised two stages: stopwords removal and manual lemmatization. There were 538 raw unique words before preprocessing. After preprocessing, there were 338 unique words considered and utilized to create the sparse normalized BoW vector representations of the topic titles. For topic outlines that contain sub-topics, these were merged with the main topic title and are considered a single topic title for this study.

Topic Title Clustering

The final similarity threshold value used for clustering was 0.75 after testing various values. Lower values caused disproportional cluster sizes, while higher values created too many clusters. The best cluster result was chosen by maximizing the Silhouette Coefficient clustering measure for each value. The PCK-means clustering algorithm produced 20 topic title clusters. Due to space limitations, Table 2 only shows the top three (3) words and the summarized topic title cluster based on the top words per cluster. This also allowed easy classification and ease of discussion for each cluster.

Topic Precedence Graph

The topic precedence graph (TPG) in Figure 1 was the resulting graph after linking the topic title clusters with respect to their original precedence graphs. The TPG visualized the similarities and variations of topics based on the topic title clusters and topic order across different HEIs. The letters in the nodes are topic title cluster letters shown in Table 2.

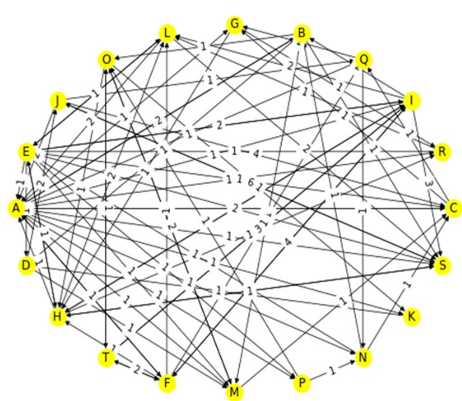

Figure 1. Topic precedence graph

Table 2. Summarized topic title cluster titles based on the top three (3) terms in their respective clusters

Cluster	Top Words	Summarized Topic Cluster Titles
A	function, defined, user	Functions
B	structure, sequential, expression	Sequential Control Structures
C	array, dimensional, one	Arrays
D	testing, debugging, techniques	Testing, Debugging, & Optimization Techniques
E	structure, statement, if	Control & Selection Structure
F	programming, introduction, language	Introduction to Programming & Programming Concepts
G	pointers, function, definition	Pointers
H	operator, logical, assignment	Logical Operators & Assignment Statement
I	data, type, program	Data Types, Identifiers, Keywords, & Variables
J	loop, while, iteration	Loops & Iterative Constructs
K	package, java, handling	Java Applet & Error Handling
L	input, function, output	Input/Output & File Processing
M	structure, repetition, control	Repetition Control Structures
N	classes, object, storage	Objects & Classes
O	java, variable, package	Java Variables & Packages
P	tree, search, recurrence	Search Tree & Recurrence Tree
Q	recursion, case, recursive	Recursion
R	string, function, dimensional	Strings
S	statement, if, else	Conditional Statements & Constructs
T	algorithm, flowchart, pseudocode	Flowchart, Pseudocode, & Algorithms

Structure of Traditional Introductory Programming Courses

This section analyzes the structure of traditional introductory programming courses using the results of PCK-Means clustering algorithm and the TPG. This also addresses the RQ2: "*How do the topic outline structures of traditional introductory programming courses look like?*"

To give an overview of the structure of the traditional introductory programming courses and their characteristics across the Philippine HEIs, the most common starting point was embodied by cluster F "*Introduction to Programming & Programming Concepts*." The middle part tends to cover the core topics (clusters E, I, J, and S) and is similar across the HEIs but were sequenced differently. On the other hand, the end topic (clusters A, C, L, and Q) tends to be more diverse across the different outlines, and this is where the sequence usually splits and differs.

Synthesized Alternative Topic Outlines

The TPG's edges represented cluster precedence and was used as basis in the synthesis of topic outlines. The outlines were obtained by examining a path of ordered nodes in the TPG, with the starting node being the one with the most starting topic titles, which was cluster F entitled "*Introduction to Programming & Programming Concepts*."

The synthesized optimal and all-topics alternative outlines started with cluster F and used the greedy and DFS graph traversal algorithms respectively. Tables 3 and 4 show the resulting outlines with their summarized titles.

Table 3. Synthesized optimal alternative topic outline with summarized titles

Order	Cluster	Summarized Topic Cluster Titles
1	F	Introduction to Programming & Programming Concepts
2	I	Data Types, Identifiers, Keywords & Variables
3	S	Conditional Statements & Constructs
4	J	Loops & Iterative Constructs
5	C	Arrays
6	A	Functions
7	B	Sequential Control Structures
8	M	Repetition Control Structures
9	O	Java Variables & Packages
10	K	Java Applet & Error Handling
11	D	Testing, Debugging, & Optimization Techniques

Table 4. Synthesized all-topics alternative topic outline with summarized titles

Order	Cluster	Summarized Topic Cluster Titles
1	F	Introduction to Programming & Programming Concepts
2	I	Data Types, Identifiers, Keywords & Variables
3	B	Sequential Control Structures
4	O	Java Variables & Packages
5	H	Logical Operators & Assignment Statement
6	R	Strings
7	L	Input/Output & File Processing
8	E	Control & Selection Structure
9	C	Arrays
10	Q	Recursion
11	G	Pointers
12	N	Objects & Classes
13	A	Functions
14	P	Search Tree & Recurrence Tree
15	K	Java Applet & Error Handling
16	D	Testing, Debugging, & Optimization Techniques
17	J	Loops & Iterative Constructs
18	S	Conditional Statements & Constructs
19	M	Repetition Control Structures
20	T	Flowchart, Pseudocode, & Algorithms

Viability of Topic Clusters and Synthesized Topic Outlines

This section discusses the evaluators' feedback on the viability of the topic title clusters and synthesized outlines. Evaluators were asked three questions via one-on-one guided online interviews. The questions evaluated cohesion, order quality, and viability of the outlines. Their comments and inputs were also solicited during the interviews. This analysis addresses RQ3: "Are *the synthesized course topic outlines viable as an actual introductory programming course topic outline?*"

Evaluators' Demographics

The study recruited 19 evaluators, including 12 educators with different degrees (ranging from bachelor's to Ph.D.) and seven (7) freshmen students who had completed their introductory programming course in the previous semester. Table 5 shows the evaluators' demographic data.

Table 5. Evaluators' demographical data

Category	Educators	Students	All Evaluators
Count	12	7	19
Male	8	5	13
Female	4	2	6
HEI-Private	6	2	8
HEI-SUC	6	5	11

Evaluation for Topic Cohesion in Each Cluster

Evaluation question 1 evaluated topic cluster cohesion and the quality of the PCK-Means Algorithm in clustering the topic title vectors. Evaluators used Krippendorff's alpha (α) to measure inter-rater agreement (Krippendorff, 1970). Overall agreement was α = 0.394, indicating imperfect but fair agreement. Educators had α = 0.394 and students had α = 0.461, indicating fair and moderate agreement, respectively.

To also identify evaluators who evaluated the topics either too harshly or too leniently, an analysis of variance (ANOVA) was performed on the count of odd topics for each topic title cluster as rated by the evaluators. As shown in Figure 2, the majority of the count of odd topics per cluster does not differ significantly from all the evaluator's ratings. However, there are outliers for clusters G, N, P, and R. Removing these outliers resulted in a slight increase of the overall agreement α = 0.462 which indicates moderate agreement.

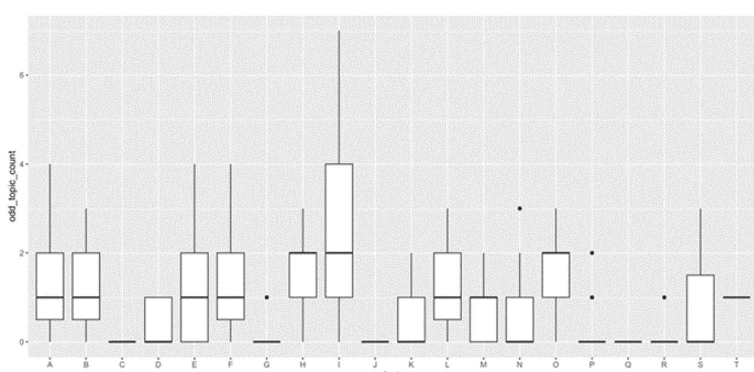

Figure 2. ANOVA results for evaluation question 1

Table 6 summarizes the results of evaluation question 1. It shows the number of to topic titles per cluster, minimum, maximum, and mean of the number of odd topics identified by the evaluators. The percentage of mean odd topics is the mean divided by the number of topic titles in the cluster. The weighted percent of mean odd topics was derived by multiplying the percent of mean odd topics by the number of titles in the cluster and then dividing by the total topic titles which was 129. The last column shows the count of odd topics with a majority evaluators vote (n>=9).

Table 6. Summary results for evaluation Q1 (count of odd topics)

Cluster	Num of Titles	Min	Max	Mean	% Mean Odd Topics	Weighted % Mean Odd Topics	Count of Odd Topic Titles (n>=9)
A	13	0	4	1.32	10.15%	1.02%	2
B	6	0	3	1.21	20.17%	0.94%	1
C	8	0	0	0	0.00%	0.00%	No Odd Topics
D	4	0	1	0.32	8.00%	0.25%	No Odd Topics
E	9	0	4	1.42	15.78%	1.10%	1
F	10	0	4	1.42	14.20%	1.10%	1
G	3	0	1	0.11	3.67%	0.09%	No Odd Topics
H	7	0	3	1.58	22.57%	1.22%	1
I	15	0	7	2.58	17.20%	2.00%	2
J	9	0	0	0	0.00%	0.00%	No Odd Topics
K	2	0	2	0.68	34.00%	0.53%	No Odd Topics
L	7	0	3	1.32	18.86%	1.02%	1
M	4	0	2	0.79	19.75%	0.61%	No Majority Vote
N	3	0	3	0.63	21.00%	0.49%	No Odd Topics
O	5	0	3	1.53	30.60%	1.19%	2
P	2	0	2	0.26	13.00%	0.20%	No Odd Topics
Q	4	0	0	0	0.00%	0.00%	No Odd Topics
R	3	0	1	0.11	3.67%	0.09%	No Odd Topics
S	10	0	3	0.63	6.30%	0.49%	No Odd Topics
T	5	1	1	1	20.00%	0.78%	3

Nine (9) out of the 20 topic clusters (C, D, G, J, N, P, Q, R, S) have been identified by a majority of the evaluators as having no odd topics. This meant that these topic clusters had a cohesive set of topic titles in their respective clusters as evaluated and rated by the evaluators. 6 topic clusters (B, E, F, H, L, T) have been identified by a majority of the evaluators as having one odd topic title and 3 topic clusters (A, I, O) as having two odd topic titles. This does not sum up to 20 topic clusters as 2 topic clusters are anomalous (K and M).

While cluster K garnered a majority vote of "No Odd Topics", it also contained just two (2) topic titles. 10 evaluators identified the cluster as having no odd topics, seven (7) evaluators identified topic title 1 (Java Applet) and topic title 2 (Error Handling Routines) as odd topics. Four (4) evaluators also identified the two topic titles as odd since they noted that the topic titles should not be together in the cluster; thus, both of them are odd. With just two topic titles, it was also difficult to single out which one is the odd topic title; this is especially highlighted with the titles discussing entirely different topics.

Cluster M was also anomalous as it garnered to majority vote and the identification of odd topic in this cluster was spread out. This was due to how the topic titles in this cluster was grouped. While cluster M was about Repetition Control Structures, two (2) out of the four (4) topic titles were "*Flowchart; Understanding Structure, Module, Hierarchy Charts, and Documentation, Decision, Loop*" and "*Program Design and Structure; Programming Fundamentals, Comments, Statement, Identifier, Keyword, Literals, Primitive Data Type, Variable, Output Data, Operator, Getting Input from the Keyboard, Control Structure, Decision, Repetition, Branching, Command Line Arguments*." One factor that these two topic titles that were not about repetition control structure were grouped in this cluster was the presence of the word "structure" and "loop" in their topic titles, especially in their sub-topic titles.

This odd inclusion of the topic titles in a cluster is one of the disadvantages of the PCK-means clustering algorithm. An important finding when using the PCK-Means clustering method on normalized BoW representations of topic titles is that it tends to interpret words in a very literal sense, without considering their context or semantic similarities and differences to neighboring words in the title. Although this approach was effective in grouping topic titles based on their shared words, it was unable to capture the nuances of language and the semantic meanings conveyed by the topic titles.

The evaluation showed that the topic clusters had a strong level of cohesion with only slight imperfections, as evidenced by a total weighted percentage mean of odd topics at 13.11%. However, evaluators also noted that some topic titles were considered odd due to the presence of a commonly repeated word in the cluster, highlighting the algorithm's inability to distinguish semantic similarities and differences between words. Despite this limitation, the evaluators expressed appreciation for the algorithm's ability to cluster topics automatically without any manual intervention by an expert.

Evaluation of the Quality of the Synthesized Topic Outlines

Evaluation question 2 asked the evaluators to arrange their own topic outlines given the topic title clusters produced by the PCK-means with their summarized cluster titles. The goal here was to do a side-by-side comparison with the synthesized outlines (optimal and all-topics topic outlines) and the evaluators' topic outlines to gauge their quality.

To determine the consistency between the synthesized topic outlines and the evaluators' outlines, a one-to-one matching approach was employed. This method involved comparing the placement of topic clusters in the sequence of the topic outlines created by the evaluators and the synthesized outlines. The findings revealed that only the first topic cluster, topic cluster F, had consistent placement across 18 evaluators for both optimal and all-topics outline.

While the evaluators' topic outlines showed varying arrangements, certain patterns and subsequences emerged, indicating common ordering of topic titles among them. These subsequences were labeled as Subsequences A, B, and C, with Subsequence A [F, T] appearing in 12 outlines, Subsequence B [F, T, I] appearing in 9 outlines, and Subsequence C [T, I, H] appearing in six (6) outlines. Cluster F focused on "*Introduction to Programming & Programming Languages and Concepts*," cluster T dealt with "*Flowchart, Pseudocode, & Algorithms*," cluster I was about "*Data Types, Identifiers, Keywords, & Variables*," and cluster H was about "*Logical Operators & Assignment Statement*." Furthermore, the mode of topic title clusters from the evaluators' topic outlines for the first four topic clusters [F, T, I, H] was a combination of these subsequences, as shown in Table 7. Some clusters were repeated in the mode of evaluators' topic outlines since it represents the most frequently repeated cluster for that particular order in the sequence. These subsequences were used as an additional factor to gauge if the synthesized outlines will make sense in a classroom scenario.

Table 7. Comparison of the synthesized and the mode of evaluators' outlines

Topic Outlines	Topic Title Cluster Sequence																			
Optimal	F	I	S	J	C	A	B	M	O	K	D									
All-Topics	F	I	B	O	H	R	L	E	C	Q	G	N	A	P	K	D	J	S	M	T
All Evaluators (Mode)	F	T	I	H	S	S	M	S	J	C	G	B	G	K	N	P	N	Q	P	Q
Educators Only (Mode)	F	T	I	H	S	J	C	A	Q	D	G	D	L	O	P	Q	N	G	P	Q
Students Only (Mode)	F	I	A	E	R	S	M	S	H	L	M	C	C	D	N	O	K	O	R	

An investigation into the presence of longest common subsequence (LCS) was also conducted in the topic outlines of educators, students, and the outline mode for all evaluators. An LCS refers to the longest subsequence that is common to all the sequences in a set of sequences. In this investigation, the LCSs identified may serve as a guide for the relative placement of topics in an outline, but not necessarily the exact topic sequence. The criteria for a subsequence to be considered an LCS is that it must be shared by the majority of the group size, and the study used a combination of the group size and majority number to find possible outline combinations. The majority number for educators (n=12) was seven (7), resulting in 729 possible combinations, while for students (n=7), the majority number was four (4), resulting in 35 possible combinations. Table 8 presents the topic cluster node LCS for the different evaluator groups.

The emergence of these LCS among the evaluators' outlines indicates that the relative ordering of topics is still significant, despite the variations in the overall arrangement of the outlines. Take, for instance, the LCS "FIHJ" found in the educators' outlines. It implies that, in the majority of the educators' outlines, topic cluster F was typically positioned before I, followed by H and then J. Since these are LCS, other topic clusters could have been inserted between these LCS topic cluster nodes.

The evaluators' own topic outline arrangements revealed certain patterns and subsequences, which were not found in any of the synthesized outlines. In the optimal outline, only two (2) out of the 11 topic clusters had a one-to-one match, and no subsequences were present. Similarly, in the all-topics covered outline, only two (2) out of the 20 topic clusters had a one-to-one match, and no subsequences were present.

While most of the LCS topic cluster sequences were observed in both optimal and all-topics outline, they cannot be considered strong indicators of a well-arranged topic outline since they only depict the relative ordering of topic clusters.

Table 8. Longest common subsequences (LCS) in the evaluators' outlines grouped by educators, students, and the mode outlines

Educators	Students	Mode Outline
FIHJ	FALSM	FIHCN
FIHJR	FIHB	
FIHSJ	FIHJC	
FIHSJR	FIHSM	
FTHJ	FISCGO	
FTHJR	FISJCN	
FTIHSJ	FISM	
FTIJ	FISMB	
FTJ	FISMC	
	FISMK	
	FISMQ	
	FRSM	
	FTAL	
	FTSM	

Evaluation for Viability of the Synthesized Topic Outlines

Evaluation question 3 asked the evaluators to rate the viability of synthesized topic outlines for an introductory programming course on a scale of Strongly Disagree (1) to Strongly Agree (5). The first outline presented was the optimal alternative topic outline while the second outline used the same topic clusters but was ordered randomly. The third outline was the all-topics alternative topic outline. During the evaluation, the outlines were not labelled to prevent bias. Figure 3 shows the summary of evaluation question 3.

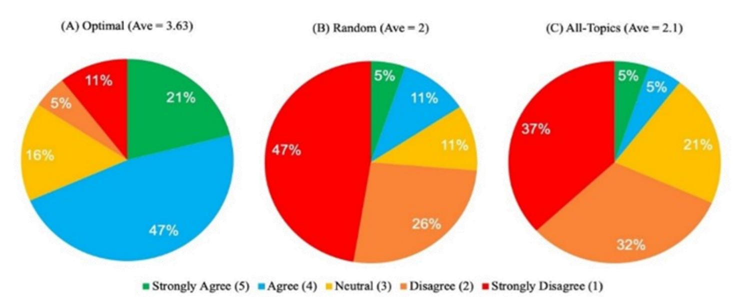

Figure 3. Likert scale summary of the viability of the synthesized topic outlines

The optimal topic outline received an overall positive rating of 68% (21% Strongly Agree and 47% Agree) and an overall average score of 3.63, leaning towards an Agree rating. Although it may be a viable starting point for an introductory programming outline, the evaluators noted that the optimal topic outline became specific for Java programming language in the later parts of the outline, which may require modification if another programming language were to be used.

On the other hand, the randomized topic outline received an overall negative rating of 73% (47% Strongly Disagree and 26% Disagree) and an overall average score of 2. Only one educator rated it Strongly Agree, implying that the topic title sequence is crucial, even if the topic titles are deemed optimal or appropriate for the knowledge domain.

The all-topics topic outline received an overall negative rating of 69% (37% Strongly Disagree and 32% Disagree) and an overall average score of 2.1. While it was the most comprehensive in terms of topic coverage, it may not fit into a single semester of an introductory programming course due to its length and may cause information overload to the students. However, the major observation of the evaluators was not due to the length but more to the order of the topics. They noted that the basic, general, and fundamental topics are discussed very late into the topic outline, which may not be beneficial for students.

Conclusion

The aim of this study was to structure the topics of traditional introductory programming courses across different Philippine HEIs. The study used PCK-Means to cluster 129 topic titles into 20 topic clusters, which showed strong cohesion but with some imperfections due to the clustering algorithm's limitations. The constructed TPG model provided a structured overview of the landscape of introductory programming in the Philippines, revealing that most HEIs' traditional introductory programming topic outlines tend to start the same, while the middle part has core programming topics that are sequenced differently, and the latter part has divergent ways of ending their outlines.

Although the clustering provided good topic cohesion, the synthesis for alternative topic outlines from the TPG model performed poorly, as the synthesized optimal and all-topics covered outlines were not ideal. Since these graph traversal algorithms heavily rely on linking the topic title clusters (edges), this also meant that the linking of the topic title clusters just via their original precedence order on their source topic outlines was not enough to base the transitions between topic clusters. This meant that future researchers could improve the generation of alternative topic outlines by incorporating more features to link the topic title clusters, such as investigating module descriptions as an additional basis for the edge weight computation.

Overall, this study proposed a method to use computer science technologies to assist educators in creating effective topic outlines for traditional introductory programming courses. This method could potentially improve the learning experiences of students and can be integrated into an accessible system for educators. The constructed TPG model provided a structured overview of the topics and how they are connected in the domain of introductory programming in the Philippines. In addition, having more data from other HEIs may refine the model in structuring the topics of introductory programming. While this study demonstrates the potential of using data mining and clustering algorithms for topic outline design, further experiments and validation are necessary before implementing them in real classrooms.

Recommendation

The study suggests the use of Latent Dirichlet Allocation (LDA) as a method to improve the limitations of PCK-Means clustering of the topic titles in the context of introductory programming education. By searching for latent topics within the topic titles and classifying certain words that may form a topic, LDA can provide a more thorough investigation of the semantic relationships between words in the topic titles. A better semantic network could be built by conducting a more thorough investigation of the semantic relationships of the words in the introductory programming domain, such as looking at hyponyms and hypernyms of the words. This semantic network could be used as an additional input in the PCK-Means constraint by either (1) adding the topic titles with the words that are on the same topic group in the LDA table in the must-link constraint and those on different topic groups in the cannot-link constraint, or (2) modifying the calculation for the similarity threshold to include a weighted variable dependent on the grouping of words in the LDA topic-word table. These modifications could possibly help reduce the number of anomaly clusters or odd topics per cluster produced by the PCK-Means clustering algorithm.

References

Agrawal, R., Golshan, B., & Papalexakis, E. (2016). Toward data-driven design of educational courses: A feasibility study. *Journal of Educational Data Mining*, *8*(1), 1-21. https://doi.org/10.5281/zenodo.3554601

Almaiah, M. A., & Alyoussef, I. Y. (2019). Analysis of the effect of course design, course content support, course assessment and instructor characteristics on the actual use of e-learning system. *IEEE Access*, *7*, 171907-171922. https://doi.org/10.1109/ACCESS.2019.2956349

Alsaad, F., & Alawini, A. (2020). Unsupervised approach for modeling content structures of MOOCs. In A. N. Rafferty, J. Whitehill, V. Cavalli-Sforza, & C. Romero (Eds.), *Proceedings of the 13th international conference on educational data mining (EDM 2020)* (pp. 18-28). https://files.eric.ed.gov/fulltext/ED608062.pdf

Alsaad, F., Reichel, T., Zeng, Y., & Alawini, A. (2021). Topic transitions in MOOCs: An analysis study. In *Proceedings of the 14th international conference on educational data mining (EDM 2021)* (pp. 139-149). https://files.eric.ed.gov/fulltext/ED615623.pdf

Babaki, B. (2017). COP-Kmeans version 1.5 (1.5) [Python]. https://doi.org/10.5281/zenodo.831850

Bain, A. (2020). Addressing the challenges of program and course design in higher education with design technologies. *Journal of Applied Instruction Design*, *9*(2), 1-9. https://researchoutput.csu.edu.au/ws/portalfiles/portal/47805957/47786144_Published_article.pdf

Basu, S., Banerjee, A., & Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In M. W. Berry, U. Dayal, C. Kamath, & D. Skillicorn (Eds.), *Proceedings of the 2004 SIAM international conference on data mining* (pp. 333-344). https://doi.org/10.1137/1.9781611972740.31

Black, G. S., Daughtrey, C. L., & Lewis, J. S. (2014). The importance of course design on classroom performance of marketing students. *Marketing Education Review, 24*(3), 213-226. https://doi.org/10.2753/MER1052-8008240303

Ekren, G. (2017). Existing criteria determining course quality in distance education. *The Online Journal of Quality in Higher Education, 4*(4), 17-24. https://www.tojqih.net/journals/tojqih/articles/v04i04/v04i04-04.pdf

Krippendorff, K. (1970). Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement, 30*(1), 61-70. https://doi.org/10.1177/001316447003000105

Kyriacou, C. (2019, July 10). *Why is planning so important for effective teaching?* Oxford University Press. https://educationblog.oup.com/secondary/english/why-is-planning-so-important-for-effective-teaching

Licuanan, P. B. (2015, August 3). *Revised policies, standards, and guidelines for Bachelor of Science in Computer Science (BSCS), Bachelor of Science in Information Systems (BSIS), and Bachelor of Science in Information Technology (BSIT) programs*. Commission on Higher Education. https://pdfcoffee.com/15-cmo-25-s2015-revised-psgs-for-bscs-bsis-and-bsit-programs-pdf-pdf-free.html

Manrique, R., Sosa, J., Marino, O., Pereira Nunes, B., & Cardozo, N. (2018). Investigating learning resources precedence relations via concept prerequisite learning. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (pp. 198-205). https://doi.org/10.1109/WI.2018.00-89

McNeil, J. D. (2014). *Contemporary curriculum: In thought and action* (8th ed.). Wiley.

Mtebe, J. S., & Raisamo, R. (2014). A model for assessing Learning Management System success in higher education in Sub-Saharan countries. *The Electronic Journal of Information Systems in Developing Countries, 61*(1), 1-17. https://doi.org/10.1002/j.1681-4835.2014.tb00436.x

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825-2830. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Rüdian, L., & Pinkwart, N. (2021). Finding the optimal topic sequence for online courses using SERPs as a proxy. In *Proceedings of the 14th international conference on educational data mining (EDM 2021)* (pp. 317-322). https://files.eric.ed.gov/fulltext/ED615579.pdf

Shen, S., Lee, H., Li, S. W., Zue, V., & Lee, L. S. (2015). Structuring lectures in massive open online courses (MOOCs) for efficient learning by linking similar sections and predicting prerequisites. In *16th annual conference of the international speech communication association (INTERSPEECH 2015)* (pp. 1363-1367). https://groups.csail.mit.edu/sls/publications/2015/DanielLi_Interspeech-2015.pdf

Vuong, A., Nixon, T., & Towle, B. (2011). A method for finding prerequisites within a curriculum. In *Proceedings of the 4th international conference on educational data mining* (pp. 211-216). http://educationaldatamining.org/EDM2011/wp-content/uploads/proc/edm2011_paper8_short_Vuong.pdf