SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP

A Thesis
Presented to the Faculty of the
Computer Studies Department
College of Science
Technological University of the Philippines
Ayala Boulevard, Manila

by

Elin Amalthea P. Espiritu
Kent Ivan R. Maguad
Khennedy Onnasis M. Maten
Mark Angelo L. Salita
Roswell Nathan I. Chua

In Partial Fulfillment of the
Requirements for the Degree
Bachelor of Science in Computer Science

May 2024

APPROVAL SHEET

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines | Tel No. +632-5301-3001 local 608 Fax No. +632-8521-4063 | Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	TUPM-F-COS-16-TAU
	Revision No.	00
THESIS APPROVAL SHEET FOR THE UNDERGRADUATE PROGRAMS OF THE COS	Date	07012022
	Page	1 / 1

This thesis hereto entitled:

SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP

prepared and submitted by **ELIN AMALTHEA P. ESPIRITU, KHENNEDY ONNASIS M. MATEN, KENT IVAN R. MAGUAD, MARK ANGELO L. SALITA, AND ROSWELL NATHAN I. CHUA** in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE** has been examined and is recommended for approval and acceptance.

PROF. JAN EILBERT L. LEE
Adviser

Approved by the Committee on Oral Examination with a grade of **PASSED** on JUNE 3, 2024.

PROF. MAY M. GARCIA
Member

PROF. PRISCILLA SOTELO-BATOR
Member

PROF. DOLORES L. MONTESINES
Department Head/Chair

Accepted in partial fulfillment of the requirements for the degree **BACHELOR OF SCIENCE IN COMPUTER SCIENCE**.

Date: June 15, 2024

DR. JOSHUA T. SORIANO
Acting Dean

Transaction ID	TUPM-COS-TAU-ELS-07012022 -0258PM
Signature	

Transaction ID Legend: TUPX-AAA (Office Code)-BBB (Type of Transaction)-CCC (Initial of employee)-MMDDDYYYY (month day year)-HHMMAM/PM (hourminutesAM/PM)

ACKNOWLEDGMENT

First and foremost, we want to give a huge thanks to our advisor, Prof. Jan Eilbert Lee, for his incredible guidance, encouragement, and patience during our research. His expertise and insights have been invaluable, helping not just this project but also our growth as researchers. We are really grateful for his consistent support and mentorship, which have been key in shaping the direction and quality of this study.

We also want to give a big thanks to Prof. May M. Garcia and the panelists for their invaluable feedback and suggestions. Their high standards and insightful critiques have really improved the quality of our work. We deeply appreciate their dedication and willingness to offer constructive input, which has helped us refine and elevate our research.

We are incredibly thankful to the Department of Science and Technology - Science Education Institute (DOST-SEI), the Department of Science and Technology Regional Office III (DOST R.O. III), and the Technological University of the Philippines Manila - DOST (TUPM-DOST) for their financial and academic support to our peer, Khennedy Onnasis M. Maten. As a scholar in their respected program, Khennedy received the resources, guidance, and opportunities needed to pursue and complete this research. Their dedication to supporting young researchers and scholars is truly admirable.

I, Elin, want to give my heartfelt thanks to Mr. Jerome A. Domanico for his unwavering support, comfort, and guidance. His dedication and encouragement were

crucial in helping me get to the finish line. I could not have achieved this milestone without his invaluable assistance. Thank you, Mr. Domanico, for everything.

I, Khennedy, wish to express my deepest gratitude to Miss Angela Nicole S. Pinlac for her unwavering support and guidance. Her dedication and belief in me were crucial in helping me overcome challenges and achieve this milestone. Thank you, Miss Pinlac, for everything.

We are also thankful to our colleagues and peers for their help and encouragement throughout this journey. Their collaboration and shared knowledge have been invaluable. Lastly, a big thank you to our families and friends for their unwavering support and understanding during this demanding time. Their encouragement kept us going and helped us reach our goals. We truly appreciate everyone who contributed to the success of this research.

And of course, we want to thank God for His guidance and blessings throughout our research journey. We are deeply grateful for the wisdom, patience, and perseverance He has given us, helping us overcome challenges and achieve our goals. His unwavering presence has been our strength and inspiration. Thank you, God, for your infinite goodness and love.

- The Researchers

ABSTRACT

Creating educational syllabi is a time-consuming and complex task that often reduces from educators' primary responsibilities, leading to inconsistencies and errors due to the manual and repetitive nature of the process. To address this, this study introduces "Syllagenius," an AI-powered syllabus generator that uses Natural Language Processing (NLP) to automate the creation of comprehensive and well-structured syllabi from PDF e-books. With its innovative approach to syllabus creation, it presents a significant advancement in the realm of educational planning. This desktop application streamlines the process of generating syllabi by utilizing user-provided PDF e-books, ensuring that each syllabus is tailored to the specific content and objectives of the course. Adding to its comprehensive features, Syllagenius automates the creation of course objectives, course content, course descriptions, and references, all by APA 7 standards. Throughout the project development, the researchers implemented Data Preprocessing, Data Analysis, Grammar Based Automata, Bidirectional and Auto-Regressive Transformer (BART), N-gram, and Natural Language Processing (NLP). While in testing, Functional Suitability, Performance Efficiency, and Reliability testing were performed and evaluated by purposively selected individuals, including professors, IT professionals, and individuals possessing expertise in both fields. The result showed that the desktop application was suitable with a weighted mean of 3.67, efficient with a weighted mean of 3.80, and reliable with a weighted mean of 3.82, resulting in a grand weighted mean of 3.76, or a “Highly Acceptable” rating from (34) respondents. Overall, this study demonstrated that the AI-powered desktop application "Syllagenius" can significantly streamline syllabus creation for educators. By automating the generation of comprehensive and well-structured syllabi, Syllagenius reduces the time and effort required by educators and enhances the quality of educational planning and delivery.

Keywords: Syllabus, NLP, APA 7, data preprocessing, data analysis, grammar-based automata, BART, N-gram

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

Chapter 1

THE PROBLEM AND ITS SETTING

Background of the study

Winkelmes et al. (2016) highlighted the importance of a well-structured syllabus as a communication tool, emphasizing its role in setting clear course expectations. This key document lays out the overall framework and serves as a comprehensive guide for students, providing essential information on learning objectives, assignments, and assessment criteria. This dual purpose is crucial for effective communication and mutual understanding between educators and learners, ultimately leading to a more organized and transparent learning environment.

Building on this idea, Cox and Orehovec's (2007) study looked at how clear communication in course syllabi affects students. They found that well-detailed syllabi are crucial for setting expectations and boosting student satisfaction. When a syllabus clearly lays out course requirements, goals, and grading criteria, it helps students understand what to expect and engages them more positively with the course.

Similarly, Grunert O'Brien, Millis, and Cohen's (2008) research also showed how important a well-made syllabus is for student engagement and success. They emphasized that clear course guidelines and a well-organized syllabus can significantly boost students' commitment to their studies. This finding highlighted how a thoughtfully prepared syllabus can play a key role in helping students succeed academically.

The challenges in education today highlight the need for new ways to boost efficiency and effectiveness. Creating syllabi, a key part of teaching, is especially tough

because it takes a lot of time and careful planning. Educators have to constantly update syllabi to keep up with changing standards (Freeman et al., 2014). It gets even more complicated when they need to include various materials to meet current standards and address the different needs of students (Alexander et al., 2018).

Designing a syllabus gets even trickier because teachers need to cater to different learning styles and levels in one classroom, as Talbert (2017) pointed out. Finding the right balance between being flexible and maintaining high standards is a big challenge. Plus, today's education often requires blending information from different subjects into one cohesive learning experience, which adds another layer of complexity (Brownell et al., 2016).

Resource limitations within educational institutions exacerbate these challenges, impeding comprehensive syllabus development by hindering access to the latest teaching materials (Mandinach & Gummer, 2016). Quality assurance and standardization across different courses and instructors are essential but formidable tasks (Rice, 2019).

In response to these multifaceted challenges, this paper proposes a forward-looking initiative—an innovative methodology for developing a generic syllabus AI generator. This approach aims to streamline the syllabus creation process by recognizing the potential of state-of-the-art technologies, including E-books, Tokenization, Grammar-Based Automata, BART, NLP, and NLG. By strategically integrating these advanced technologies, the methodology ensures that the generated content aligns with contemporary educational standards, promoting inclusivity and relevancy. This proposal represents a transformative initiative poised to enhance the landscape of educational content creation significantly.

Objectives of the Study

General Objective

The general objective of the study is to implement a desktop application called Syllagenius, utilizing Natural Language Processing (NLP), which can effectively generate a generic syllabus.

Specifically, this study aims to:

1. Design a Natural Language Processing based AI Syllabus Generator with the following features:
    a) Enable users to input PDF e-books.
    b) Can modify the syllabus based on the type of semester and number of exams.
    c) Create a generic syllabus based on the input e-book.
    d) Edit or delete the generated syllabus.
    e) Can customize output syllabus.
    f) Create an APA 7 reference base on the input e-book.
2. Create the desktop application Syllagenius using the following applications:
    a) Python programming
    b) TKinter
    c) Visual Studio
3. Test and improve the system regarding the performance model's functional suitability, performance efficiency, and reliability.

4. Determine the system's acceptability level based on different applicable criteria in ISO 25010, such as functional suitability, performance efficiency, and reliability.

Scope and Limitations of the Study

Syllagenius, with its innovative approach to syllabus creation, presents a significant advancement in the realm of educational planning. This tool streamlines the process of generating syllabi by utilizing user-provided PDF e-books, ensuring that each syllabus is tailored to the specific content and objectives of the course.

Syllagenius offers a bunch of helpful features, like automatically creating course objectives, content, descriptions, and references, all following APA 7 standards. This makes it a super useful tool because it ensures every part of a syllabus is well-crafted and in line with the course's goals. By streamlining these tasks, Syllagenius saves teachers a lot of time and boosts the quality of educational planning and delivery, making it a must-have for top-notch teaching. Plus, it uses Bloom's Taxonomy to make sure the course material hits different cognitive levels, providing a richer and more effective learning experience.

However, it is important to recognize that the reliance on Bloom's Taxonomy within Syllagenius also introduces certain limitations. Bloom's Taxonomy, while widely respected, has its critics and is not universally applicable to all types of learning and teaching methodologies. Some educational contexts might require different frameworks for understanding and categorizing learning objectives and outcomes. Therefore,

Syllagenius's exclusive use of Bloom's Taxonomy could potentially limit its applicability in scenarios where alternative pedagogical theories are preferred or more effective.

Moreover, the restriction to PDF e-books as the sole format for input materials can be seen as another limitation. This constraint may exclude a range of other valuable educational resources that are not available in PDF format, such as interactive online content, videos, or physical books that have not been digitized. This limitation could affect the comprehensiveness and diversity of the syllabus content.

Chapter 2

CONCEPTUAL FRAMEWORK

This chapter presents different related literature and studies used to explore and thoroughly discuss the research topic effectively. Additionally, the conceptual model and the operational definition of words that were utilized in the study are both included in this chapter.

Review of Related Literature

This section discusses key concepts and ideas on the subject matter of the study. It includes discussions on syllabus, e-book, PDF, NLP, Difflib, Unicode, tokenization, lemmatization, text generation and language model, natural language generation, cosine similarity, VS code, and other technological tools.

Syllabus

Wagner, Smith, Johnson et al. (2023) focused on the syllabus's role in guiding interactions between faculty and students in both didactic and experiential educational settings. Their research addressed the dual aspects of encompassing essential and optional details that aid students, faculty, and administration in understanding course expectations and aligning with curricular requirements for program accreditation. They also tackled the

challenges of syllabus bloat and declining student attention to detail. They provided a condensed overview of best practices for designing syllabi that foster transparent and concise communication, especially within pharmacy education.

Englund and Stockhult (2023) then explored the syllabus-boundness associated with a surface learning approach among higher education students. They argued for reevaluating authoritarian influences in education, using a qualitative study to expand the understanding of student engagement with syllabi and incorporate the concept of authority-boundness.

Complementing these perspectives, Harding (2023) examined student-led syllabus design in an Honors American Government course. This approach, involving students in creating the syllabus to reflect diverse learning styles, increases student satisfaction and motivation. Active student involvement in syllabus design can enhance engagement by fostering a sense of ownership and reflecting varied learning preferences.

Merchán Tamayo et al. (2022) contributed to this discourse by studying the impact of language tone in syllabi on student perceptions. Their research, comparing autonomous language supporting student choice against controlling language, underscores the significance of syllabi in shaping student motivation and perception in the learning environment.

Finally, Wotring, Chen, and Fraser (2021) examined curriculum alignment through syllabus document analysis, focusing on the relationship between national language policy and local English Language Teaching (ELT) practices. This study

provided insights into how syllabi align with broader language policies, highlighting the complex interplay between national directives and local ELT implementation.

E-book

Sunghee Shin's (2014) study explored the integration of e-books as primary course materials in educational technology classes for in-service and pre-service teachers. The findings revealed a notable disparity in perceptions between graduate and undergraduate students. While 78.6% of graduate students viewed their e-book reading experience positively, a significant portion of undergraduates (63.6%) expressed interest in borrowing e-book readers from the school library. Moreover, most undergraduates (78.6%) responded favorably to using e-book readers during the course. Despite these positive attitudes towards e-books, over half of the students from both educational levels have preferred print books. However, they remained open to incorporating e-books into their study routines. The study also acknowledged the limitations associated with e-books, such as eye strain and limited e-book collections, highlighting the need for a balanced approach to integrating e-books in academic settings.

Complementing these findings, Rahim et al. (2020) described e-books as dynamic technological tools that present information in a concise format, enhanced with multimedia elements like sound, graphics, images, animation, and movies. These features can make e-books more engaging than their print counterparts. E-books are accessible electronically in various file formats, such as the widely used .pdf, compatible with programs like Acrobat Reader or HTML and Excel formats, which can be opened in web

browsers or specific applications. This multifaceted nature of e-books allows for a versatile reading experience, catering to students' and educators' diverse needs and preferences in educational technology.

PDF

Yamauchi et al. (2023) highlighted that in recent years, intelligent learning environments have become integral to modern education, aiding students and instructors with tools rooted in prediction and recommendation models. These methods often utilize metadata from learning materials, such as the knowledge encapsulated in exercises, which typically requires labeling by domain experts. This process is both costly and challenging to scale. The authors noted that automated labeling can significantly reduce the workload on experts, as demonstrated in prior studies involving the automatic classification of research papers and Japanese mathematical exercises. However, they pointed out that these studies have yet to explore fine-grained labeling extensively. Furthermore, the authors observed that as the adoption of these materials increases in the system, the conversion of paper materials into PDF formats is standard, which may result in incomplete data extraction.

Natural Language Processing (NLP)

Abdul Ahad et al. (2023) described natural language processing (NLP) as a computerized approach to analyzing text, encompassing structured and unstructured data. They characterized NLP as a straightforward, empirically potent, and reliable method.

This approach has achieved state-of-the-art performance in various language processing tasks, including Machine Translation (MT), Text Summarization (TS), and Emotion Detection (ED). According to their analysis, given the current technology deployment and adoption trends, NLP is poised to be a pivotal technology.

Difflib

Difflib is a Python module that provides tools for comparing sequences, which can be applied to text and code. The SequenceMatcher function in difflib is particularly useful for identifying similarities and differences between sequences. Hetland (2005) demonstrated the effectiveness of difflib in text comparison tasks, making it a suitable tool for code comparison as well.

Unicode

Unidecode are essential for preprocessing tasks that prepare the text for comparison. Together, these tools form a comprehensive text processing pipeline that can handle diverse input formats and ensure accurate and reliable sequence matching. Studies have shown that combining these preprocessing techniques with advanced sequence matching algorithms enhances the effectiveness of text analysis in various applications, from software development to natural language processing (Baishakhi & Kim, 2011; Kapser & Godfrey, 2006).

Natural Language Processing (NLP) in Education

Building on this foundation, there is a clear and emerging focus on the role of Natural Language Processing (NLP) within the educational landscape. The increasing availability of educational text and speech data, coupled with advancements in educational technology, has spurred significant interest in utilizing NLP to meet the unique needs of educators and learners. It is essential to recognize that the educational applications of NLP possess distinct characteristics, differing from the traditional domains for which NLP systems were initially designed, as noted by Litman in 2016.

In their 2023 study, Kurni, Mohammed, and Srinivasa delved into the application of NLP in education. They explored how NLP's ability to understand and generate human language can be leveraged to create educational tools such as automated grading systems, personalized feedback mechanisms, and interactive learning assistants. This study underscored the potential of NLP to revolutionize the educational sphere by offering more engaging and tailored learning experiences.

Tokenization

The comprehensive survey conducted by Mielke et al. (2021) delved into the evolving landscape of open-vocabulary modeling and tokenization in natural language processing (NLP). The focal question guiding this exploration is determining the optimal units for text modeling, ranging from bytes to multi-word expressions. Historically, NLP models predominantly operated at the word level, treating words as discrete and atomic tokens. However, the advent of byte-pair encoding (BPE) marked a significant shift

towards subword-based approaches, which have become prevalent in various domains. This transition allows for compact vocabularies while ensuring rapid inference. The survey navigated through pre-neural and neural eras, highlighting hybrid strategies combining words and characters alongside subword-based methodologies grounded in learned segmentation. The inquiry concerned whether character-level models or byte-level processing represent the ultimate frontier. The survey concluded by emphasizing the absence of a universal solution applicable to all applications, underscoring the perpetual significance of thoughtful consideration regarding tokenization in diverse NLP contexts.

Yogish, Manjunath, and Hegadi (2019) presented a comprehensive review of the evolving landscape of Natural Language Processing (NLP) trends and techniques, with a particular focus on the utilization of the Natural Language Toolkit (NLTK). In the contemporary era of information explosion, where vast amounts of data in various forms are generated daily, effective NLP techniques play a pivotal role in information retrieval systems, text summarization, sentiment analysis, information and relationship extraction, social media monitoring, text mining, language translation, and question-answering systems. NLP involves intricate linguistic analysis to represent natural language for further computational processing. The authors highlight the challenges inherent in NLP, emphasizing that understanding human language relies not only on individual words but also on how these words are interconnected to convey precise meanings. Despite the language being an easily comprehensible concept for humans, teaching computers to comprehend natural language poses difficulties due to the inherent ambiguity in language syntax and semantics. NLP techniques, encompassing processes like tokenization, stop-word removal, stemming, lemmatization, parts of speech labeling, chunking, and named

entity recognition, contribute to the efficient retrieval of information and streamlined documentation processes. The authors underscored NLTK as a valuable resource, offering application packages that aid researchers and learners in natural language processing, computational linguistics, and artificial intelligence.

Lemmatization

Plisson, Lavrac, and Mladenic (2004) contributed to the field of lemmatization—a process crucial in obtaining the normalized form of a word. Their paper investigates explicit word endings, exploring which suffix should be removed or added to achieve a word's normalized state. The study compared the outcomes of two-word lemmatization algorithms grounded in if-then rules and ripple-down rules induction algorithms. Focused on the lemmatization challenges of Slovene free text, the paper delineates why the Ripple Down Rules (RDR) approach is highly suitable for this linguistic task. Notably, the RDR approach, when trained on a corpus of lemmatized Slovene words, yields rules characterized by improved classification accuracy, surpassing the achievements of rule learning in prior research efforts. This contribution advances our understanding of effective lemmatization strategies, emphasizing the significance of rule-based approaches and their applicability to specific linguistic contexts.

Khyani et al. (2021) contributed to the foundational understanding of Natural Language Processing (NLP), specifically focusing on lemmatization and stemming. In their research, the authors aimed to establish a comprehensive overview of NLP technology, tracing its evolution applications and assessing its merits and demerits. The

paper delved into critical concepts such as Lemmatization and Stemming, elucidating their operational mechanisms, algorithms, and practical applications. By providing insights into these linguistic processes, the research equips readers with a thorough understanding of their functionalities. Notably, the study concluded with a comparative analysis of lemmatization and stemming, seeking to determine the superior approach. This work is valuable for those seeking a foundational grasp of NLP and its integral components, contributing to the broader discourse on language processing techniques.

Text Generation and Language Model

Junyi Li et al. (2022) highlighted the significance of text generation, also known as natural language generation, within the broader field of natural language processing (NLP). They emphasize its core aim of producing coherent and understandable text in human language from various types of input data such as text, images, tables, and knowledge bases. This sub-field has seen widespread application across numerous scenarios over recent decades.

In their 2022 publication in IEEE Access, Fatima et al. offered an in-depth analysis of deep neural network models for text generation. Their review encompassed a range of neural network architectures, evaluating their efficacy in generating coherent and contextually accurate text. They discussed challenges in model training and the impact of different network structures on the quality of the generated text. This comprehensive examination is a vital academic resource, providing insights into neural

network-based text generation's technological advancements, capabilities, and limitations.

Similarly, Morrison's 2022 article offered an extensive overview of large language models and text generators. This work is particularly insightful for educators, as it delved into the complexities of these advanced language technologies, exploring their applications, strengths, and potential impact in educational settings. Morrison’s article illuminated how these technologies can be utilized as tools for learning, communication, and content creation within the educational sphere.

Natural Language Generation

The 2018 study by Gatt and Krahmer, titled "Survey of the State of the Art in Natural Language Generation: Core tasks, applications, and Evaluation," offered a comprehensive overview of the critical tasks, applications, and evaluation methods in the field of natural language generation (NLG). This work delved deeply into the core aspects of NLG, providing a valuable resource for understanding its various dimensions.

Expanding upon these insights, Bisen and Agrawal's 2022 paper, published in the International Journal of Health Sciences, presented a detailed review of NLG, focusing on its development, applications, and technological foundations. Their review explored the evolution of NLG within computational linguistics and its ability to generate coherent and contextually relevant text. They covered a broad spectrum of NLG applications, highlighting its importance in healthcare, where clarity and precision in communication are crucial. The paper also discussed the various techniques and models developed in

NLG, the advancements in the field, and the challenges researchers and practitioners face.

Cosine Similarity

The study by Rahutomo, Kitasuka, and Aritsugi (2012) addressed the prevalent use of cosine similarity as a metric in information retrieval and related studies. Cosine similarity, while widely implemented, faces limitations in accurately capturing the semantic meaning of text. In response to this challenge, the paper proposed a notable enhancement to cosine similarity measurement by introducing semantic checking between dimensions of two-term vectors. The objective is to amplify the similarity value between term vectors that exhibit semantic relations among their dimensions with varying syntax. The experimental results presented in the study demonstrated the effectiveness of this proposed approach, highlighting promising outcomes. This research contributed valuable insights to the field, offering a solution-oriented perspective to improve the performance of cosine similarity by incorporating semantic considerations, thereby advancing the nuanced understanding of text similarity measurement.

Lahitani, Permanasari, and Setiawan's (2016) investigation centered on using cosine similarity in the realm of Online Essay Assessment (OEA) within the educational sphere. In response to integrating technology into education, the study underscored the importance of streamlining learning processes, file sharing, and assignment and assessment procedures through automated systems. Automated Essay Scoring (AES) stands out as a crucial advancement, seeking to autonomously ascertain scores from text

documents, simplifying correction and scoring processes for educators through computer-based applications. The adoption of AES enhances efficiency and addresses subjectivity concerns in scoring. The study delves into the obstacles encountered in AES implementation, especially in essay scoring, where language subtleties and scoring mechanisms play pivotal roles.

Visual Studio Code (VS Code)

According to Plainer (2021), Visual Studio Code, a versatile and free source code editor, offers developers a range of features. Its adaptability is a critical advantage, enabling support for multiple programming languages by integrating plugins. Project management in the editor is facilitated by folders and workspaces, allowing for customization and language independence. Three protocols are in place to enable decoupling between the editor and extensions, ensuring consistency and ease of maintenance. Visual Studio Code draws on a mix of internal dependencies, including open-source and proprietary elements, and its functionality can be expanded through external plugins. Developed in TypeScript, a statically typed superset of JavaScript, Microsoft actively maintains the editor. The editor's origins can be traced back to the Monaco web editor project.

Python

Python's versatility and power as an object-oriented programming language have propelled it to popularity in recent years. One of the main drivers behind its widespread adoption is its dynamic semantics, which empowers developers to create flexible and adaptable code. A notable strength of Python is its user-friendliness. The language has been meticulously designed to learn and read thanks to its concise syntax. This focus on readability not only makes Python accessible to beginners but also brings long-term benefits by reducing software maintenance costs. Clear and understandable code is easier to maintain and update, saving developers valuable time and resources. (What is Python? Executive Summary, n.d.-b).

NLTK

The Natural Language Toolkit (NLTK), a Python library developed by Steven Bird, Edward Lopez, and Ewan Klein for educational and developmental purposes (Rouse, 2023), stands out as an excellent resource. It features a user-friendly guide designed to teach computational linguistics and Python programming basics, making it accessible to a broad audience, including linguists, engineers, researchers, students, and educators interested in delving into computational linguistics. NLTK's distinctive characteristic lies in its vast collection of over 50 datasets and word sources, such as the Penn Treebank Corpus, Open Multilingual Wordnet, Problem Report Corpus, and Lin's Dependency Thesaurus. Users of NLTK can thoroughly analyze language and explore various research topics by utilizing these datasets and word sources.

N-gram

The study conducted by Ojo et al. (2021) explored the use of n-grams (unigrams, bigrams, and trigrams) as features for sentiment classification using various machine learning and deep learning methods. This research aimed to address the challenges and advantages of employing n-grams in the analysis of economic texts, filling a gap in existing literature. The study evaluated the performance of these n-gram features across different datasets in the economic domain, employing nine different sentiment analysis techniques. By comparing the precision, recall, f1-score, and accuracy of these methods, the researchers extracted valuable insights into the effectiveness of n-grams for sentiment analysis. The datasets used for testing include economic reviews from Amazon, IMDB, Reuters, and Yelp. The comprehensive experiments demonstrated the efficacy of n-grams in accurately analyzing sentiments in economic texts.

Bidirectional and Auto-Regressive Transformer (BART)

The study by Lewis et al. (2019) introduced BART, a denoising autoencoder designed for pretraining sequence-to-sequence models. BART operates by corrupting text with various noising functions and then training a model to reconstruct the original text. It utilizes a standard Transformer-based neural machine translation architecture, integrating aspects of BERT's bidirectional encoder and GPT's left-to-right decoder, among other pretraining schemes. The researchers evaluated several noising techniques, discovering optimal performance through random sentence shuffling and a novel in-filling scheme, where text spans are replaced with a single mask token. BART excels in

text generation and comprehension tasks, matching RoBERTa's performance on GLUE and SQuAD with similar training resources. It sets new benchmarks in abstractive dialogue, question answering, and summarization tasks, achieving up to 6 ROUGE points improvement. Additionally, BART enhances machine translation, providing a 1.1 BLEU increase over a back-translation system with only target language pretraining. The study also included ablation experiments to assess the impact of different pretraining schemes within the BART framework on task performance.

Related Studies

The following studies were found relevant to the present study:

The realm of educational technology has witnessed a surge in interest and research surrounding the utilization of automatic syllabus generators, particularly those integrating sophisticated algorithms such as Natural Language Processing (NLP). Among these endeavors, a notable study conducted by Sodhi and Choudhary in 2023 stands out for its ambitious goal of enhancing syllabus creation efficiency through the deployment of cutting-edge text analysis tools. The study, titled “Text Analyzing Tool for Simplifying the Syllabus Creation Process” embarked on a journey to harness the power of NLP algorithms to revolutionize the process of syllabus creation.

With a keen focus on developing a tool that could comprehensively analyze educational texts, Sodhi and Choudhary aimed to alleviate the myriad challenges commonly encountered by educators during syllabus crafting. These challenges, ranging from ensuring content relevance and coherence to aligning syllabi with learning

objectives, often present significant hurdles in the educational landscape. By leveraging advanced text analysis techniques, the researchers endeavored to provide educators with valuable insights into their subject matter, thus simplifying the arduous process of syllabus creation.

The benefits offered by employing text-analyzing tools in syllabus generation are manifold. Not only do they enhance the organization and clarity of syllabi, but they also drastically reduce the time and effort required for manual analysis. Furthermore, these tools promote consistency across courses, ensuring coherence in educational materials disseminated to students. Sodhi and Choudhary's tool, in particular, aimed to foster collaboration among educators by providing a platform for sharing and refining syllabus content, thus facilitating a culture of collective improvement within educational institutions.

One of the standout features of Sodhi and Choudhary's research lies in their emphasis on creating a universally applicable tool. By leveraging the flexibility and adaptability of NLP algorithms, they ensured that their platform could be customized to meet the diverse needs and preferences of educators. Moreover, the researchers placed a strong emphasis on accessibility and convenience, acknowledging the varying technical backgrounds of educators. This commitment to inclusivity ensured that educators from all walks of life could effectively utilize the tool to enhance their syllabus creation process.

Synthesis of Reviewed Studies

In response to the formidable challenges educators face in manually crafting syllabi, Sodhi and Choudhary (2023) embarked on a pioneering study titled “Text Analyzing Tool for Simplifying the Syllabus Creation Process” Their research delved into the realm of Natural Language Processing (NLP), aiming to revolutionize syllabus creation by leveraging advanced text analysis technology. With a focus on developing a user-friendly platform, the researchers sought to extract core topics and essential concepts from a myriad of educational texts, including academic literature and textbooks. This ambitious endeavor was driven by the imperative to address prevalent issues such as content relevance, coherence, and alignment with learning objectives, which often plague the traditional syllabus creation process.

The tool conceptualized by Sodhi and Choudhary represents a significant leap forward in the realm of educational technology. By harnessing the power of NLP algorithms, the platform not only simplifies syllabus creation but also enhances its organization and clarity. Educators stand to benefit immensely from the insights provided by this tool, gaining a deeper understanding of their subject area’s structure and content. Moreover, by facilitating collaboration among educators through a shared platform, the tool fosters a culture of collective improvement and knowledge exchange, thereby enriching the overall educational landscape.

In contrast, Syllagenius offers a distinct yet equally innovative approach to syllabus generation. As a desktop application, it harnesses the capabilities of NLP and Python programming to automatically generate syllabi from e-books. However, what sets

Syllagenius apart is its comprehensive nature. Not content with merely extracting content, it meticulously crafts course objectives, content outlines, descriptions, and references—all meticulously formatted according to the stringent guidelines of APA 7 standards. This holistic approach ensures not only efficiency but also adherence to academic rigor and citation standards, thereby elevating the quality and consistency of syllabi generated.

Moreover, the user interface of Syllagenius, built using Tkinter, offers educators a seamless and intuitive experience, regardless of their technical proficiency. By prioritizing accessibility and user-friendliness, Syllagenius democratizes syllabus creation, empowering educators of varying backgrounds to harness the power of technology for educational enhancement.

While Sodhi and Choudhary's tool and Syllagenius both aim to simplify syllabus creation with technology but focus on different aspects. Sodhi and Choudhary's platform emphasizes content analysis and educator collaboration, while Syllagenius provides a comprehensive solution for content extraction, organization, and formatting aligned with academic standards. Together, they enhance educational efficiency and effectiveness through their unique strengths.

Conceptual Model of the Study

As illustrated in Figure 1, a conceptual model was created based on the ideas, theories, and research or in the relevant literature, studies, and insights.

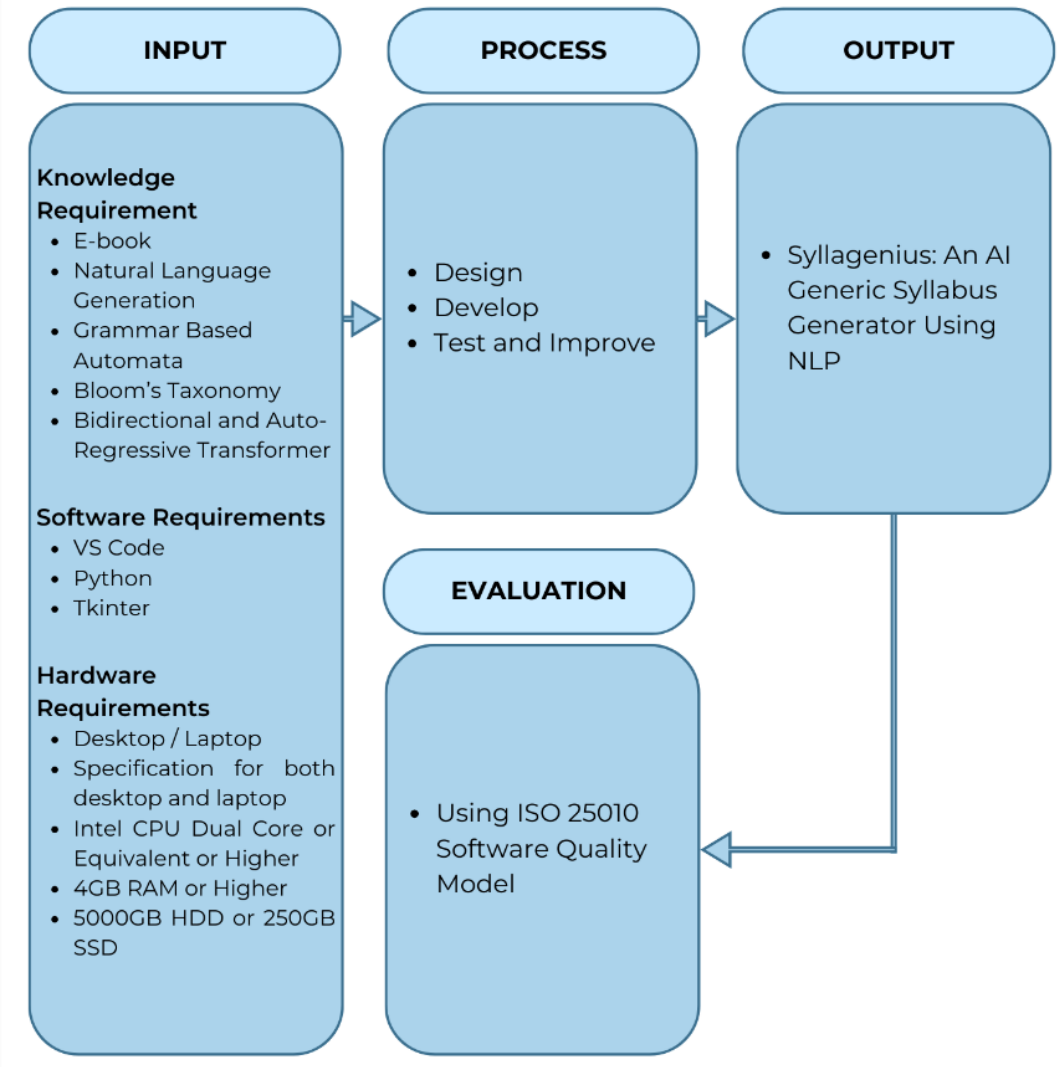


Figure 1. Conceptual Model of the Study

Input

The knowledge, software, and hardware required for the development of desktop applications are contained in the input block. The knowledge requirements are the concepts and facts that the researchers must understand and study to create a foundation for the study. These requirements include understanding E-books, Natural Language Generation, Grammar Based Automata, Bloom's Taxonomy, Bidirectional and Auto-Regressive Transformer, Visual Code Studio, Python, and Tkinter. The software required for developing a Natural Language Processing generic syllabus generator includes application programs, programming languages, frameworks, and operating systems. The hardware requirements are the tools for creating for creating desktop application, such as personal computer or laptops.

Process

The process block includes designing, developing, and testing activities to develop the desktop application.

Design. In this phase, the researchers created the block diagram for the desktop application, which serves as the desktop application model. The researchers also prepared a use case diagram specifying the desktop application's expected functionalities where external users can interact by detailing each element and the interaction flow between the user and desktop application. The use case diagram is a model for efficiently communicating the system's behavior in the user's terms.

Develop. In the development phase, the construction of the desktop application took place. The desktop application will be built using Python and Tkinter in Visual Studio Code IDE.

Test. After developing, the developed desktop application should undergo tests to identify the overall functionality and accuracy in generating a syllabus.

Output

Connecting with the inputs and processes mentioned earlier, the result of the system is a "SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP." This Syllagenius undergoes evaluation, involving respondents selected through a purposive sampling technique to assess its functions and performance.

Evaluation

The evaluation block contains the system needed to satisfy the different needs of stakeholders. The ISO 25010 assessment tool examined the system's functional suitability, performance efficiency, and reliability.

Chapter 3

METHODOLOGY

The techniques for project design, development, operation, and testing, as well as the evaluation technique, are described in this chapter. This section of the study explains how the application was created and developed, as well as how the researchers installed and tested it.

Project Design

This research project focused on creating and developing a system called Syllagenius that leverages the Natural Language Processing (NLP) technique to generate a generic syllabus based on the inputted e-book. The application was built using various cutting-edge technologies, including Visual Studio Code, NumPy, PyMuPDF, Unicode, Difflib, NLTK, Regular Expression Library, Anydesk, Tkinter, and Python as the primary programming language.

To achieve this, a series of operations were carried out to implement the system. An overview of the process utilized in the system is illustrated in Figure 2. This process flow diagram clearly and concisely represents the steps to execute the system. By following this flow, the development team can systematically implement the system in an organized and efficient manner.

System Design

The process flow of the Syllagenius is represented through a flowchart, illustrated in Figure 2. The diagram reveals how the input, process, conditions, and output are sequenced accordingly to furtherly comprehend how the system works.

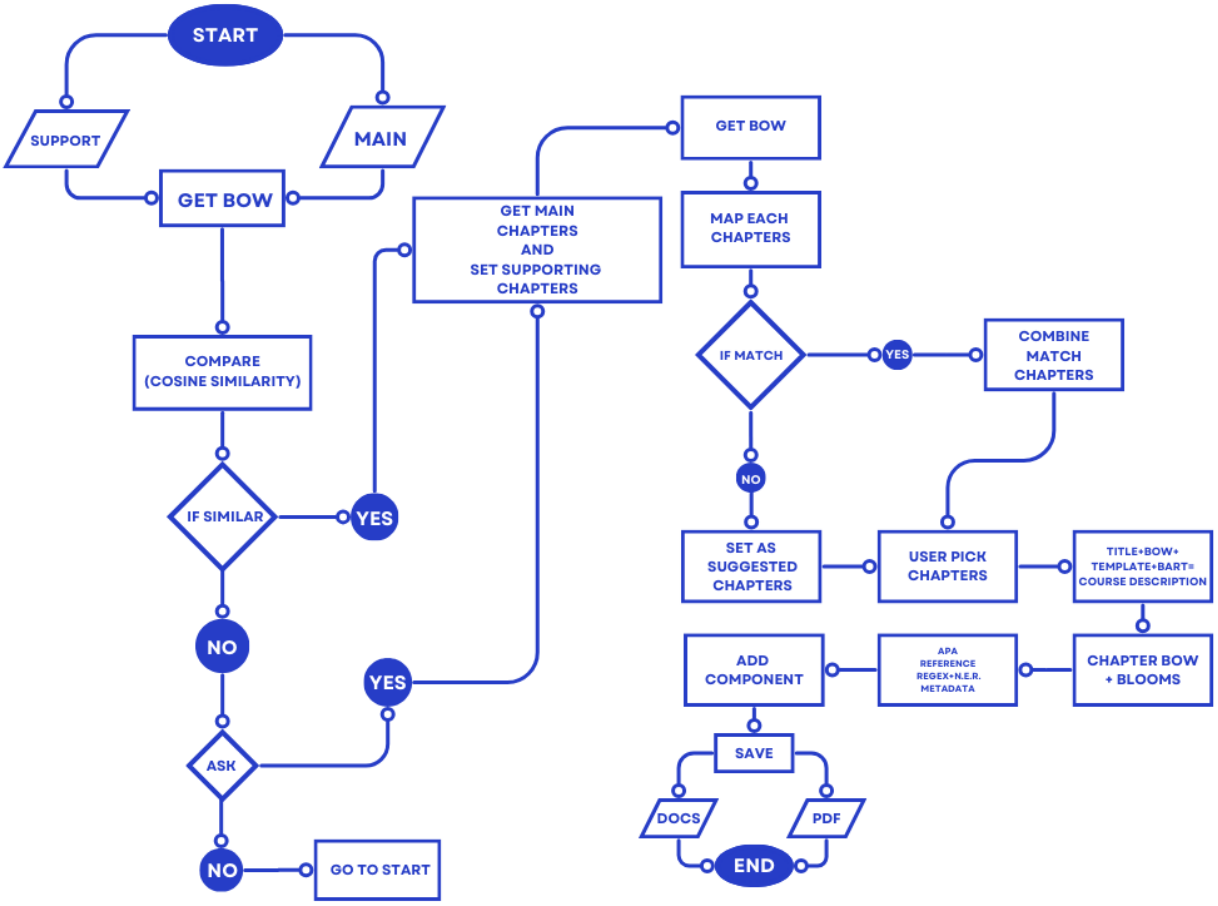


Figure 2. System Flowchart

Algorithm Design

This design outlines a systematic approach for comparing the content of two ebooks using text extraction, preprocessing, and cosine similarity analysis. It includes steps for handling cases where no supporting ebook is provided and incorporates a user decision point to control the process. This method ensures accurate assessment and comparison of ebook content, enabling further analysis or actions based on similarity results.

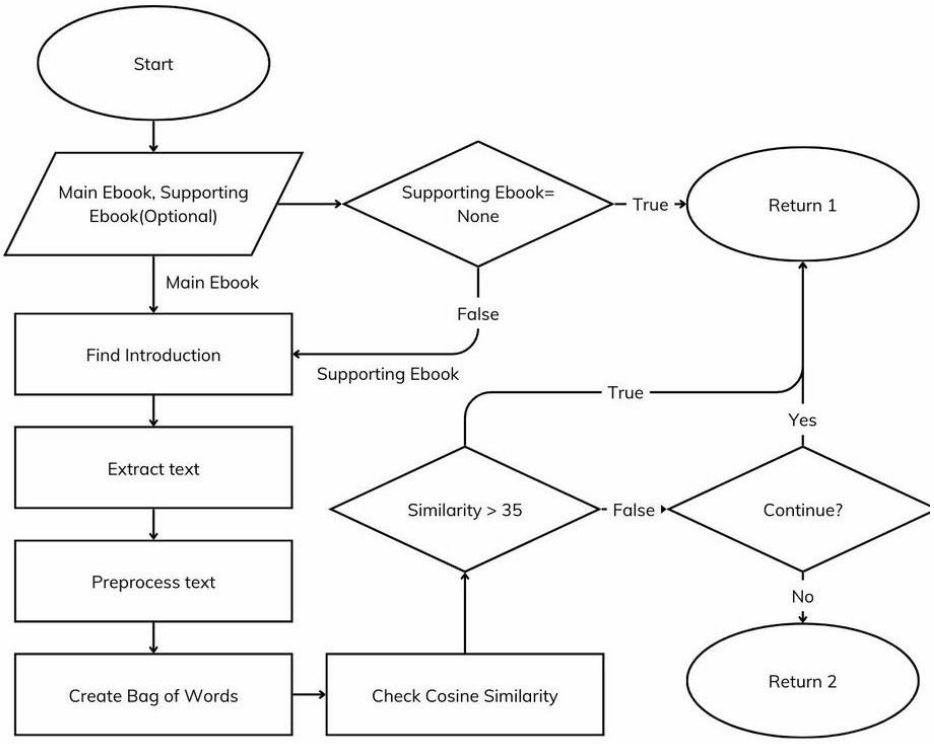


Figure 3. Algorithm Flowchart for Text Similarity Analysis

Project Development

This section discusses the procedures followed on how the desktop application was developed based on the design specifications.

Data Preprocessing

Data preprocessing is pivotal in data analysis and natural language processing, involving several indispensable tasks. Initially, tokenization is employed, wherein text data undergoes segmentation into smaller units, such as words or phrases, rendering it more amenable for analysis. This process facilitates the dissection of the text, preparing it for subsequent analytical procedures. Following this, the removal of stop words becomes imperative. These ubiquitous terms, such as “the,” “is,” or “in,”, often carry scant meaning in the context of data analysis. Their exclusion is analogous to refining a text by removing non-essential elements, focusing exclusively on the most substantive components, and enhancing the overall quality of the data for downstream processing.

Subsequently, cleaning is another critical step involving meticulously removing extraneous or disruptive elements, such as symbols or irrelevant characters. This process aims to refine the dataset, ensuring it is free from unwanted factors contributing little to the overall quality. Lastly, lemmatization is applied, entailing the reduction of words to their base or root form. For instance, “running,” “runs,” and “ran” converge to “run.” This procedural consistency aids in precisely capturing the essence of the data, contributing to its uniformity and cleanliness. This step is comparable to refining raw material to its essential form, preparing it for subsequent processing or analysis.

Collectively, these data preprocessing steps impart uniformity, cleanliness, and readiness to the data for the ensuing analytical endeavors.

This pseudocode provides a preprocessing function for text data, crucial for preparing raw text for analysis or model input. The function ‘preprocess_text’ normalizes text to ASCII, removes non-alphanumeric characters, converts text to lowercase, tokenizes the text, removes stop words, and lemmatizes the tokens. It returns the cleaned and processed text as a single string, ensuring the data is consistent and manageable for subsequent analysis or machine learning tasks.

```
FUNCTION preprocess_text(text):
    text = unidecode(text)  // Normalize text to ASCII
    text = remove_non_alphanumeric(text)  // Remove non-alphanumeric characters
    text = convert_to_lowercase(text)  // Convert text to lowercase

    // Remove stop words
    tokens = tokenize_text(text)  // Tokenize the text
    stop_words = get_english_stopwords()  // Get English stop words
    tokens = FILTER tokens WHERE word NOT IN stop_words  // Remove stop words

    // Lemmatize tokens
    lemmatizer = initialize_lemmatizer()
    tokens = MAP tokens TO lemmatizer.lemmatize(word)
    RETURN join_tokens(tokens)
END FUNCTION
```

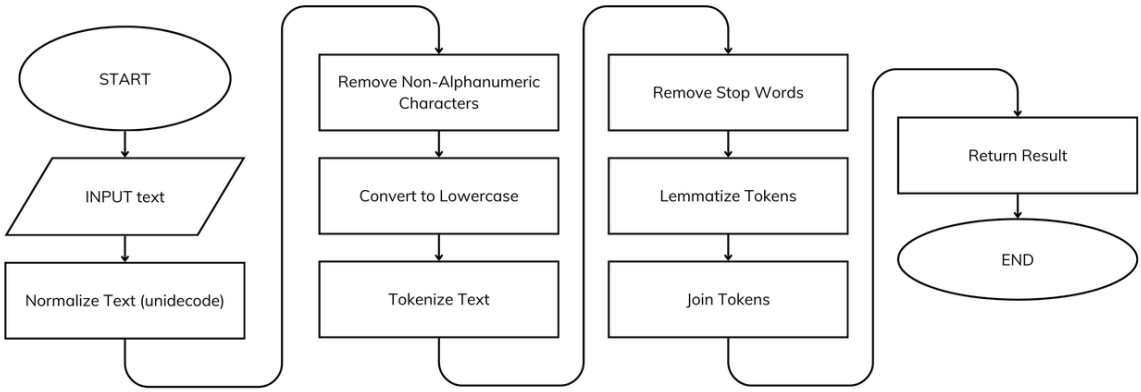


Figure 4. Flowchart for Data Preprocessing

Data Analysis

Incorporating data analysis brings significant benefits that enhance both its functionality and effectiveness. By leveraging data analysis, such a system can deeply understand diverse learning needs and preferences. It can analyze historical data, student feedback, and performance metrics to tailor the syllabus to meet specific educational goals and learning styles, making the learning experience more effective and engaging.

Additionally, data analysis aids in continuously improving the syllabus content. By evaluating the effectiveness of different modules and topics, the system can identify areas that need refinement, ensuring that the syllabus remains relevant and up to date in the fast-evolving educational landscape.

Moreover, data analysis uncovers insights into learning patterns and potential difficulties students might face with specific topics. This allows for proactive adjustments to the syllabus, incorporating additional resources or support where necessary and

empowering educators to make data-driven decisions that enhance the overall quality of education.

Furthermore, using NLP, the system can analyze vast amounts of text data, extracting and synthesizing information from academic research, textbooks, and curriculum standards. This capability ensures that the syllabus is comprehensive, aligning with educational best practices and current knowledge in the field.

This pseudocode creates a Bag-of-Words (BoW) representation from a list of texts. It ensures the input is a list of strings and contains meaningful content. It initializes a TF-IDF vectorizer with uni-grams and bi-grams, transforms the texts into a TF-IDF matrix, and extracts feature names. The TF-IDF scores are summed and sorted, with bi-grams given a boost. It then selects and prints the top features, returning the vectorizer, TF-IDF matrix, and top features.

```
FUNCTION create_bow_from_text(texts):
    // Ensure texts is a list of strings
    IF texts IS STRING:
        texts = [texts]
    END IF

    // Check if texts are not empty
    IF texts IS EMPTY OR all(text IS EMPTY OR text CONTAINS ONLY STOP WORDS FOR EACH text IN texts):
        RAISE ValueError("The provided texts are empty or contain only stop words.")
    END IF

    // Initialize TfidfVectorizer to include uni-grams and bi-grams
```

```
    vectorizer = initialize_tfidf_vectorizer(ngram_range=(1, 2),
stop_words='english')

    // Transform texts into TF-IDF matrix
    X = vectorizer.fit_transform(texts)

    // Extract feature names
    feature_names = vectorizer.get_feature_names_out()

    // Sum TF-IDF scores for each feature
    summed_tfidf = sum_tfidf_scores(X)

    // Sort features by summed TF-IDF scores in descending order
    sorted_indices = sort_indices_descending(summed_tfidf)

    // Boost bi-grams in the sorting
    bi_gram_boosted_indices = sort_by_bi_grams_and_scores(sorted_indices,
feature_names, summed_tfidf)

    // Get top features with highest summed TF-IDF scores
    top_features = get_top_features(bi_gram_boosted_indices, feature_names, 7)

    // Print top features for verification
    PRINT "Top Features (uni-grams or bi-grams) based on summed TF-IDF
scores:", top_features

    RETURN vectorizer, X, top_features
END FUNCTION
```

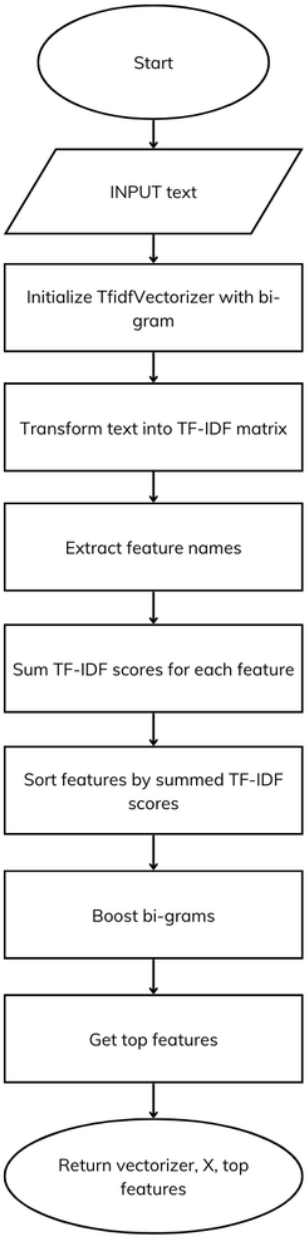


Figure 5. Flowchart for Data Analysis

Grammar Based Automata

Integrating grammar-based automata and aligning it with Bloom's Taxonomy creates an advanced tool for crafting educational content. Bloom's Taxonomy is a hierarchical model that classifies educational goals, promoting the development of higher order thinking skills, ranging from basic knowledge recall to more complex tasks like analysis, synthesis, and evaluation. In this setup, grammar-based automata can be programmed to ensure that the syllabus content and structure align with the various levels of Bloom's Taxonomy, as shown in Figure 4.

This integration enables the automata to formulate learning objectives at different cognitive levels, ensuring a comprehensive educational approach. For instance, the syllabus might focus on lower-level knowledge and comprehension, while higher levels could target analysis and evaluation skills. The automata's grammar-based capabilities ensure that these objectives are expressed clearly and correctly, adhering to the linguistic nuances of educational content.

Furthermore, incorporating Bloom's Taxonomy into the syllabus generation process encourages a more holistic and balanced educational experience. It ensures that students are not only memorizing facts but are also engaging in critical thinking and problem-solving.

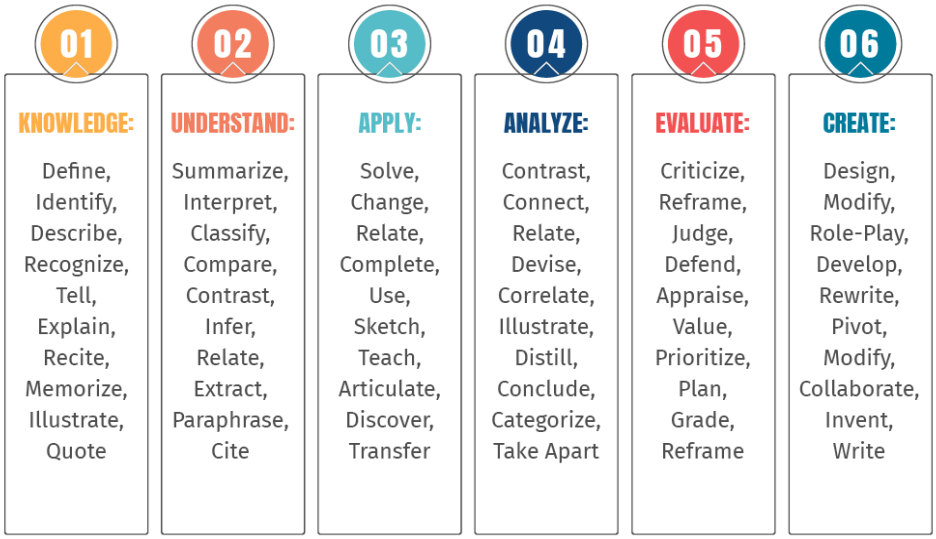


Figure 6. Bloom's Taxonomy

This pseudocode defines a function to generate course outcomes from chapter texts using a vectorizer. The function ‘generate_course_outcomes' creates an empty list for outcomes, transforms chapter text into a Bag of Words (BoW) representation, and selects top keywords. It generates 2 or 3 random outcomes using these keywords with predefined templates and categories and paraphrases them for variety and coherence. The function returns the outcomes as a list, aiming to create meaningful and relevant learning outcomes for each chapter by extracting and utilizing key terms effectively.

FUNCTION generate_course_outcomes(chapter_texts, vectorizer):
INITIALIZE outcomes AS EMPTY LIST

FOR EACH title, text IN chapter_texts:
// Get BoW representation for the cleaned text

```
        bow = vectorizer.transform([text])
        feature_array = vectorizer.get_feature_names_out()

        // Sort terms by frequency and select top keywords
        tfidf_sorting = SORT bow.toarray().flatten() IN DESCENDING ORDER
        top_keywords = SELECT feature_array[i] FOR i IN tfidf_sorting[:5]

        // Generate 2 or 3 random outcomes using these terms
        ri = RANDOM INTEGER BETWEEN 2 AND 3
        FOR _ IN RANGE(ri):
            template = RANDOM CHOICE FROM learning_outcomes_templates
            category_name = RANDOM CHOICE FROM list(categories.keys())
            action = RANDOM CHOICE FROM categories[category_name]
            keyword = RANDOM CHOICE FROM top_keywords
            outcome = template.format(action, keyword)
            paraphrased_outcome = paraphraser(outcome, max_new_tokens=50)
            outcomes.append((title, paraphrased_outcome[0]['generated_text']))
        END FOR
    END FOR
    RETURN outcomes
END FUNCTION
```

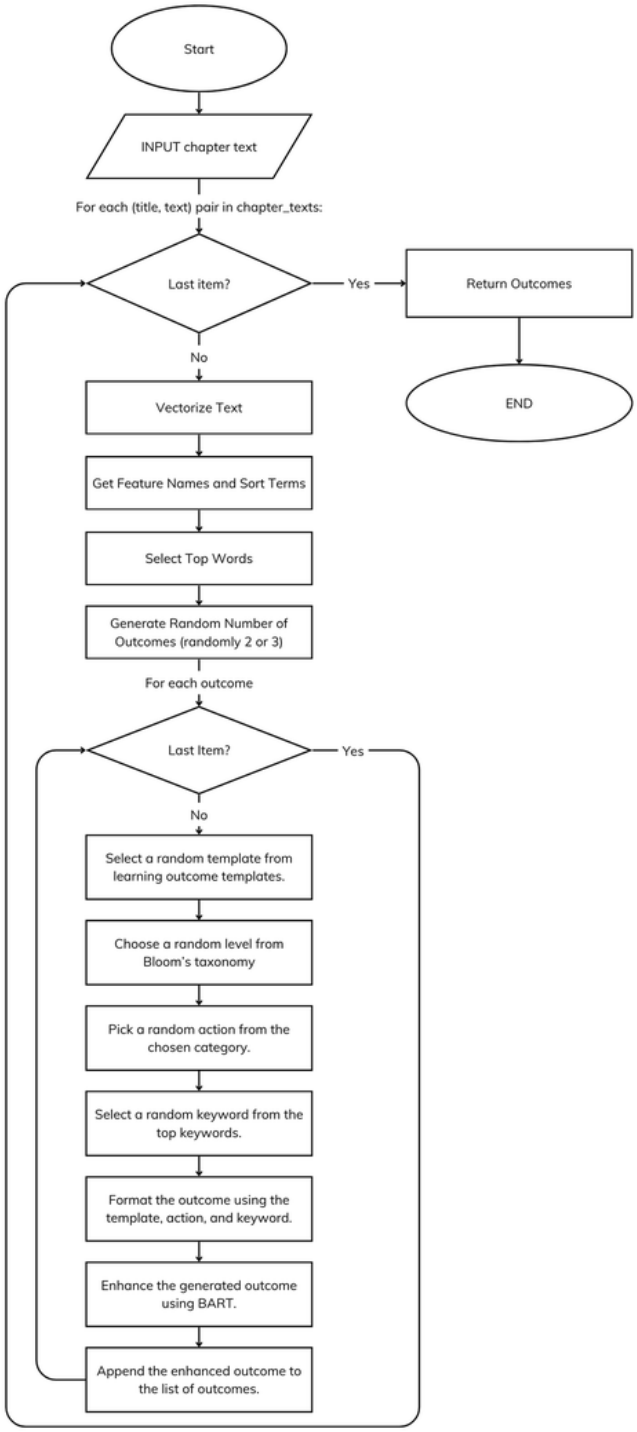


Figure 7. Flowchart for Grammar Base Automata Course Outcomes

Bidirectional and Auto-Regressive Transformer (BART)

Using Bidirectional and Auto-Regressive Transformers (BART) for grammar correction in the context of syllabus creation significantly enhances the linguistic accuracy and professionalism of educational materials. BART's architecture, which combines a bidirectional encoder and an autoregressive decoder, makes it exceptionally capable of handling complex grammatical structures in academic syllabi.

When applied to syllabus creation, BART meticulously reviews and refines the language used, ensuring that the syllabus is not only grammatically correct but also clear and understandable. This precision is crucial in educational settings, where the clarity of instructions and descriptions can significantly impact learning outcomes. BART's proficiency in detecting and correcting nuanced grammatical issues, such as subject-verb agreement, tense consistency, and the correct usage of technical terminology, is particularly valuable.

Moreover, BART's training on extensive and diverse text datasets allows it to adapt to the specific language style of academic writing. This adaptability ensures that the syllabus maintains a professional tone appropriate for educational environments. By integrating BART into the syllabus generation process, educational institutions and educators can ensure their syllabi are pedagogically sound and linguistically polished, thereby enhancing the overall quality and effectiveness of their educational materials.

This pseudocode contains a function to generate course descriptions using templates and a dictionary of words. The function 'generate_description' initializes an empty list for sentences, fills each template with randomly selected words from the

dictionary (including adjectives, nouns, verbs, and topics), and joins the generated sentences into a single descriptive text. The goal is to create varied and engaging course descriptions using a structured approach.

```
FUNCTION generate_description(templates, word_dict, user_title):
    INITIALIZE description AS EMPTY LIST

    FOR EACH template IN templates:
        // Generate a sentence using the template and randomly selected words from word_dict
        sentence = template.format(
            title=user_title,
            adjective=RANDOM CHOICE FROM word_dict['adjectives'],
            noun=RANDOM CHOICE FROM word_dict['nouns'],
            verb=RANDOM CHOICE FROM word_dict['verbs'],
            topic=RANDOM CHOICE FROM word_dict['topics']
        )
        APPEND sentence TO description
    END FOR

    RETURN JOIN description INTO SINGLE STRING WITH SPACES
END FUNCTION
```

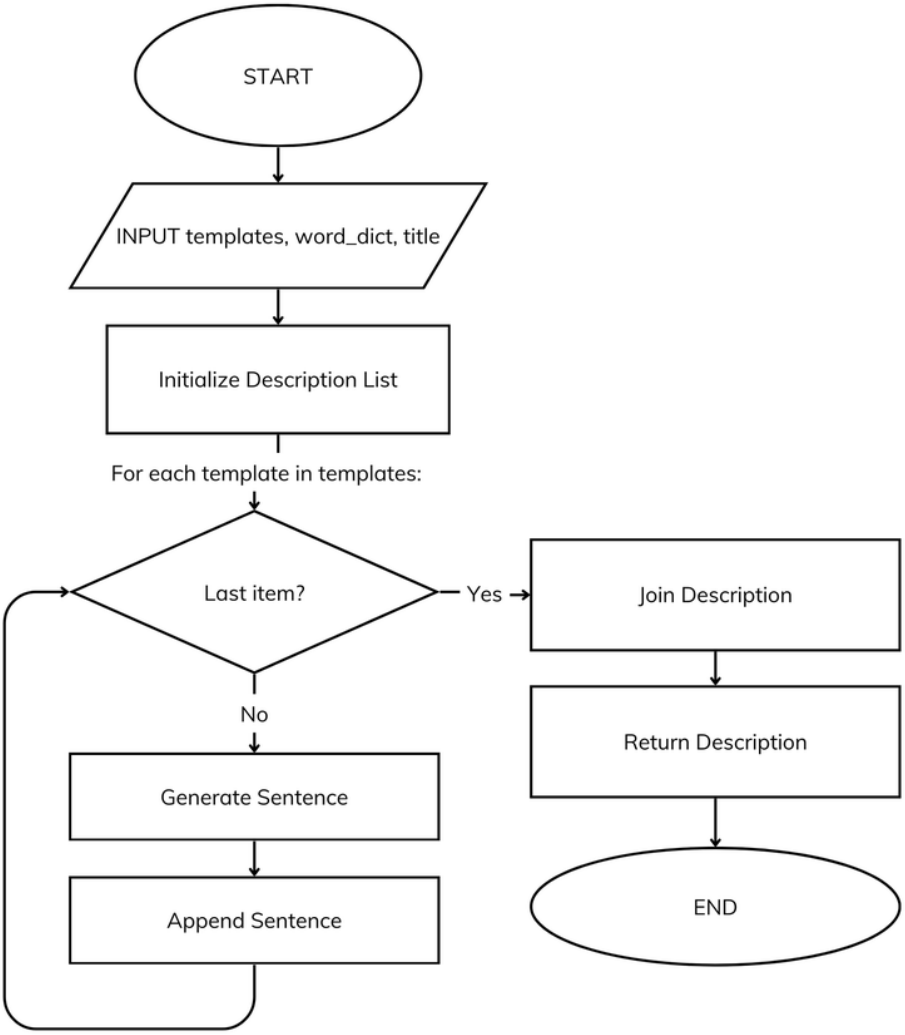


Figure 8. Flowchart for Grammar Base Automata Course Description

This pseudocode contains code for enhancing descriptions using the pre-trained BART language model. The main function, ‘enhance_description’, initializes a text-to-text generation pipeline with the ‘stanford-oval/paraphraser-bart-large’ model, splits the input description into sentences, paraphrases each sentence, and joins the enhanced sentences into a coherent text. This process improves the readability and quality of the text using advanced NLP techniques.

```
FUNCTION enhance_description(description):
    // Initialize the text-to-text generation pipeline with the specified model
    enhancer = initialize_pipeline("text2text-generation", model="stanford-oval/paraphraser-bart-large")

    // Split the description into sentences
    sentences = SPLIT description BY '.'

    // Initialize a list to store enhanced sentences
    INITIALIZE enhanced_sentences AS EMPTY LIST

    // Loop through each sentence
    FOR EACH sentence IN sentences:
        // Check if the sentence is not just spaces
        IF sentence IS NOT EMPTY:
            // Enhance the sentence using the enhancer pipeline
            enhance = enhancer(sentence + '.', max_length=60)
            // Append the enhanced sentence to the list
            enhanced_sentences.append(enhance[0]['generated_text'])
        END IF
```

END FOR

// Join the enhanced sentences into a single string with spaces

RETURN JOIN enhanced_sentences INTO SINGLE STRING WITH SPACES

END FUNCTION

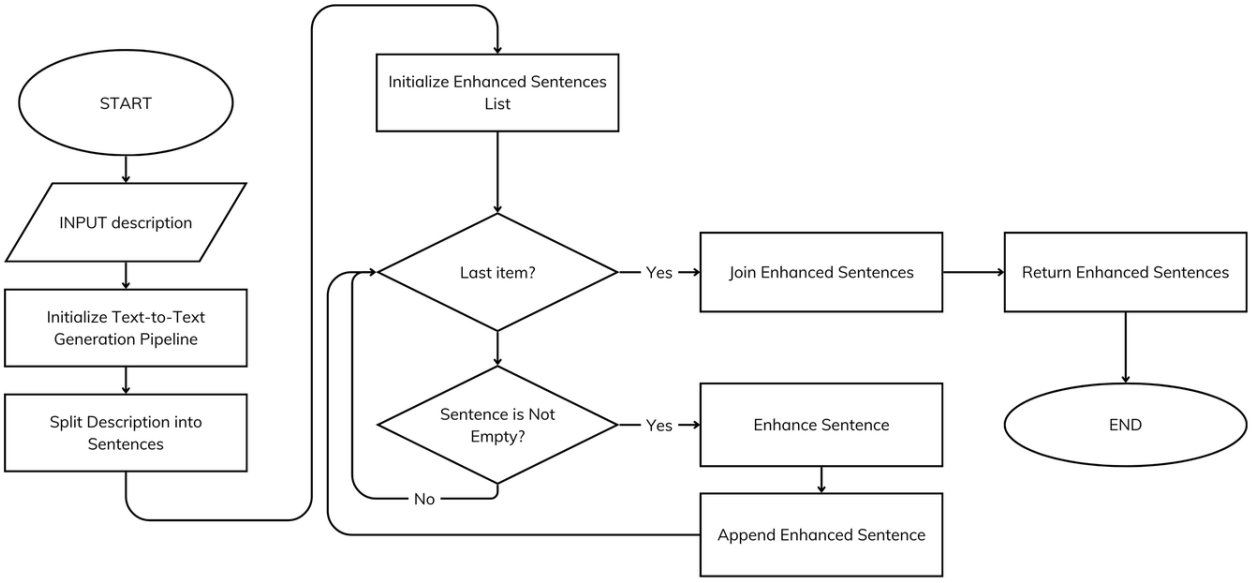


Figure 9. Flowchart for BART

Natural Language Generation

Using Natural Language Generation (NLG) in syllabus creation offers several compelling advantages for educators and institutions. One of the primary benefits is the ability of NLG to automate the content generation process, expediting the creation of clear and structured syllabi. This not only saves time for educators but also ensures a consistent and standardized presentation of course information. NLG brings efficiency to syllabus creation by automating repetitive tasks associated with text generation. Educators can leverage NLG to produce comprehensive syllabi quickly, allowing them to

allocate more time and effort to other crucial aspects of course development, such as designing engaging learning activities and assessments.

Moreover, NLG enables customization and personalization in syllabus content. Educators can tailor syllabi to individual student needs, providing a more personalized learning experience. This adaptability is valuable in accommodating diverse learning styles, preferences, and proficiency levels. The dynamic nature of educational content is another area where NLG proves beneficial. Syllabi often needs updates to reflect changes in course content, learning objectives, or assessment methods. NLG systems can be easily adapted to incorporate these changes, ensuring that syllabi remain current and relevant.

Clear and accessible communication is vital in education, and NLG contributes to creating syllabi with coherent and easily understandable language. This clarity enhances student comprehension of course expectations, objectives, and assessment criteria, fostering a positive learning environment. Furthermore, NLG systems can support multilingual capabilities, facilitating the creation of syllabi for courses taught in diverse linguistic settings. This is particularly beneficial in educational institutions with international or multilingual student populations.

N-gram

Integrating n-grams into syllabus creation can enhance the organization and coherence of your educational content. An n-gram is a contiguous sequence of 'n' items from a given sample of text or speech. For example, bigrams (2-grams) of "syllabus creation" are "syllabus creation." To begin, gather educational materials such as

textbooks, academic papers, articles, and any other relevant resources. Use text analysis tools to extract n-grams from these materials, focusing on key concepts, phrases, and terminology that frequently appear. This data-driven approach helps identify important themes and topics that should be included in the syllabus. By organizing the syllabus around these common n-grams, you ensure that the content is aligned with the core ideas and language of the subject matter, making the syllabus more relevant and easier for students to understand.

Natural Language Processing

Integrating Natural Language Processing (NLP) in syllabus creation offers numerous advantages for educators and institutions. By leveraging NLP, automated content analysis can streamline the syllabus development process. This involves the algorithmic examination of extensive textual resources, such as textbooks and scholarly articles, to identify pertinent topics, key concepts, and learning objectives that should be incorporated into the syllabus. Additionally, NLP facilitates the personalization of learning materials, allowing educators to tailor syllabi to individual student needs by analyzing natural language patterns. This personalization enhances student engagement and comprehension.

Furthermore, NLP aids in efficiently curating resources by sifting through vast amounts of educational content to identify the most relevant and up-to-date materials for inclusion in the syllabus. This ensures that students have access to the latest information in the field. Moreover, NLP helps align syllabi with learning standards and guidelines by

analyzing the language against established criteria, ensuring that educational curricula meet institutional or industry-specific requirements. Lastly, the adaptability of NLP to changing trends enables educators to create syllabi that remain dynamic and responsive to evolving educational landscapes and subject matter. Incorporating NLP in syllabus creation enhances efficiency, personalization, resource curation, standards alignment, and adaptability, ultimately contributing to a more effective and responsive educational experience.

Operation and Testing Procedure

The system must be free of bugs and errors that could affect the system processes. Software testing was implemented to anticipate issues with user experience and ensure the software's functionality in the future.

1. Upon opening the system, the user will be able to see the home page of the system.
2. The Home page includes the name and the features of the system.
3. In Generate, the input box is shown, asking for input. The process is as follows:
    a. Insert an e-book in PDF format that the user would like to generate.
    b. Select the most desired PDF from the inputted.
    c. Select the type of semester and how many exams.
    d. After processing the syllabus, the output will pop up.
4. On the edit page, users can modify the generated generic syllabus.

Functional Testing

Functional testing is a software testing procedure that assesses the functionality of a system. It is performed to confirm that all application features are operating as anticipated.

1. Determine the functions of every feature.
2. Adjust the input data accordingly to the requirements of the feature.
3. Verify the result output of the designated feature.
4. Run the test cases.
5. Compare the test results to the expected result.
6. Evaluate the test results, check if it passes or fail.

Each feature will be tested using a test case to document the performance tests. The results were evaluated to test the functionality.

Table 1. *Test Case Form (Sample)*

Test Case ID	TC – F1	**UC Reference**	Users
Objective	To generate generic syllabus within the inputted e-book		
Preconditions	Sample course data (e.g., course name, description, learning outcomes, etc.) is available for syllabus creation.		
Postconditions	The generated syllabus is downloadable in a specified format (e.g., PDF).		

Actions	**Expected Results**
1. Open the system. 2. Press the Generate tab. 3. Input the e-book, select the most preferred e-book, input the type of semester and how many exams. 4. Press Generate button.	● The input fields should accept the sample course data. ● On clicking the 'Generate Syllabus' button, the application should process the input and generate a syllabus. ● The generated syllabus should accurately reflect the input data, adhering to the predefined template and format.

Table 1 contains essential information related to test cases for a particular software application. This table's purpose is to document each test case's details, including its identification, objective, preconditions, actions, expected output, and status.

1. Test Case ID, a code that distinguishes the test case specifically.
2. UC References, which identifies the necessary document reference that formed the basis for the case.
3. Objectives, which specify the test scenarios.
4. Preconditions, which conditions must be satisfied before carrying out the test method.
5. Actions, describing the steps necessary to carry out the test case.
6. Expected Output, which describes what is likely to happen throughout the test, and.

7. Status, which displays the appropriate severity and priority markings to show if the application passed or failed the test case.

Evaluation Procedure

The evaluation instrument that was used to assess the acceptability of the system was adapted from the ISO 25010 titled "Systems and software engineering – Systems and software Quality Requirements and Evaluation (SquaRE) – System and software quality models."

The following procedures were conducted to evaluate the acceptability of the desktop application:

1. Each respondent was provided a software assessment form to measure the system's performance.
2. A demonstration of the system's functions and a discussion regarding its objective was conducted to illustrate the usage of the application.
3. All respondents were requested to use the application.
4. The respondents were asked to evaluate the system using the ISO 25010 standard evaluation criteria using a 4-point Likert scale, as shown in Table 2, where (4) is the highest and (1) is the lowest.
5. The overall weighted mean rating for each criterion and the grand weighted mean will be computed based on the collected evaluation.

6. The evaluation results will utilize the range of weighted mean values presented in Table 3 and the qualitative interpretation corresponding to that range.

Table 2. *Four- Point Likert Scale*

Numerical Rating	Qualitative Interpretation
4	Highly Acceptable
3	Very Acceptable
2	Fairly Acceptable
1	Not Acceptable

Table 3. *Range of Weighted Mean Ratings and its Qualitative Representation*

Numerical Rating	Qualitative Interpretation
3.4 – 4.0	Highly Acceptable
2.6 – 3.3	Very Acceptable
1.8 – 2.5	Fairly Acceptable
1.0 – 1.7	Not Acceptable

Functional Suitability

The testing was conducted to check if the application is complete, correct, and appropriate. The following were considered:

1. Conduct testing to ensure that the application includes all essential functions, such as syllabus customization, topic inclusion, learning objective specification, and other necessary features for syllabus creation.
2. Verify the correctness of the application's outputs by comparing the generated syllabi against predefined templates and expected outcomes. Test scenarios should cover various inputs and conditions to ensure accurate and precise results.
3. Test whether the application's functions align with the needs and workflows of educators creating syllabi. Evaluate the user interface, workflow efficiency, and overall user experience to determine the appropriateness of the application's functionalities.

Performance Efficiency

To assess the system's efficiency, considering various factors, the following aspects were taken into account during the application testing:

1. Assess how quickly the application responds to user inputs, processes syllabus-related tasks, and achieves the desired throughput rates. Ensure these aspects align with the predefined performance standards for efficient syllabus creation.

2. Monitor and analyze how efficiently the application utilizes system resources to perform syllabus-related functions. Verify that resource usage is optimized without exceeding acceptable levels.
3. Test the application's capacity by pushing it to its maximum limits to ensure it can handle the expected workload associated with syllabus creation without breaching predefined limits.

Usability

The evaluation centered on key sub-characteristics within the Usability category, explicitly focusing on effectiveness, efficiency, and user satisfaction; the following aspects were taken into account during the application testing:

1. Determine if users can quickly identify whether the application is appropriate for creating and managing syllabi.
2. Assess the attributes of the application that contribute to its ease of operation and control during syllabus creation tasks.
3. Test the mechanisms in place within the application to protect users from errors during the syllabus creation process.
4. Evaluate the visual and interactive elements of the application to ensure a positive and enjoyable user experience.
5. Assess the inclusivity of the application by ensuring that users with diverse needs and abilities can effectively use it.

Chapter 4

RESULTS AND DESCUSSION

This chapter contains the project description, project structure, project capabilities and limitations, and results of evaluation for the project.

Project Description

This study developed a desktop application software called Syllagenius, aimed at simplifying the process of syllabus creation. By incorporating Natural Language Processing, particularly our system's unique content generation capabilities. The answer to the difficulties educators face in creating a syllabus is Syllagenius. The objective is to offer educational institutions and instructors a user-friendly tool that makes creating a syllabus simple.

Figure 10. Home 1

Project Structure

The project structure of our developed desktop application software called Syllagenius, was designed to simplify the process of syllabus creation. Our desktop application software begins with a home page, as shown in figure 6. This first step provides a brief introduction to the software's purpose and main features, ensuring that new users are ready for their experience with the app.

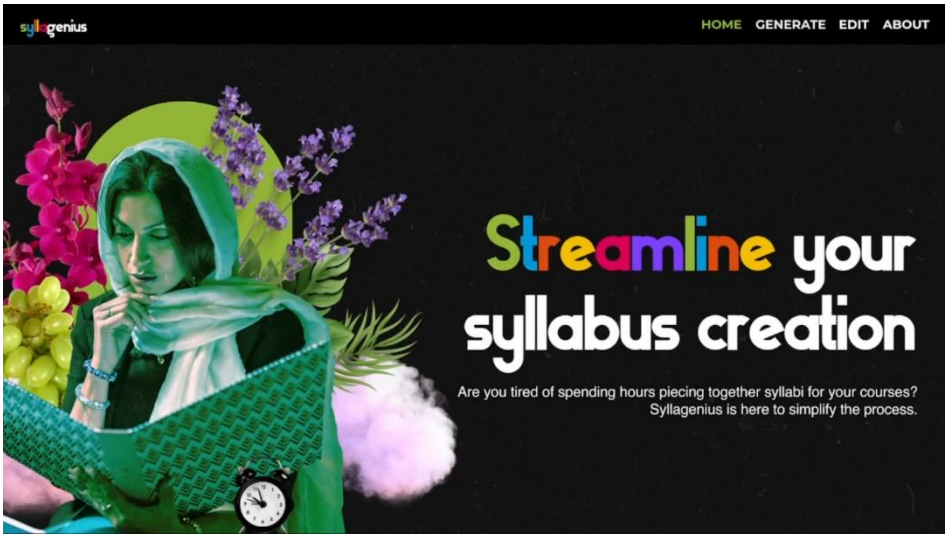


Figure 11. Home 2

Users are taken to a screen where they can fill out a form and upload their e-book for processing after clicking the "generate" button. Our desktop application utilizes Natural Language Processing to streamline the process of syllabus creation, to make it simpler and more user-friendly. this procedure is shown in Figure 7.

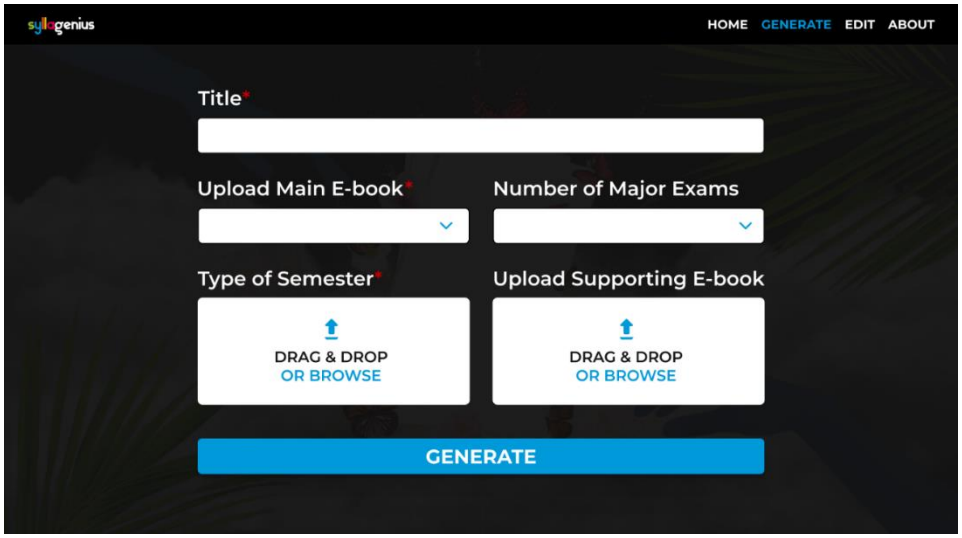


Figure 12. Generate Page

After entering the necessary data and uploading their e-books the users will be redirected to a preview page. Here they can view the semester type and length in weeks visible on the right side of the screen, while the scanned chapters are visible on the left side, the users can also see how many major exams they have selected. The number of major exams picked, and the type of semester selected yield the following results. Refer to Figure 8-15 for visual representation.

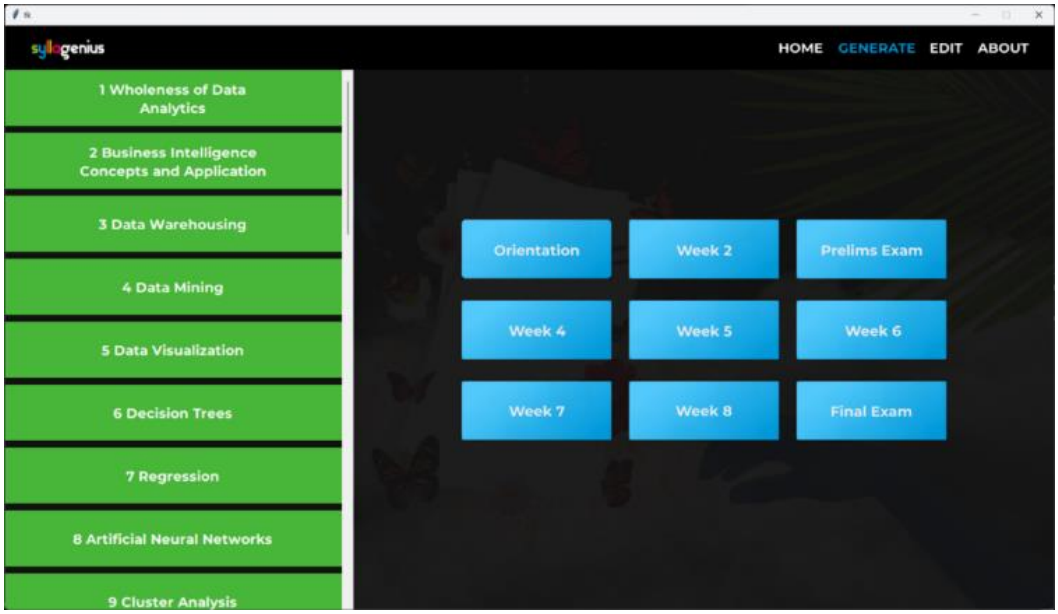


Figure 13. Quarter - 2 Major

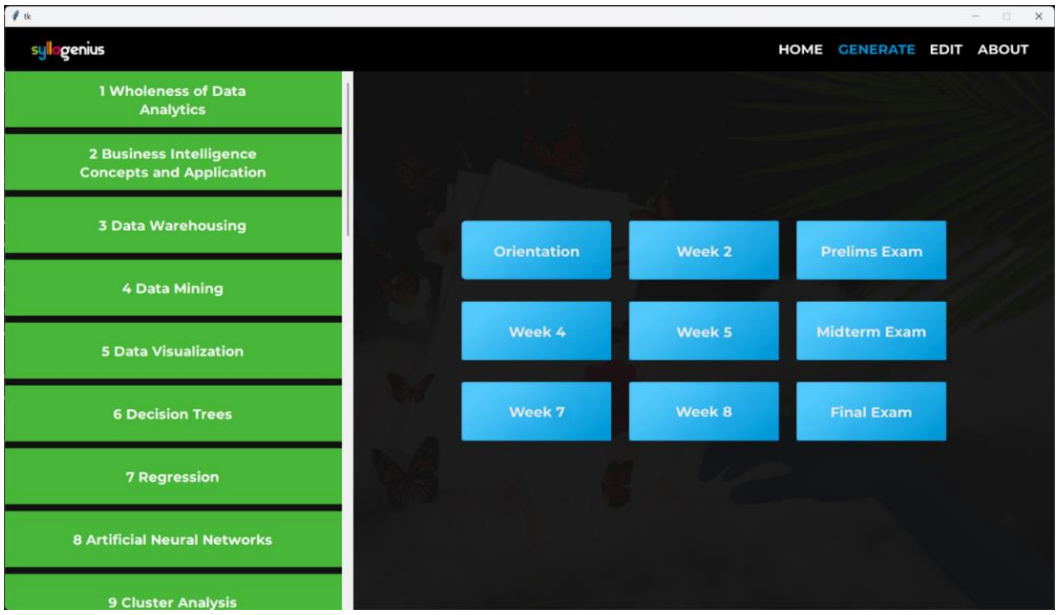


Figure 14. Quarter - 3 Major

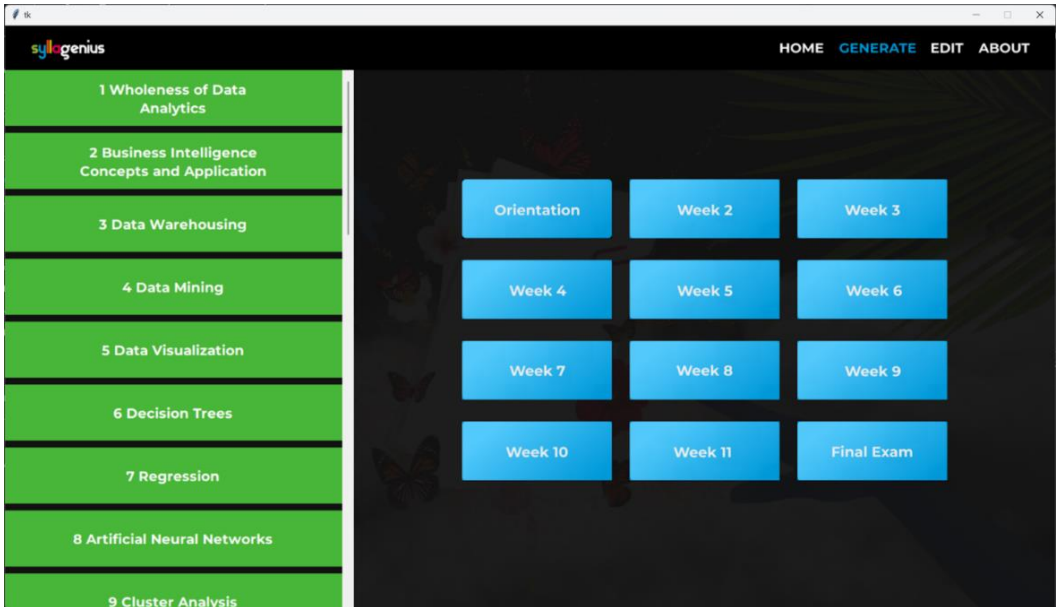


Figure 15. Trimester - 1 Major

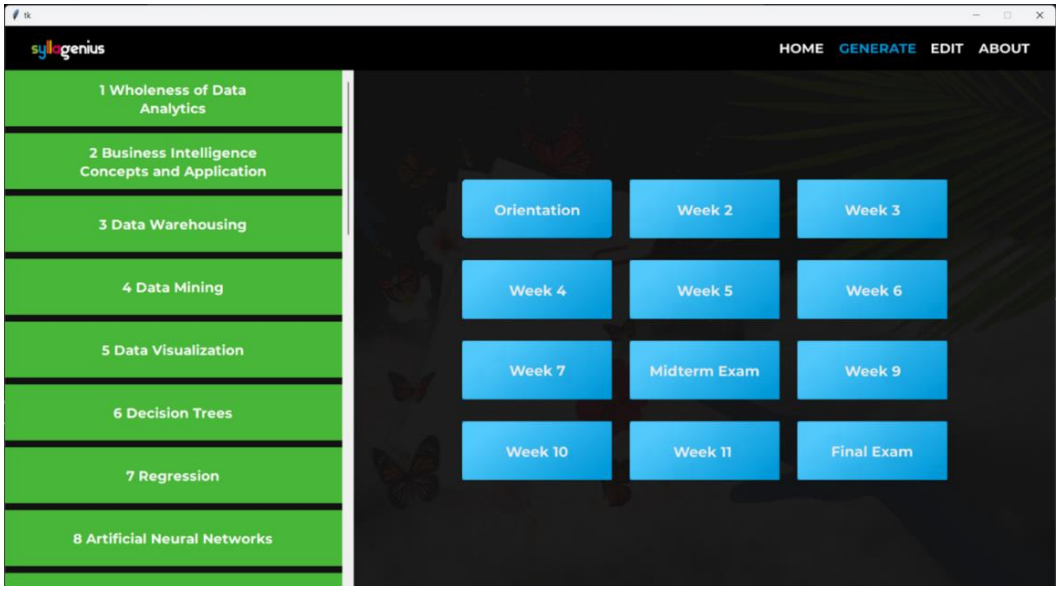


Figure 16. Trimester - 2 Major

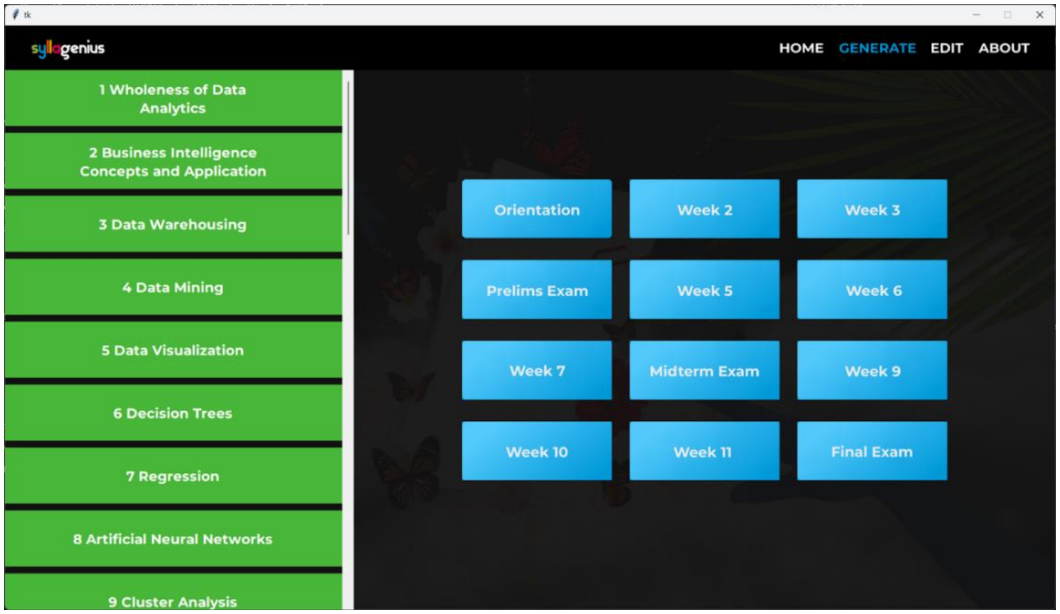


Figure 17. Trimester - 3 Major

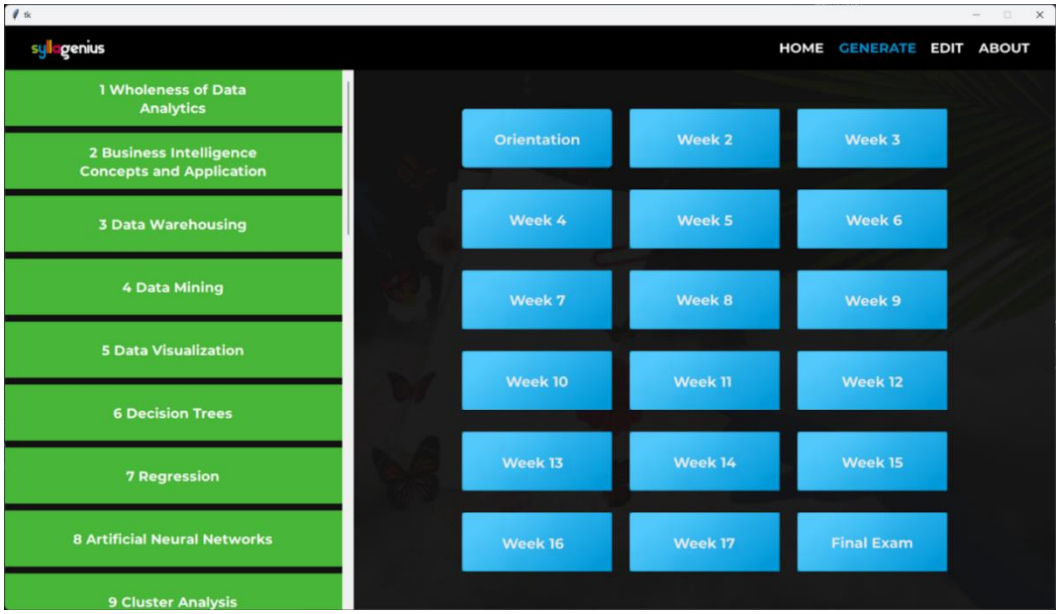


Figure 18. Semestral - 1 Major

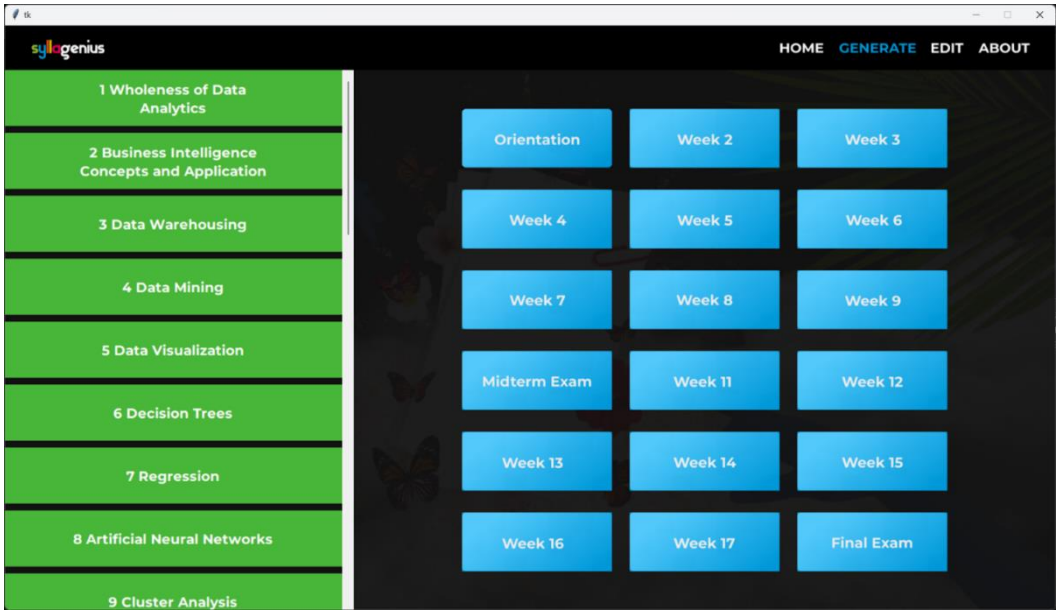


Figure 19. Semestral - 2 Major

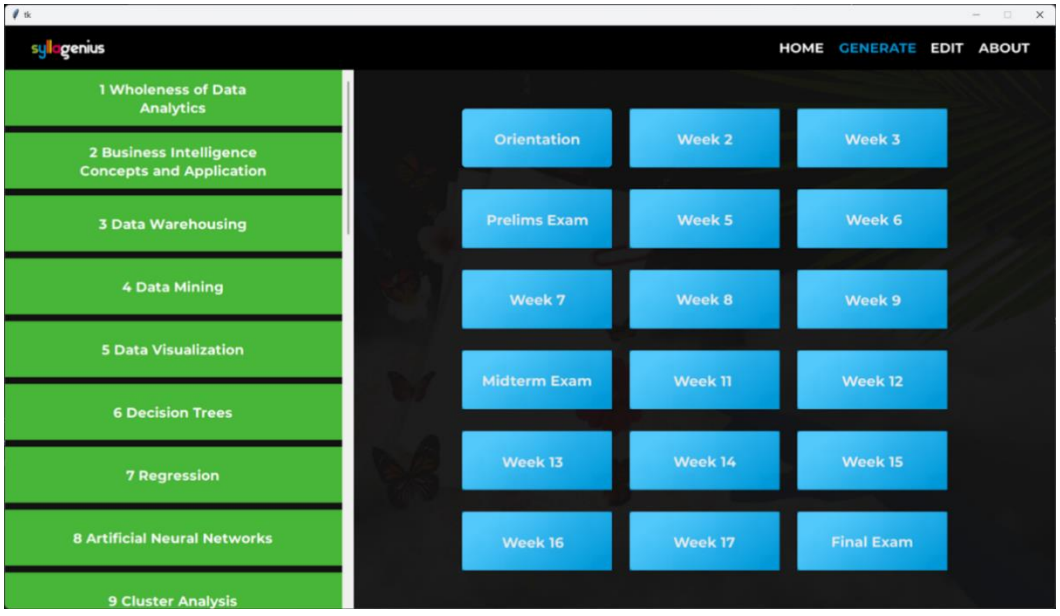


Figure 20. Semestral - 3 Major

The user can begin the editing process by either clicking the right arrow button on the preview page or by directly clicking the edit button located above. Within the edit interface the user can fill out and select his/her desired recitation, project, and attendance, along with their respective percentages. Furthermore, the edit page also displays the generated syllabus output below, for the user to review, and position at the right side are two buttons to save the generated syllabus in either DOC or PDF format, to provide flexibility and convenience in document management.

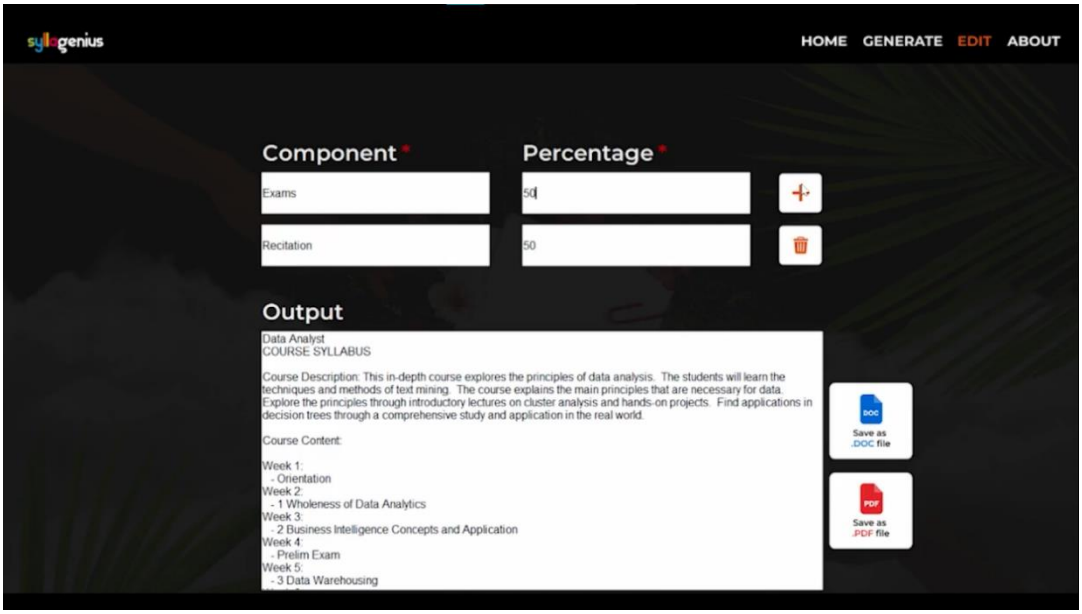


Figure 21. Edit Page

Project Capabilities and Limitations

The following are the capabilities of the developed desktop application software:

1. By using user-provided PDF e-books, Syllagenius streamlines the process of creating a syllabus.
2. The program creates course objectives, content, descriptions, and references automatically while following APA 7 guidelines.
3. The use of Bloom's Taxonomy guarantees that cognitive levels are taken into account, which improves the learning process.
4. increases the effectiveness of educational planning and delivery while saving educator’s time.
5. Syllagenius makes it possible to create syllabi instantly, enabling more streamlined and effective syllabus preparation procedures.

The following are the limitations of the developed desktop application software:

1. One of the limitations of our application is the reliance on Bloom's Taxonomy it may limit the applicability in contexts favoring alternative pedagogical theories.
2. Restriction to PDF e-books as input materials is also one of the limitations because it may exclude other valuable educational resources not available in this format, missing out on important possibilities and viewpoints that could improve students' educational experiences.
3. Currently only the DOC or PDF file formats are available for users to save their created syllabus.

4. The way the application currently sets up limits the number of main ebooks and supporting ebooks that may be added, which could limit the system's functionality and adaptability.

Test Results

The Tkinter framework and Python programming language were used in the development of Syllagenius to ensure compatibility with various operating systems. The results of experiments carried out to evaluate the suitability, efficiency, and reliability of Syllagenius. shown in the following tables.

Table 4. *Test Results*

Test Case	Steps Undertaken	Observed Results
Generate process	1. Opened the application “SYLLAGENIUS:AN AI GENERIC SYLLABUS GENERATOR USING NLP” 2. Clicked on Generate button above. 3. Filled out the required fields: Title, Type of semester, and No. of major exams. 4. Uploaded the desired main e-books and supporting e-books. 5. Clicked on the Generate button below. 6. The users will be redirected to a preview page where they can select a specific week, and the corresponding topics will be highlighted on the left	After opening the "SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP" application, the user clicked the "Generate" button and filled out the required fields, including Title, Type of semester, and Number of major exams. They then uploaded the main e-books and supporting e-books. Upon clicking the "Generate" button again, the user was redirected to a preview page where they could select a specific week, with the corresponding topics highlighted on the left side.

	side.	
Edit process	1. The users can enter this process by clicking the right arrow button on the preview page after generating the syllabus, or by directly selecting the edit button above. 2. The users can complete the component field by entering the recitation, project, and attendance details along with their corresponding percentages. 3. The edit page also displays the generated syllabus output below. 4. On the right side of the output area, there are two buttons that allow one to save the generated syllabus as either a DOC file or a PDF file.	The user was able to enter the editing process by clicking the right arrow button on the preview page after generating the syllabus or by directly selecting the edit button above. On the edit page, the user can complete the component fields by entering details for recitation, project, and attendance along with their corresponding percentages. The generated syllabus output is displayed below, and on the right side of the output area, there are two buttons that allow the user to save the syllabus as either a DOC file or a PDF file.

Evaluation Results

To assess the application's acceptability, it underwent an evaluation process. The following respondents were purposively selected: professors, IT professionals, and individuals possessing expertise in both fields. The evaluation results summary is presented in the Table 5.

Table 5. *Summary of Evaluation (Desktop Application Software)*

Criteria	Weighted Mean	Description
A. Functional Suitability		
1. Functional Completeness	3.76	Highly Acceptable
2. Functional Correctness	3.56	Highly Acceptable
3. Functional Appropriateness	3.71	Highly Acceptable
Criterion Weighted Mean	*3.67*	*Highly Acceptable*
B. Performance Efficiency		
1. Time Behavior	3.88	Highly Acceptable
2. Time Behavior	3.79	Highly Acceptable
3. Resource Utilization	3.79	Highly Acceptable
4. Capacity	3.76	Highly Acceptable
Criterion Weighted Mean	*3.80*	*Highly Acceptable*
C. Reliability		
1. Maturity	3.76	Highly Acceptable
2. Availability	3.85	Highly Acceptable
3. Recoverability	3.85	Highly Acceptable
Criterion Weighted Mean	*3.82*	*Highly Acceptable*
Grand Weighted Mean	**3.76**	**Highly Acceptable**

Table 5. shows that the "SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP" achieved its highest rating in "Reliability," described as "Highly Acceptable." With a weighted mean of 3.82, This result shows that the software performed well under different tasks given circumstances and time constraints.

And secondly, the rating for "Performance Efficiency," with Criterion Weighted Mean of 3.80, described as "Highly Acceptable." This result shows that the software performs relative to the resources utilized under defined conditions.

Lastly, the software received its lowest rating in "Functional Suitability," with Criterion Weighted Mean of 3.67, but still described as "Highly Acceptable." This result shows that the software meets both stated and implied needs when used in specific conditions.

Chapter 5

SUMMARY OF FINDINGS, CONCLUSIONS, AND RECOMMENDATIONS

This chapter presents the summary of findings, the conclusions drawn from the findings and the corresponding recommendations for further understanding of the project.

Summary of Findings

Syllagenius: An AI Generic Syllabus Generator Using NLP" is an innovative tool designed to automate and optimize the creation of academic syllabuses using natural language processing (NLP) and machine learning techniques. Here are the key findings and features of this AI-driven syllabus generator.

- Test Results

1. **Suitability.** The test results have shown that the system is complete, correct, and appropriate. The system includes all essential functions, such as syllabus customization, topic inclusion, learning objective specification, and other necessary features for syllabus creation. The correctness of the application's outputs has been verified by comparing the generated syllabi against predefined templates and expected outcomes. Test scenarios covered various inputs and conditions, ensuring accurate and precise results. Additionally, the application's functions align well with

the needs and workflows of educators creating syllabi. The user interface, workflow efficiency, and overall user experience were evaluated, confirming the appropriateness of the application's functionalities.

2. **Efficiency.** The test results have shown that the application's efficiency meets the required standards. The application responds quickly to user inputs, processes syllabus-related tasks efficiently, and achieves the desired throughput rates, aligning with predefined performance standards for efficient syllabus creation. The application efficiently utilizes system resources to perform syllabus-related functions, with resource usage optimized and within acceptable levels. Additionally, the application's capacity was tested by pushing it to its maximum limits, ensuring it can handle the expected workload associated with syllabus creation without breaching predefined limits.

3. **Reliability.** The test results have shown that the application meets the key sub-characteristics within the Usability category, focusing on effectiveness, efficiency, and user satisfaction. Users can quickly identify that the application is appropriate for creating and managing syllabi. The application's attributes contribute to its ease of operation and control during syllabus creation tasks. The mechanisms in place protect users from errors during the syllabus creation process. Additionally, the visual and interactive elements of the application ensure a positive and enjoyable user experience. The application is inclusive, allowing users with diverse needs and abilities to use it effectively.

- Evaluation Results

    1. **Functional Suitability.** The evaluators rated the system as highly acceptable regarding its capacity to be generally responsive and functional.
    2. **Performance Efficiency.** The evaluators rated the system as highly acceptable regarding its capacity to be generally efficient, with fast response times for most user interactions.
    3. **Reliability.** The evaluators rated the system as highly acceptable regarding its reliability in handling various e-books and Word/PDF files to create a syllabus.

Conclusions

The following conclusions were derived from the above findings:

1. The developed system was successfully designed to have the following features:
    a. Allowed the users to type a title, choose what semestral system and number of major exams, and upload a main syllabus and a supporting syllabus to generate them.
    b. Allowed the users to manage chapters and weeks.
    c. Allowed the users to add a component and its percentage and to edit the syllabus output without restrictions.
2. The system was successfully developed using various tools, including Visual Studio Code, Tkinter, Anydesk, NLTK, BART, Natural Language Toolkit,

Unicode, Difflib, PyMuPDF, Regular Expression Library, Scikit-learn, NumPy, and Python.

3. The test results showed that the desktop application is reliable to the users.
4. The system was evaluated to be highly acceptable in terms of Functional Suitability, Performance Efficiency, and Reliability which proves that the system can be helpful to educators and can benefit them.

Recommendations

The following recommendations are put forward for further enhancement of the developed desktop application.

1. Utilize advanced Natural Language Processing (NLP) techniques such as sentiment analysis, topic modeling, and named entity recognition (NER) to refine the syllabus generation process and ensure more accurate and meaningful content extraction.
2. Implement a more interactive user interface with drag-and-drop functionality and real-time preview of the generated syllabus to enhance user engagement and ease of customization, making the application more user-friendly and efficient.
3. Add features that enable collaborative editing and sharing of syllabi among multiple users, facilitating teamwork and ensuring consistency and diverse input in the creation and modification of syllabi.
4. Support additional input formats such as Word documents and web URLs, and provide output options in various formats like PDF, Word, and HTML to increase

the versatility and accessibility of the application for different user preferences and needs.

5. Integrate machine learning algorithms that learn from user feedback and improve the accuracy and relevance of the generated syllabi over time, enhancing the system's ability to adapt and meet user expectations effectively.
6. Perform extensive usability testing with a diverse group of educators from various disciplines to gather detailed feedback and identify potential areas for improvement, ensuring the application meets the needs of its target audience and functions effectively in real-world scenarios.
7. Design the application with a modular architecture that allows for easy updates and integration of new features, ensuring the system remains flexible, scalable, and maintainable for future enhancements and adaptations.

Appendix A

FUNCTIONAL SUITABILITY AND PERFORMANCE EFFICIENCY TEST RESULTS

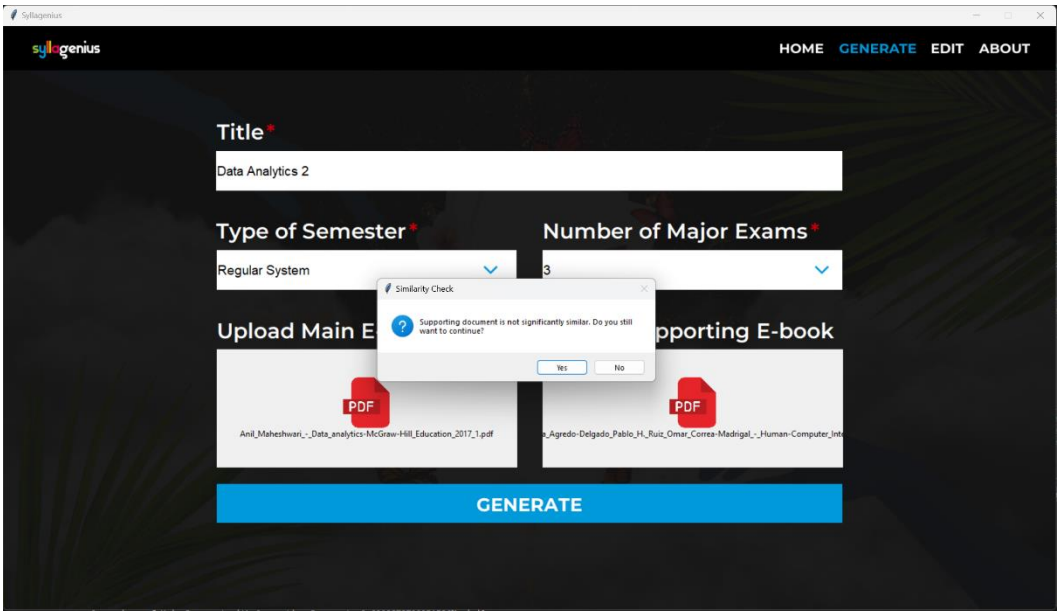


Figure 22. Desktop Application Functional Suitability Test Result on Generate Input

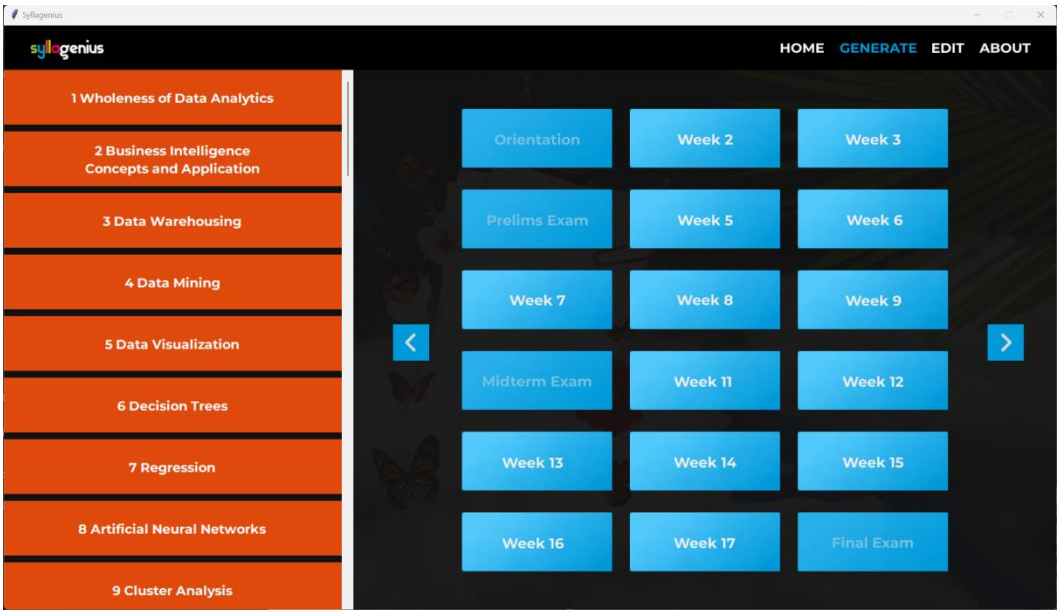


Figure 23. Desktop Application Functional Suitability Test Result on Generate Selection

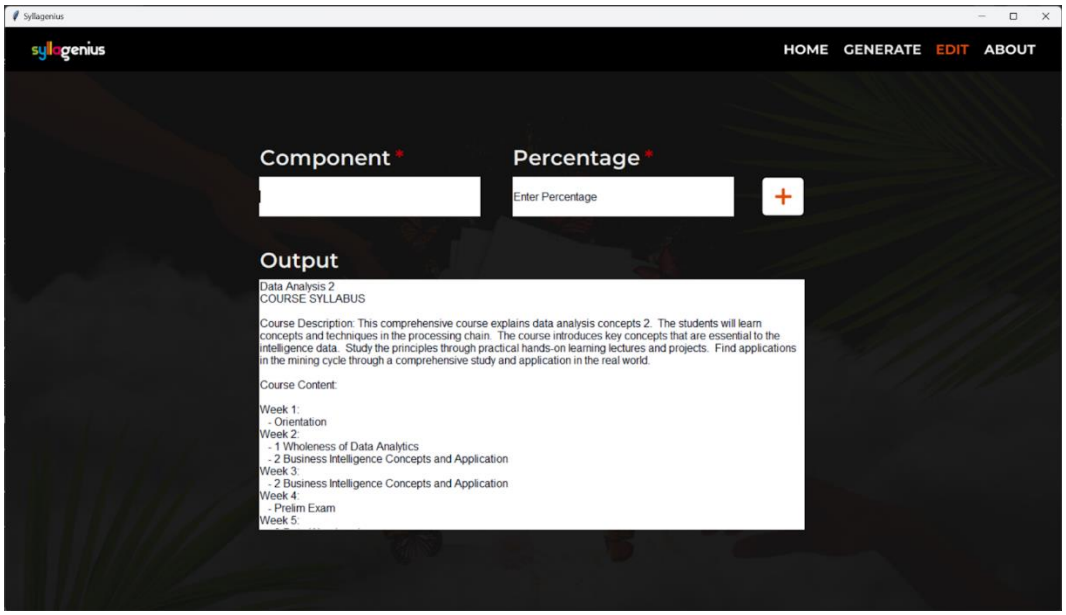


Figure 24. Desktop Application Performance Efficiency Test Result on Edit Output

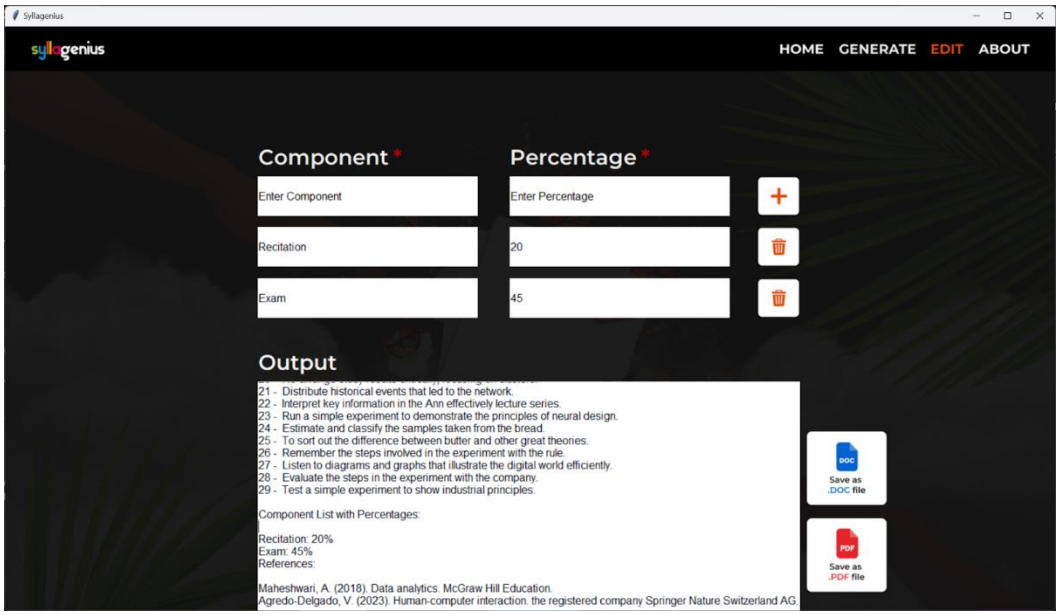


Figure 25. Desktop Application Functional Suitability Test Result on Edit Process

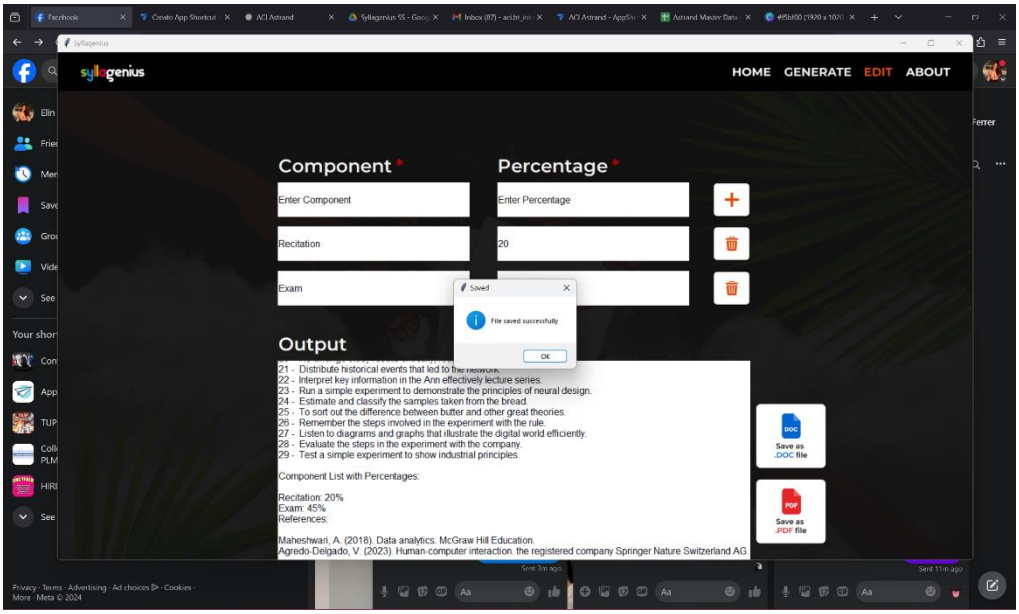


Figure 26. Desktop Application Functional Suitability Test Result on Edit Save

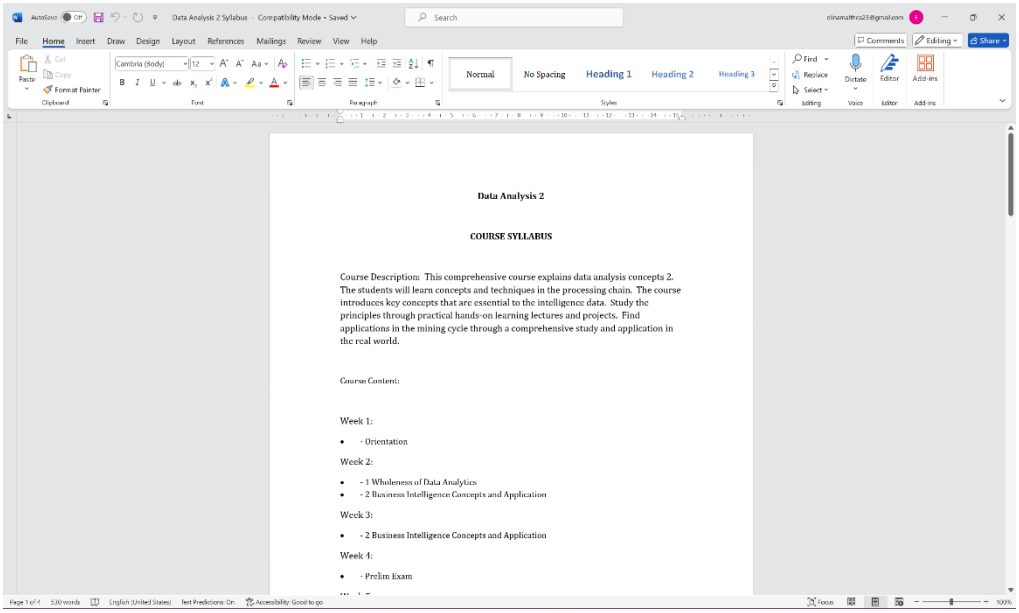


Figure 27. Desktop Application Performance Efficiency Test Result on Word Save

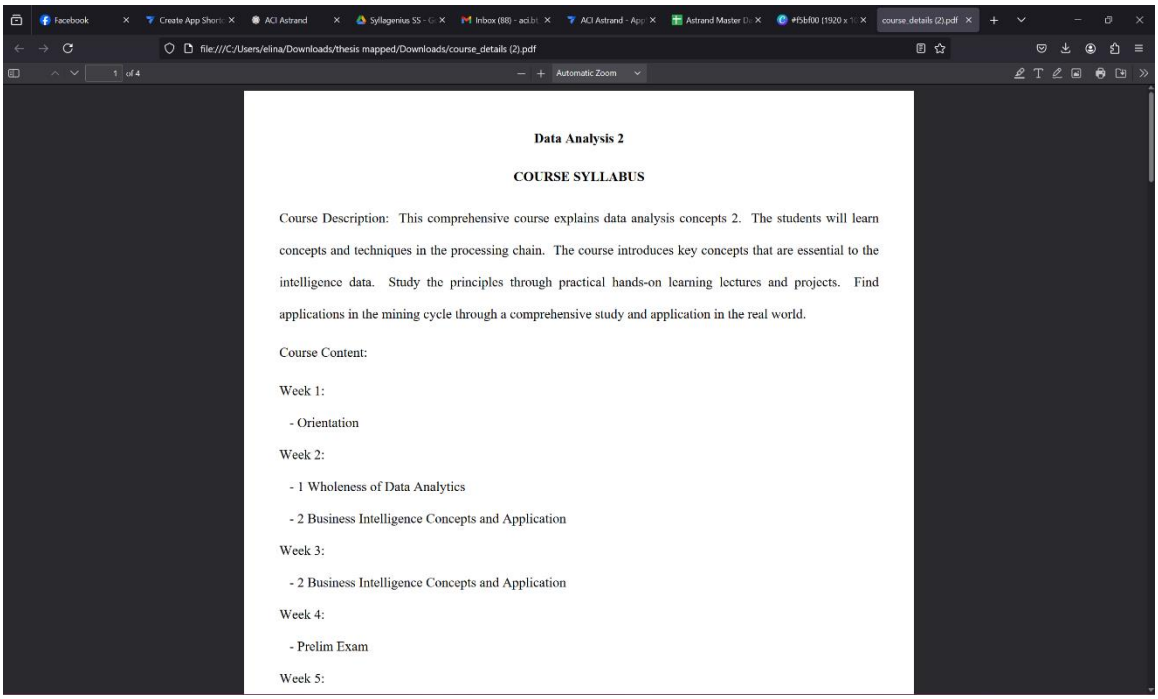


Figure 28. Desktop Application Performance Efficiency Test Result on PDF Save

Appendix B

SURVEY QUESTIONNAIRE FORM

SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP

Syllagenius introduces an innovative approach to syllabus creation, leveraging NLP and Python programming within a user-friendly desktop application. Beyond mere content extraction, it intricately designs course objectives, outlines, descriptions, and references—all meticulously aligned with APA 7 standards. This holistic process not only ensures efficiency but also upholds academic rigor and citation standards, enhancing the quality and consistency of syllabi. The interface, crafted with Tkinter, prioritizes accessibility, enabling educators of all technical backgrounds to effortlessly navigate the platform and embrace the benefits of technology in educational planning.

In addition, by submitting this survey form and in accordance with the Data Privacy Act of 2012, we will be asking for your consent to the processing of your personal information. Please be informed that all information gathered shall be treated with confidentiality to protect the privacy.

kento.magi@gmail.com Switch account

* Indicates required question

Email *

Your email

By checking "I agree," You hereby consent to the following claims by completing this survey: *

1. Please ensure that the provided personal information is accurate, as it will serve as the basis for all communications with you regarding the procedure and any additional information.
2. You can trust us to respect your privacy and use your information solely for the specified purposes.

☐ Yes, I agree.

Personal Information

First Name

Your answer

Middle Name

Your answer

Surname

Your answer

Type of Respondent *

- [ ] College Professor
- [ ] IT Professional
- [ ] Both
- [ ] Others

"If 'others' is selected, please provide further details."

Your answer

Syllagenius Demo

Youtube Link: https://www.youtube.com/watch?v=uDXSN6CqpxI

Evaluate the system features of the proposed system. Please rate the level by selecting the desired rate that corresponds to your assessment. Use the scale below.

Numeric Value Equivalent Rating:

4 -Highly Acceptable

3 -Very Acceptable

2 -Acceptable

1 -Not Acceptable

How does the developed system comply with the ISO 25010 software quality standards in terms of the following factors:

A. Functional Suitability

This characteristic represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

1. The Syllagenius effectively address all the tasks and objectives set by users for generating syllabi efficiently. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

2. The Syllagenius deliver accurate results with the required level of precision. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

3. The Syllagenius facilitates the accomplishment of specified tasks and objectives. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

B. Performance Efficiency

This characteristic represents the degree to which a product performs its functions within specified time and throughput parameters and is efficient in the use of resources (such as CPU, memory, storage, network devices, energy, materials...) under specified conditions.

4. The Syllagenius rate of speed in generating syllabus is within the specified time. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

5. The Syllagenius respond and process tasks within acceptable timeframes, meeting the necessary requirements. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

6. The Syllagenius amounts and types of resources used when performing its functions, meet requirements. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

7. The Syllagenius maximum limits of parameter meet requirements. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

C. Reliability

Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

8. The Syllagenius meets the needs for reliability under normal operation. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

9. The Syllagenius is operational and easily accessible when required for use. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

10. The Syllagenius can recover the data directly affected and re-establish the desired state. *

	1	2	3	4	
Not Acceptable	○	○	○	○	Highly Acceptable

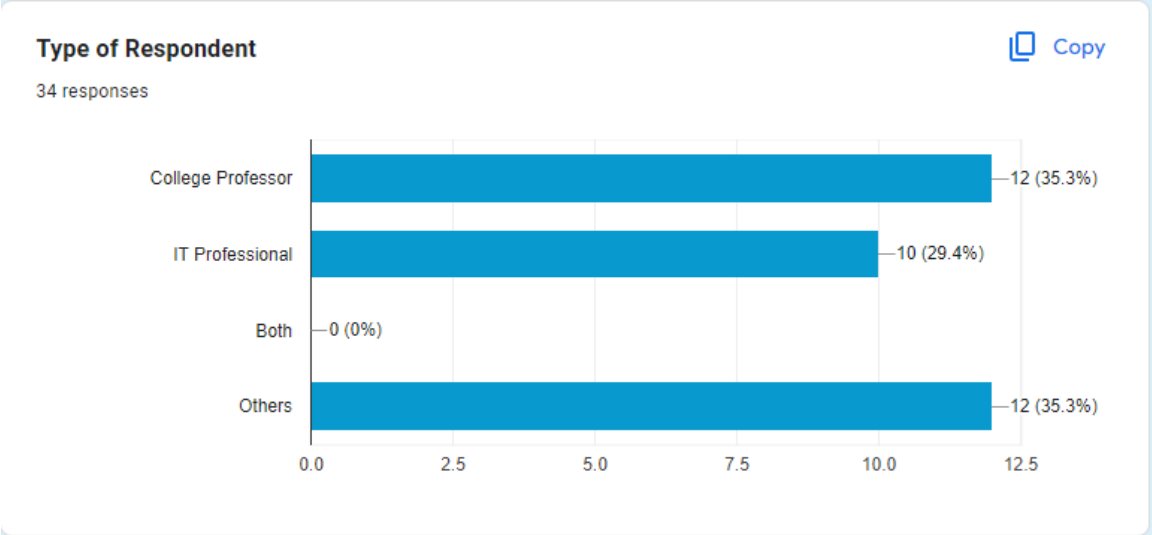
Any Comments / Suggestions? *

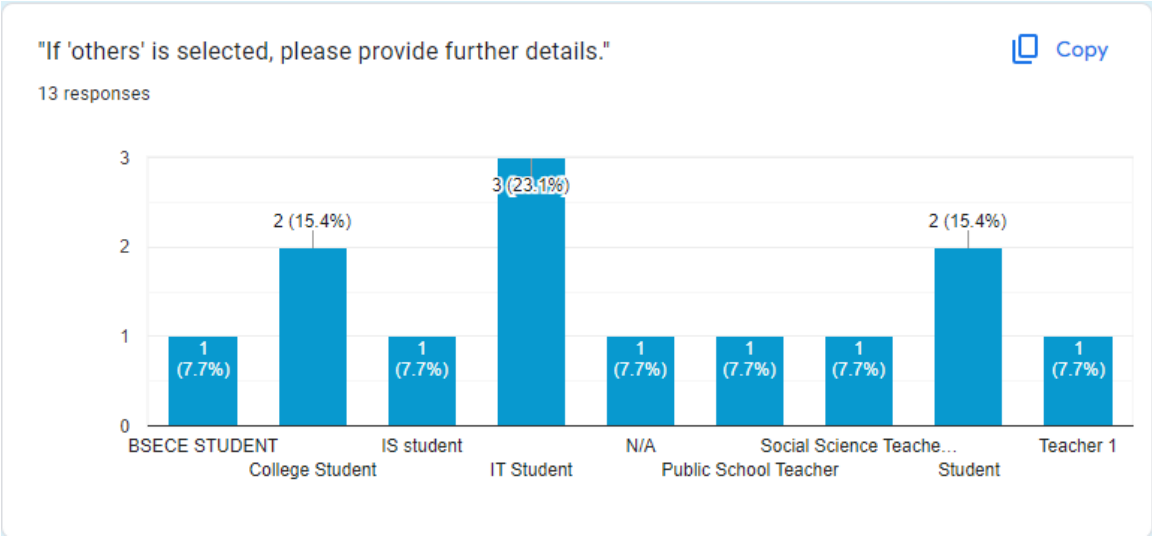
Your answer

Back | Submit | Page 4 of 4 | Clear form

Appendix C

RESPONDENTS' RESULT SHEETS



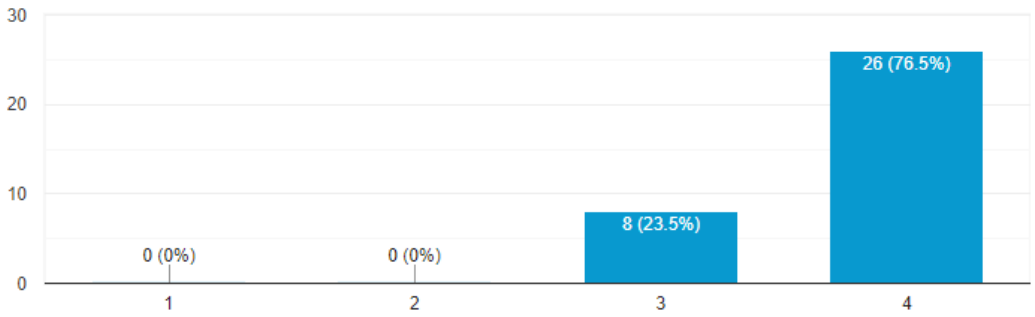


A. Functional Suitability

1. The Syllagenius effectively address all the tasks and objectives set by users for generating syllabi efficiently.

Copy

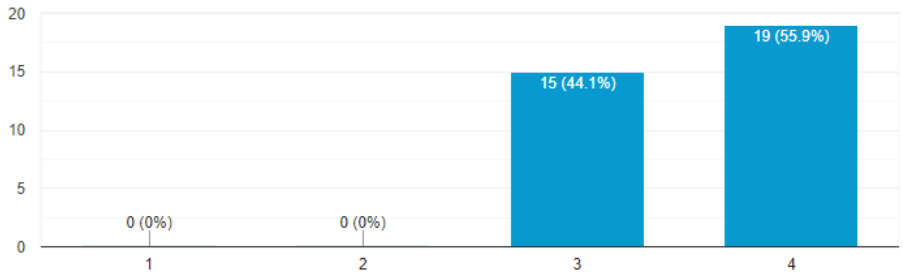
34 responses



2. The Syllagenius deliver accurate results with the required level of precision.

Copy

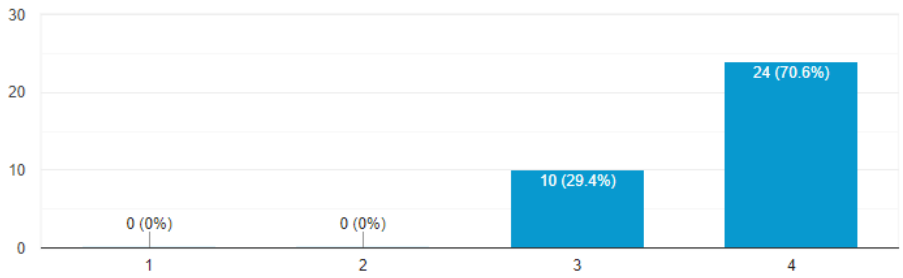
34 responses



3. The Syllagenius facilitates the accomplishment of specified tasks and objectives.

Copy

34 responses

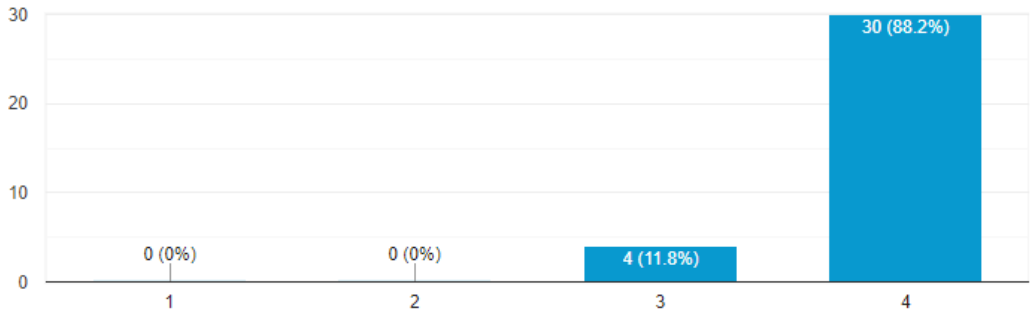


B. Performance Efficiency

4. The Syllagenius rate of speed in generating syllabus is within the specified time.

Copy

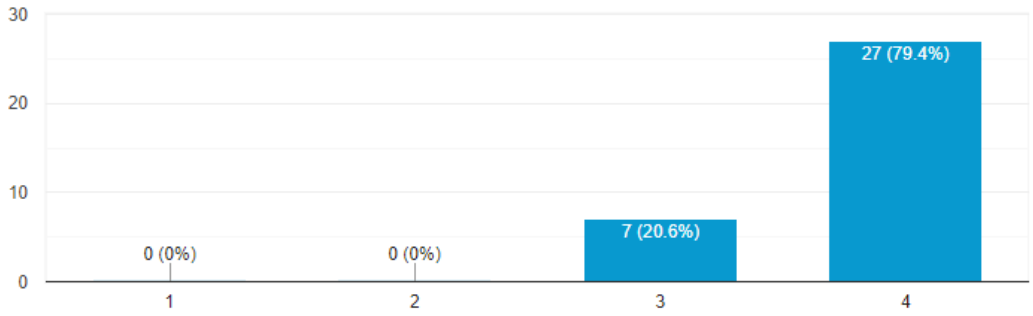
34 responses



5. The Syllagenius respond and process tasks within acceptable timeframes, meeting the necessary requirements.

Copy

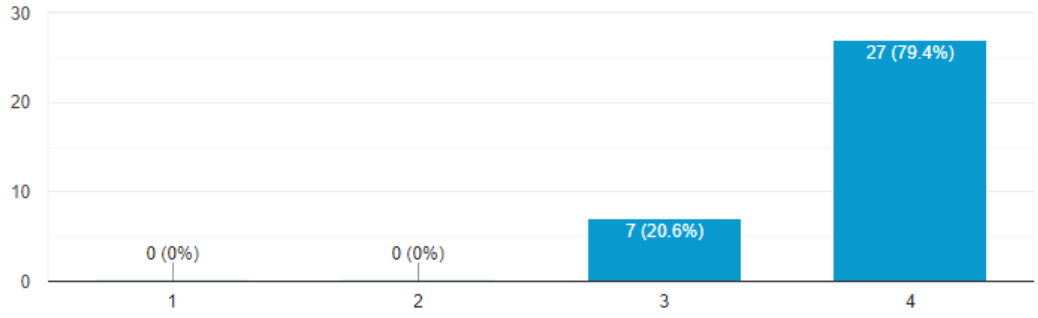
34 responses



6. The Syllagenius amounts and types of resources used when performing its functions, meet requirements.

Copy

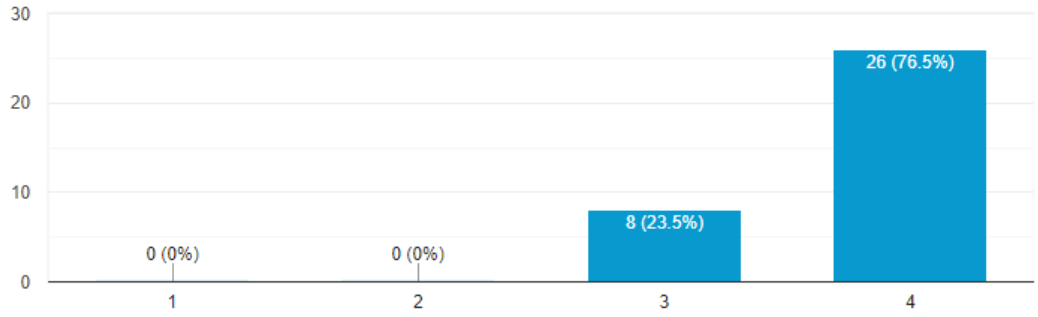
34 responses



7. The Syllagenius maximum limits of parameter meet requirements.

Copy

34 responses

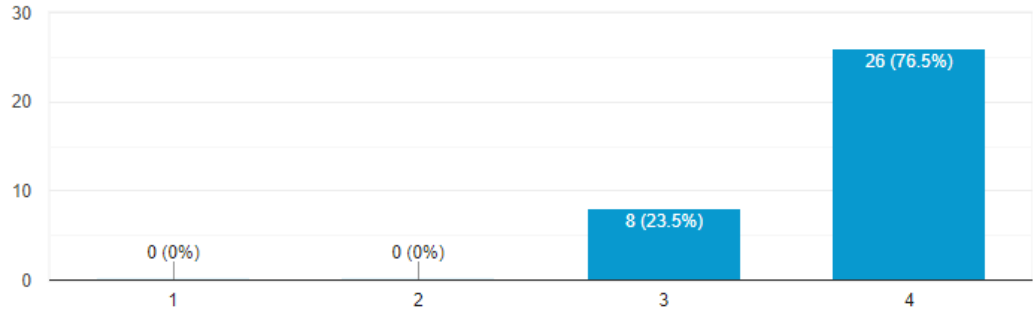


C. Reliability

8. The Syllagenius meets the needs for reliability under normal operation.

Copy

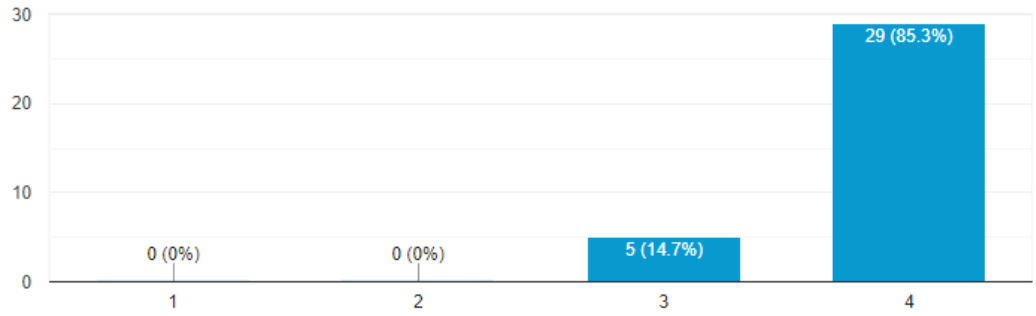
34 responses



9. The Syllagenius is operational and easily accessible when required for use.

Copy

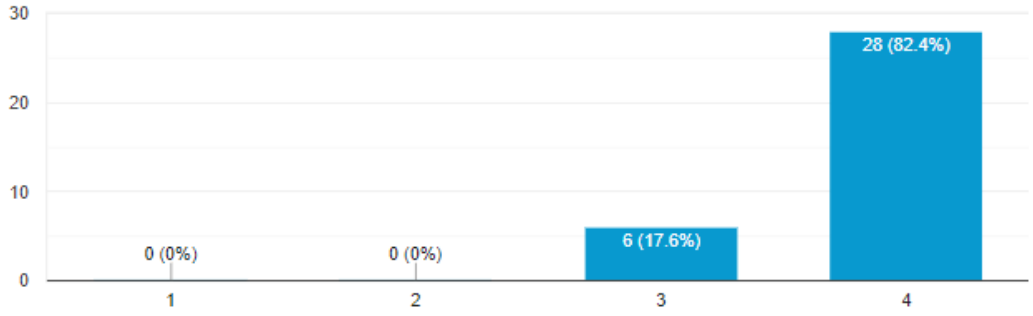
34 responses



10. The Syllagenius can recover the data directly affected and re-establish the desired state.

Copy

34 responses



Any Comments / Suggestions?

34 responses

None

N/A

None so far

Syllagenius is a good project.

The system presents reliability and is user-friendly. It can also be further enhanced for future research.

This is a good system, it could help the professors by using this for making syllabus.

n/a

The system is helpful for everyone

The actual concept was great. One thing I would suggest is the UI looks kinda simple so it's great if you will work on that to better improve the User Experience. Overall, the satisfactory rating was highly acceptable

Any Comments / Suggestions?

34 responses

add another action like make a PPT base on the syllabus for the upgrade of the system

This is for future improvements. When uploading ng supporting e-books, is it possible to upload more than one e-book? For example, I have specific topic in mind that requires additional information to be further understood. That's where I upload multiple e-books.

Creating a syllabus generator using natural language processing (NLP) could be incredibly beneficial for educators, trainers, and students alike. With such a tool, users could input parameters such as subject area, learning objectives, level of difficulty, and desired resources, and the AI could generate a comprehensive syllabus tailored to those specifications.

excellent application, much better to have a browser-based version

All goods

none

Its a good system, but make it more user friendly and more alive the user interface.

Any Comments / Suggestions?

34 responses

Make it automatic, where the user don't need to manually input the chapter into weeks that the user want to put it. Just let the user input chapter that they want to include into the syllabus and let the AI learn how to equally distribute the chapter/s per week. The AI has also the ability to predict on how many days the student can learn a chapter based on the words and difficulty of the subject or does it include computation or not. Again, this is for improvement to make an automated syllabus maker. Kudos for this project, this will be helpful especially for teachers.

minor revision lang sa UI at color palette

Highly Acceptable

The system was so amazing. I'm thinking about the professors or teachers that does not have access to e-book or doesn't know how to get or make an e-book? I suggest that there is a page where the topics can be manually inputted by the teachers like inputting the whole book topics and title of the book and it generate the syllabus for them. Also, you can add date from or ending date of semester to prescribe how many months are left before the Semester ends and what topics are possibly can be tackled within the timeframe. Again, the system was so informative and helpful.

Appendix D

EVALUATOR HIGHLIGHTS

1. **darrenflores001@gmail.com**

   Darren Reyes Flores

   IT Professional

   "The actual concept was great. One thing I would suggest is the UI looks kinda simple so it's great if you will work on that to better improve the User Experience. Overall, the satisfactory rating was highly acceptable."

2. **joshua.cervera@century-phirst.com**

   Joshua Cardaño Cervera

   IT Professional

   "Creating a syllabus generator using natural language processing (NLP) could be incredibly beneficial for educators, trainers, and students alike. With such a tool, users could input parameters such as subject area, learning objectives, level of difficulty, and desired resources, and the AI could generate a comprehensive syllabus tailored to those specifications."

3. **denzrevz26@gmail.com**

   College Student

   "The system was so amazing. I'm thinking about the professors or teachers that does not have access to e-book or doesn't know how to get or make an e-book? I suggest that there is a page where the topics can be manually inputted by the teachers like inputting the whole book topics and title of the book and it generate the syllabus for them. Also, you can add date from or ending date of semester to prescribe how many months are left before the Semester ends and what topics are possibly can be tackled within the timeframe. Again, the system was so informative and helpful."

4. **johnace_hingada@tup.edu.ph**

   John Ace Cosio Hingada

   College Professor

   "This is very exciting. However, teachers are irreplaceable in the context of students learning. Per se, syllabus is design for specific course objectives and not dependent on modules or e-books."

Appendix E

THESIS GRAMMARIAN CERTIFICATION

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-5301-3001 local 608| Fax No. +632-8521-4063 Email: cos@tup.edu.ph | Website: www.tup.edu.ph	Index No.	REF-COS-3.5-INT-TGC
		Revision No.	00
		Effectivity Date	06132022
VAA-COS	**THESIS GRAMMARIAN CERTIFICATION**	Page	1 / 1

THESIS GRAMMARIAN CERTIFICATION

This is to certify that the thesis entitled,

SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP

authored by

Espiritu, Elin Amalthea P.
Maguad, Kent Ivan R.
Salita, Mark Angelo I.
Maten, Khennedy Onnasis M.
Chua, Roswell Nathan I.

has undergone editing and proofreading by the undersigned.

This Certification is being issued upon the request of Juan A. Dela Cruz, Juanita B. Montefalco, Josefa C. Salazar, and Lorena D. Yu for whatever purposes it may serve them.

Prof. Marilyn M. Ignacio
Grammarian

Technological University of the Philippines

June 11, 2024
Date of Issuance

Transaction ID	
Signature	

Appendix F

CERTIFICATE OF SIMILARITY INDEX USING TURNITIN FROM URDS

	TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES Ayala Blvd., Ermita, Manila, 1000, Philippines Tel No. +632-5301-3001 local 711 | Fax No. +632-521-4063 Email: urds@tup.edu.ph | Website: www.tup.edu.ph	Index No.	REF-URD-INT-CSI
		Issue No.	01
		Revision No.	01
		Date	04132021
VRE-URD	CERTIFICATE OF SIMILARITY INDEX USING TURNITIN	Page	2 / 4
		QAC No.	CC-04132021

This is to certify that the manuscript entitled

"SYLLAGENIUS: AN AI GENERIC SYLLABUS GENERATOR USING NLP"

authored by

ELIN AMALTHEA P. ESPIRITU
KENT IVAN R. MAGUAD
KHENNEDY ONNASIS M. MATEN
MARK ANGELO L. SALITA
ROSWELL NATHAN I. CHUA

Has been subjected to similarity check on June 14, 2024 using Turnitin with generated similarity index of 13%.

Processed by:

DOROTHY R. PERNIS
Staff, URDS

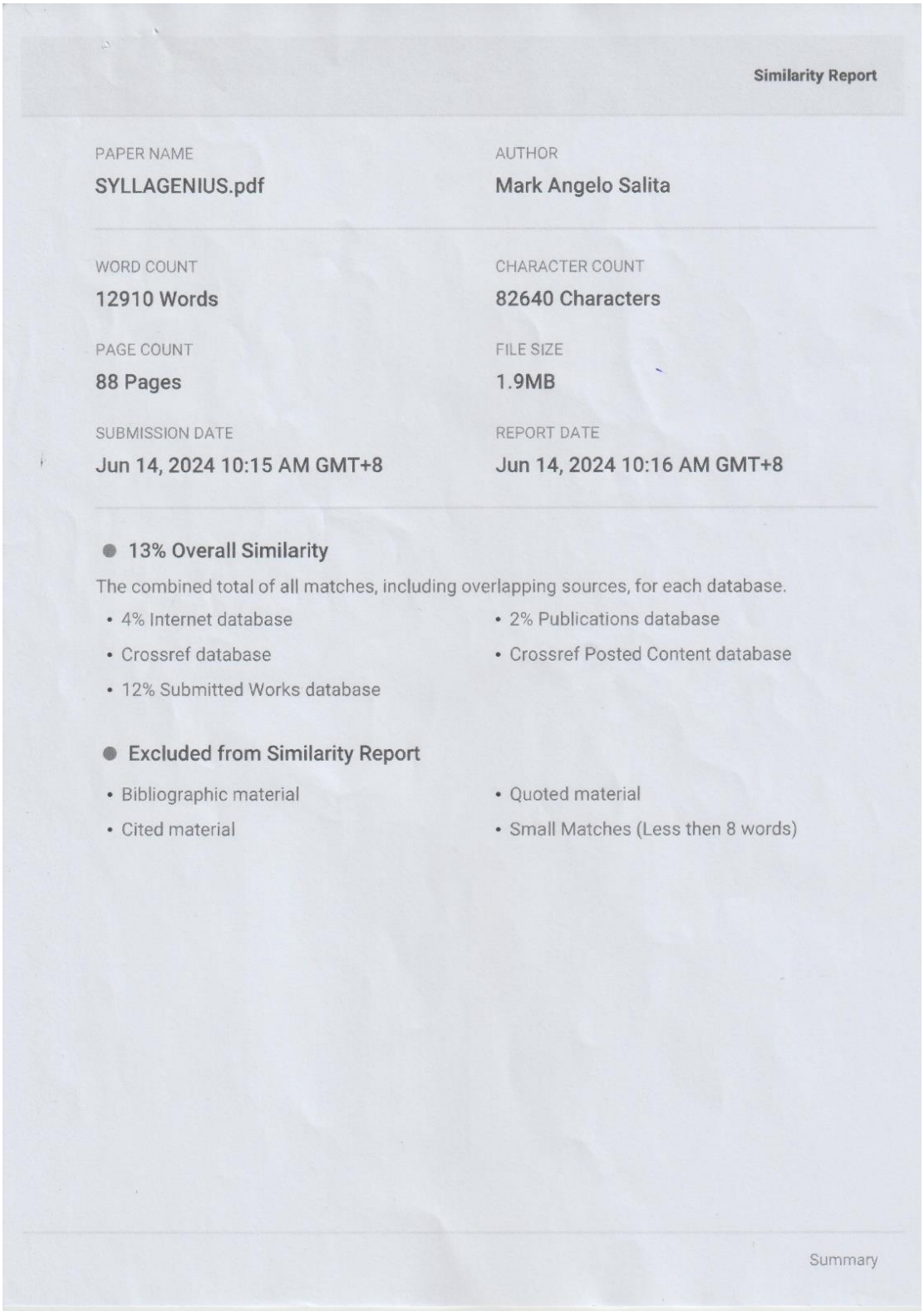
Certified correct by:

FRANCISCO D. ESPONILLA II, LPT, Ed.D.
Director, URDS

Transaction ID	
Signature	

Appendix G

CERTIFICATE OF SIMILARITY INDEX USING



Similarity Report

PAPER NAME
SYLLAGENIUS.pdf

AUTHOR
Mark Angelo Salita

WORD COUNT
12910 Words

CHARACTER COUNT
82640 Characters

PAGE COUNT
88 Pages

FILE SIZE
1.9MB

SUBMISSION DATE
Jun 14, 2024 10:15 AM GMT+8

REPORT DATE
Jun 14, 2024 10:16 AM GMT+8

● 13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 12% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 8 words)

Summary

REFERENCES

Abdul Ahad et al. (2023). Natural Language Processing Challenges and Issues: A Literature Review, 36(4)

Bisen, W., & Agrawal, A. (2022). A Review on Natural Language Generation. Bisen, WH, & Agrawal, AJ (2022). A review on natural language generation. *International Journal of Health Sciences*, p. 6, 10365–10376.

Englund, H., & Stockhult, H. (2023). Authority-Boundness as a Constitutive Aspect of syllabus. Boundness among Higher Education Students. *Scandinavian Journal of Educational Research, 67*(3), 406-418.

Fatima, N., Imran, A. S., Kastrati, Z., Daudpota, S. M., & Soomro, A. (2022). Systematic literature review on text generation using deep neural network models. IEEE Access, 10, 53490-53503.

Gatt, A., & Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research, 61, 65-170.

Harding, L. H. (2023). Supporting Diverse Learning Styles: A Case Study in Student Led Syllabus Design. *Journal of Political Science Education*, *19*(1), 83–90.

Junyi, L., Tianyi, T., Wayne, X. Z., Jian-Yun, N., & Ji-Rong, W. (2023). Pre-trained Language Models for Text Generation: A Survey.

Khani, D., Siddhartha, B. S., Niveditha, N. M., & Divya, B. M. (2021). An interpretation Of lemmatization and stemming in natural language processing. *Journal of University of Shanghai for Science and Technology*, *22*(10), 350–357.

Kurni, M, Mohammed M.S., Srinivasa K G, (2023). A Beginner's Guide to Introduce Artificial Intelligence in Teaching and Learning.

Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016, April). Cosine similarity to determine. similarity measure: Study case in online essay assessment. In 2016 4th International Conference on Cyber and IT Service Management (pp. 1-6). IEEE.

Lewis M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., (2019, October). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461v1 [cs.CL]

Lindgren, N.T. (2018). Recognizing names out of a string field – Sanitation of invoicing data.

Litman, D. (2016, March). Natural language processing for enhancing teaching and learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 30, No. 1).

Merchán Tamayo, J. P., Rocchi, M., Lennox Terrion, J., & Beaudry, S. (2022). First

Impressions Matter! An Experiment Comparing Autonomous and Controlling Language in Course Syllabi. *International Journal for the Scholarship of Teaching and Learning*, *16*(2), 7.

Mielke, S. J., Alyafeai, Z., Salesky, E., Raffel, C., Dey, M., Gallé, M., ... & Tan, S. (2021). Between words and characters: a brief history of open vocabulary modeling and tokenization in nlp. arXiv preprint arXiv:2112.10508.

Morrison, R. (2022). Large Language Models and Text Generators: An Overview for Educators. Online Submission.

Ojo, O. E., Alexander, G., Hiram, C., & Olaronke O. A., (2021, November). Performance Study of N-grams in the Analysis of Sentiments. 3(4):477-483

Plainer, M. (2021). Practical Study of Visual Studio Code.

Plisson, J., Lavrac, N., & Mladenic, D. (2004, October). A rule-based approach to word lemmatization. In Proceedings of IS (Vol. 3, pp. 83-86).

Rahim, F. R., Suherman, D. S., & Muttaqiin, A. (2020, March). Exploring the effectiveness of e-book for students on learning material: A literature review. In Journal of Physics: Conference Series (Vol. 1481, No. 1, p. 012105). IOP Publishing.

Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012, October). Semantic cosine similarity.

In The 7th international student conference on Advanced Science and Technology ICAST (Vol. 4, No. 1, p. 1).

Rouse, M. (2023, June 26). Natural Language Toolkit (NLTK). Techopedia. https://www.techopedia.com/definition/30343/natural-language-toolkit-nltk

Shin, S. (2014). E-book usability in educational technology classes: Teachers and teacher candidates' perception toward e-book for teaching and learning. *International Journal of Distance Education Technologies (IJDET), 12*(3), 62–74.

Wagner, J. L., Smith, K. J., Johnson, C., Hilaire, M. L., & Medina, M. S. (2023). Best practices in syllabus design. *American Journal of Pharmaceutical Education, 87*(3).

Wotring, A., Chen, H., & Fraser, M. (2021). Exploring Curriculum Alignment through Syllabus Document Analysis: From National Language Policy to Local ELT Practice. *Iranian Journal of Language Teaching Research*, *9*(2), 57–72.

Yamauchi, T., Flanagan, B., Nakamoto, R., Dai, Y., Takami, K., & Ogata, H. (2023). Automated labeling of PDF mathematical exercises with word N-grams VSM classification. Smart Learning Environments, 10(1), 51.

Yogish, D., Manjunath, T. N., & Hegadi, R. S. (2019). Review on natural language processing trends and techniques using NLTK. In Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R

2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part III 2 (pp. 589-606). Springer Singapore.

Željko J., Mihailo K., Uroš P., Slađana Đ. (2022). Determining source code repetitiveness on various types of programming assignments.

RESEARCHERS' PROFILE

KENT IVAN R. MAGUAD

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

PERSONAL & CONTACT INFORMATION

Address: 17# Fatima Ville, Road 15, Quezon City
Contact Number: 0921-288-6560
E-mail: kentivan.maguad@tup.edu.ph
Gender: Male
Age: 23
Date of Birth: December 24, 2000

EDUCATIONAL ATTAINMENT

Level	School	Year
Tertiary	Technological University of the Philippines – Manila	2020 - 2024
Secondary	STI College Muñoz-EDSA	2018 – 2020
	Ismael Mathay Sr. Highschool	2015 – 2018
	San Francisco High School	2011 – 2014
Primary	Sto Cristo Elementary School	2010 – 2011
	Child's Future Guided Academe, Inc.	2006 – 2010

KNOWLEDGE AND SKILLS

- Web Designing
- Video Editing
- Sound Editing
- Front End Development
- Communication
- Javascript, python, C#
- Back-end Development
- App Development
- Committed
- Problem-solving
- Html, css

ELIN AMALTHEA P. ESPIRITU

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

PERSONAL & CONTACT INFORMATION

Address: 137 Road 3, Project 6, Quezon City
Contact Number: 0945-471-6064
E-mail: elinamalthea23@gmail.com
Gender: Female
Age: 22
Date of Birth: April 23, 2002

EDUCATIONAL ATTAINMENT

Level	School	Year
Tertiary	Technological University of the Philippines – Manila	2020 - 2024
Secondary	STI College Muñoz-EDSA	2018 – 2020
	Ernesto Rondon High School	2016 – 2018
	Caybiga High School	2015 – 2016

KNOWLEDGE AND SKILLS

- HTML5
- CSS3
- Javascript
- Python
- C/C++
- Canva
- C#
- Java
- VB.Net
- SQL
- Figma
- Basic Photoshop

KHENNEDY ONNASIS M. MATEN

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

PERSONAL & CONTACT INFORMATION

Address: 1361 Musa Street, Barangay 516, Sampaloc, Manila, Metro Manila
Contact Number: 0995-892-5534
E-mail: khnmaten13@gmail.com
Gender: Male
Age: 23
Date of Birth: December 13, 2000

EDUCATIONAL ATTAINMENT

Level	School	Year
Tertiary	Technological University of the Philippines – Manila	2020 - 2024
Secondary	Santa Cruz Academy, Inc	2017 – 2020
	Maabud National High School	2014 – 2017
Primary	Cubamba-Gahol Elementary School	2013 – 2014
	Malabago Elementary School	2012 – 2013
	Don Marcelo C. Marty Elementary School	2008– 2012
	St. Michael School	2006-2008

KNOWLEDGE AND SKILLS

- UI/UX Designer
- Proficient in Javascript, Python, C/C++ Language
- Graphics Designer
- 2D CAD Drafting
- Communication Skills (Verbal & Written)
- Creative and Innovative
- Teamwork & Conflict Resolution
- Resourceful

MARK ANGELO L. SALITA

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

PERSONAL & CONTACT INFORMATION

Address: #37 Kalinisan St. Batasan Hills Q.C.
Contact Number: 0915-316-9930
E-mail: Markangelo.salita@tup.edu.ph
Gender: Male
Age: 21
Date of Birth: April 18, 2002

EDUCATIONAL ATTAINMENT

Level	School	Year
Tertiary	Technological University of the Philippines – Manila	2020 - Present
Secondary	Batasan Hills National High School	2013 – 2020
Primary	Adonai Shepherd Academy	2005 – 2013

KNOWLEDGE AND SKILLS

- Committed
- HTML, CSS, and JavaScript.
- C, Java, Python, C++, and/or others
- Software design, development, testing, and debugging.
- MySQL
- Networking
- Analyze problems
- Manage time efficiently

ROSWELL NATHAN I. CHUA

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

PERSONAL & CONTACT INFORMATION

Address: #1-C Premium St. GSIS Village Project 8 Q.C
Contact Number: 0947-049-8728
E-mail: Roswellnathan.chua@tup.edu.ph
Gender: Male
Age: 22
Date of Birth: December 27, 2001

EDUCATIONAL ATTAINMENT

Level	School	Year
Tertiary	Technological University of the Philippines – Manila	2020 - 2024
Secondary	STI College Muñoz-EDSA	2018 – 2020
	Ismael Mathay Sr. Highschool	2014 – 2018
	Elyseum Christian School	2013 – 2014
Primary	First line Integrated School	2008 – 2013
	GSIS Village Elementary School	2007 – 2008
	Good Samaritan Christian School	2006 – 2007

KNOWLEDGE AND SKILLS

- Committed
- Adaptive
- Communicative
- Problem-solver
- Attentive
- Hardworking
- App-Development