

# **SNIPLT: Bite-Sized Learning , Big Impact...**

**Kajal , Divya Chauhan, Priyanshu Pal, Tamanna Dubey, Ketan Singh**

Department of Computer Science & Engineering

Moradabad Institute of Technology, Moradabad, India

[kajalashok10@gmail.com](mailto:kajalashok10@gmail.com)

[chauhandivya2305@gmail.com](mailto:chauhandivya2305@gmail.com)

[priyanshupal1212@gmail.com](mailto:priyanshupal1212@gmail.com)

[tamannadubey71@gmail.com](mailto:tamannadubey71@gmail.com)

[Ketan9949@gmail.com](mailto:Ketan9949@gmail.com)

## **ABSTRACT**

The rise in digitized knowledge resources has speedily altered the landscape under which current learning or knowledge acquisition takes place. This has allowed students to readily access enormous amounts of knowledge. Nevertheless, there has also been an intensification of mental efforts for students who then have to digest a massive volume of study content within a restricted timeframe. The conventional techniques of revision, entailing physical reading and note-taking, are no longer adequate for effectively handling large PDF files, more so in higher education due to dense and lengthy knowledge texts.

In this article, the author has introduced a web-based intelligent learning system called “SNIPLT (Smart Neural Interactive Platform for Learning and Testing)”. This system has a function to automate document summarization and self-assessment. In this system, users have to upload a PDF document with a size limit of up to 30MB. Then, a maximum of 3000 words are taken from the uploaded PDF document. With the help of abstractive summarization models trained with transformers, SNIPLT generates a summarized form with a total of about 1000 words.

Apart from the summarization task, the system automatically provides 20–25 multiple-choice questions with the answer choices formulated from the summarization output. The system uses a seed mechanisms approach for the randomization of questions in order to have a different set of questions in each attempt using the same document. For enhanced accessibility and user engagement, the SNIPLT system uses a voice interpreter module. The voice interpreter module is responsible for narrating the questions using the text-to-speech approach. Also, it uses the recognition method for the user response. Apart from the voice module functionality, the SNIPLT system uses an explanation module. The module is responsible for explaining the reason behind the correct answer. Additionally, the SNIPLT system uses a translation module. The module is responsible for translating English quiz questions to Hindi.

The experimental evaluation carried out among the student users has shown the effectiveness of SNIPLT in improving efficiency, time required for revisions, and student engagement through interaction. The experiments have shown the proposed system as an applicable solution in intelligent education through digital platforms.

**Keywords**— Natural Language Processing, Abstractive Summarization, Automatic Quiz Generation, Speech Recognition, Educational Technology.

## 1. INTRODUCTION

The ever-increasing digital evolution in the field of education has significantly changed the manner in which students access and consume educational materials. Educational materials are now mostly digital in nature and found in the form of Portable Document Format (PDF), which is commonly used for the distribution of textbooks, research materials, studying notes, and examination documents. Although the digital evolution has improved the accessibility of educational materials, a major challenge has arisen in the manner in which students are required to digest a large quantity of material within a limited timeframe.

Students often have to read and edit large PDF files comprising hundreds of pages of information. Such large amounts of information create problems of cognitive overload, poor understanding, and ineffective learning. Conventional learning solutions like manual text marking, writing learning points, or reading multiple documents several times do not serve the purpose of coping with the increased complexity of learning and vast amounts of learning content. Due to this, students face difficulties in learning critical learning points, learning, and retaining learning outcomes for exams.

Artificial Intelligence (AI) has been developed as a robust method to overcome these challenges by developing capabilities in machines to understand, interpret, and produce human language. Natural Language Processing (NLP), as a subset of AI, holds procedures and methods related to automatic text summarization, information retrieval, question formation, and semantic analysis. Such capabilities enable the development of intelligent systems capable of converting unorganized educational content into organized learning materials.

Smart Neural Interactive Platform for Learning and Testing or SNIPLT would be an effective proposed system that could combine multi-functional support systems for learning in one web-based system. The proposed system would help students by summarizing academic PDFs of lengthy study materials to small summaries utilizing abstractive summary models consisting of transformers in linguistic models. Additionally, it would also produce interactive quizzes in MCQ form that could help students learn readings for assessments after revisions.

Apart from the text-based interface, SNIPLT includes speech and translation modules to increase accessibility and inclusiveness. The speech interpreter enables a learner to hear a question and respond orally. This is ideal for visionarily challenged and audio learners. The translation module is used to translate English quizzes into Hindi. This is for learners who come from regions where Hindi is their primary language and might not be able to use English educational platforms effectively.

## 2. LITERATURE REVIEW

Text summarization remains an active field of research in Natural Language Processing for several decades. In earlier research works, it was more of an extractive process where key sentences are chosen from the document by using statistical properties like term frequency, sentence number, cue phrases, and word dispersion. Nallapati et al. [1] presented one of the first works in neural abstractive summaries using sequence to sequence recurrent neural network architectures to show that neural methods are superior to traditional extractive methods.

Rush et al. proposed neural attention models for summarizing sentences to produce syntactically well-formed and semantically meaningful sentences [2]. However, with recent developments in deep learning models, transformer models further enhanced this area. Lewis et al. proposed the BART model for pre-training sequence-to-sequence models as a denoising model, whereas Raffel et al. proposed the T5 model that incorporated various NLP models in one model for different tasks in the text-to-text format [3], [4], that further enhanced state-of-the-art results for abstractive summarization techniques to capture global contextual dependencies in texts through self-attention mechanisms [12].

Automatic Question Generation (AQG) is another area of active research. Kumar et al. [5] built an automated system where questions were generated by applying natural language processing techniques, and Mazidi and Tarau [6] further improved the question generation process by applying syntactic parsing and semantics role labeling techniques to produce semantically well-formed questions. However, despite such breakthroughs in AQG systems, they still face challenges in domain adaptation.

Speaking interaction between human and computer has seen much advancement as a result of research in Deep Recurrent Neural Networks. This is clear in the study of Graves et al. [13] who proved that using Deep Neural Networks in building systems for speech recognition outperformed HMM methods [23]. Glass et al. [7] continued to research how SD systems enhanced interactive applications by proving that voice interaction aids user engagement in learning contexts.

BERT [9], Word2Vec [10], and language representation models have proved to be an invaluable addition to NLP to develop a better semantic understanding in language models. These language models have made syntactic and semantic relationships in text much more efficient to handle in NLP technology tools such as spaCy [18], NLTK [27], which made text processing and feature extraction in NLP applications on a large-scale feasible.

Text Rank algorithm for ranking the variables of the graph was proposed in the context of extractive summary generation [25], further elaborated in the context of text summary generation in [11]. Although very useful for extractive summary generation, the ranked list of summary variables using these models lacks semantic consistency as compared to the abstractive summary models.

Evaluation tools like ROUGE [16] are widely accepted standards for summarization quality, and the advent of Neural Machine Translation has made it feasible for many languages to be supported within learning systems [19]. Platforms like Hugging Face Transformers [22], Flask [21], and MongoDB databases [20] make it easy for learning systems and NLP-enabled applications to be implemented on the web.

Although substantial works have been carried out in the areas of summarization, quiz generation, speech recognition, and machine translation separately, few systems have been developed to encompass all of them in one learning platform. The SNIPLT system fills this research gap in that it brings together automated summarization, interactive quiz generation, voice-based interaction, and multilingual translation in one intelligent web-based platform..

### **3. METHODOLOGY**

The methodology employed for creating SNIPLT (Smart Neural Interactive Platform for Learning and Testing) is intended to guarantee precision, scaling, reliability, and usability in a practical learning setting. Accordingly, a comprehensive process for creating an SLP is segmented into five consecutive stages: data acquisition, text processing for extracting and cleaning the text, abstractive summary generation, creation of a quiz, and interacting with the user through interpreter and translator systems.

#### **3.1 Data Acquisition**

Only authenticated users will be able to upload PDF files. If the user wants to upload the PDF files, then only will the files be permitted. The size for each of the PDF files will be 30 MB, which will enable the learner to examine a large quantity of learning materials such as reference books, tutorial guides, and other writings. Once the files have been uploaded, the data from the files will be stored temporarily in a secured directory on the server, including the file name, time, and user ID.

#### **3.2 Text Extraction And Preprocessing**

PyMuPDF acts in extracting text page by page from a PDF document to get raw textual data. Usually, the extracted text is noisy, containing many formatting symbols, line breaks, and spacing that is not on pattern. A simple preprocessing pipeline that would clean non-textual characters, redundant whitespace, headers, footers, and page numbers was applied.

Considering the requirement for balancing performance and coverage of information, only the first 3000 words are retained for further processing. One has to be sure that the computational time is viable while maintaining the original essence of the document. Then follow the steps of cleaning of text, tokenization of words, tokenization of sentence, stop word removal, segmentation of sentences, summarization preparation, and generating questions.

#### **3.3 Abstractive Summarization**

This processed text is then fed to an abstractive summarizer developed using the BART transformer model. BART is a transformer-based model that is fine-tuned to perform long-form text summarization. It is capable of capturing relationships within the text through self-attention. This is because it converts the input text into an embedded form.

To restrict the size of the summary, it is ensured that only summaries of approximately 1000 words are produced by adding a minimum and maximum token restriction to control its size.

This is to ensure that only key notions of the input are considered in the output and no redundancy or irrelevant information exists in it.

### 3.4 Quiz Generation

The overall summary content is broken down into individual sentences, and the key terms are extracted using NER techniques and keyword extraction. For each chosen sentence, it will be turned into a question by replacing some key entities or phrases with blanks. For distractor options, a mix of synonym replacement, semantic similarity ranking, and random sampling of domain-specific vocabularies is employed.

A seed-based randomization technique has been implemented that will vary the quizzes: different seed values give the system the capability to generate a set of 20–25 multiple-choice questions even from the same PDF.

### 3.5 Interpreter And Translator Integration

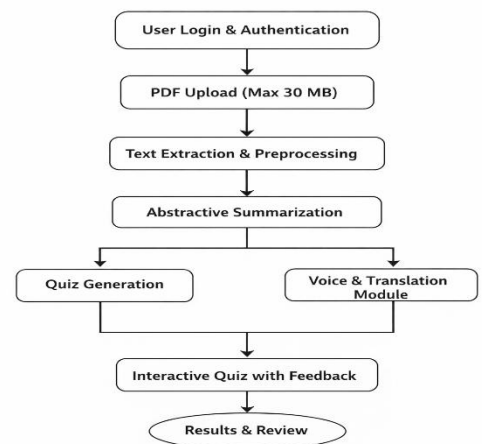
The interpreter module is an addition to user interactions, allowing for voice functionality. This is done using Text-To-Speech Technology by “pyttsx3” for reading out quiz questions, and “SpeechRecognition” for recognizing speech given by the user. The interpreter module is an important addition for the visually impaired and for those who learn by listening.

The translator module is based on neural machine translation capabilities to translate English quiz content to Hindi. This is to ensure inclusivity for users from different linguistic backgrounds. Notably, the explanation sub-module is responsible for fetching the original sentence from the summary that is aligned to the correct answer and provides an explanation.

## 4. SYSTEM OVERVIEW

SNIPLT consists of the following core components:

1. User Authentication Module
2. PDF Processing Module
3. Summarization Engine
4. Quiz Generation Module
5. Interpreter Module
6. Translator Module



Each module is loosely coupled, ensuring scalability and modularity.

### 4.1 User Authentication

SNIPLT employs a secured JSON Web Token authentication system for managing user access.

The passwords that are given to each user or customer are encrypted using hashing algorithms and stored in the database.

### 4.2 Pdf Processing Module

PDF files are processed using the library PyMuPDF. Pages are processed to fetch the words. But to maintain the performance speed, the first 3000 words are considered. This helps to process the data quickly and cover adequate academic content.

### 4.3 Summarization Engine

The summarization algorithm employs the BART transformer model, which has been pre-trained on “CNN/DailyMail” datasets. The text goes through the process of “byte pair encoding” for encoding. The BART transformer produces abstractive summaries while controlling length restrictions between 700-1000 words.

### 4.4 Quiz Generation Module

The compressed information is broken down into smaller units called sentences. Named Entity Recognition and key word extraction are employed to mark significant key words. The questions are formulated employing sentence transformation procedures, with the distractor choices formulated through the application of synonym and semantic similarity.

Seed-based randomization is used to ensure that in repeated attempts different sets of questions are produced using the identical PDF.

### 4.5 Interpreter Module

This module increases engagement through Text-to-Speech narration of quiz questions using pyttsx3 and Speech Recognition to capture user responses. It also includes an explanation submodule that gives more contextual reasoning for correct answers.

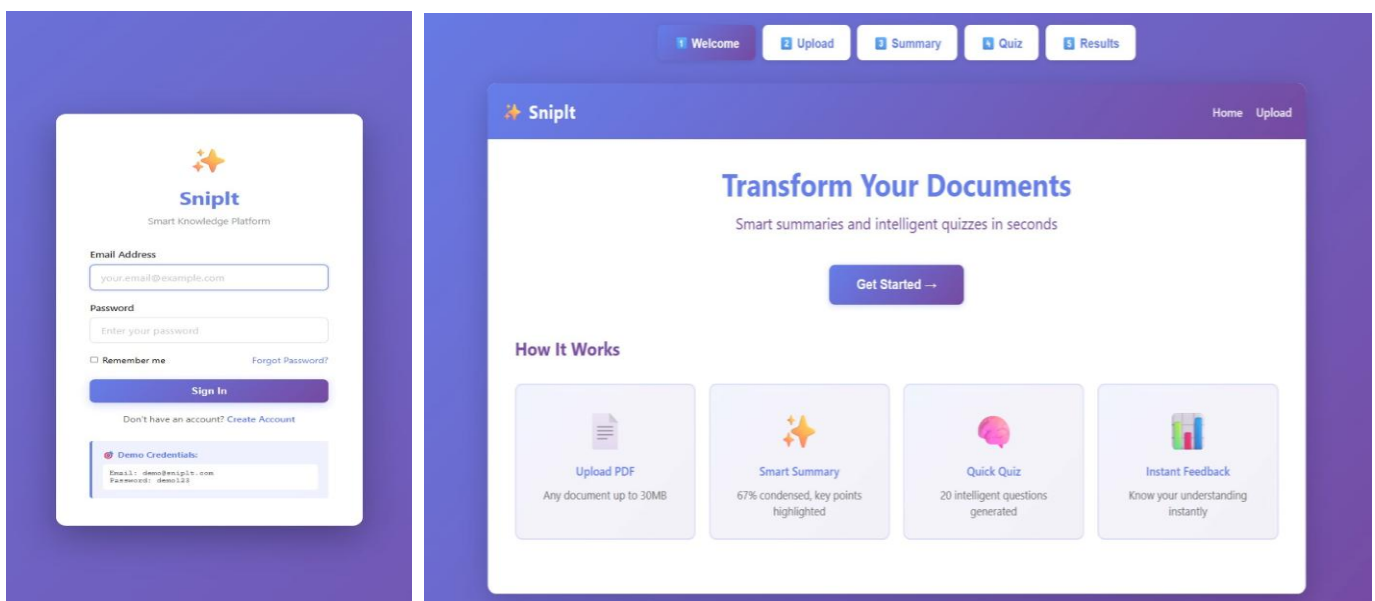
### 4.6 Translator Module

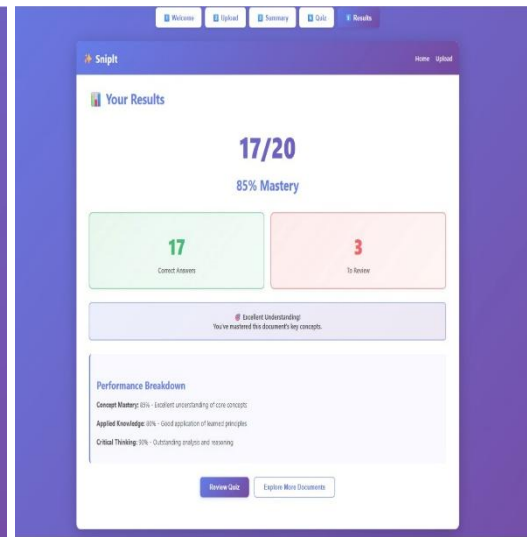
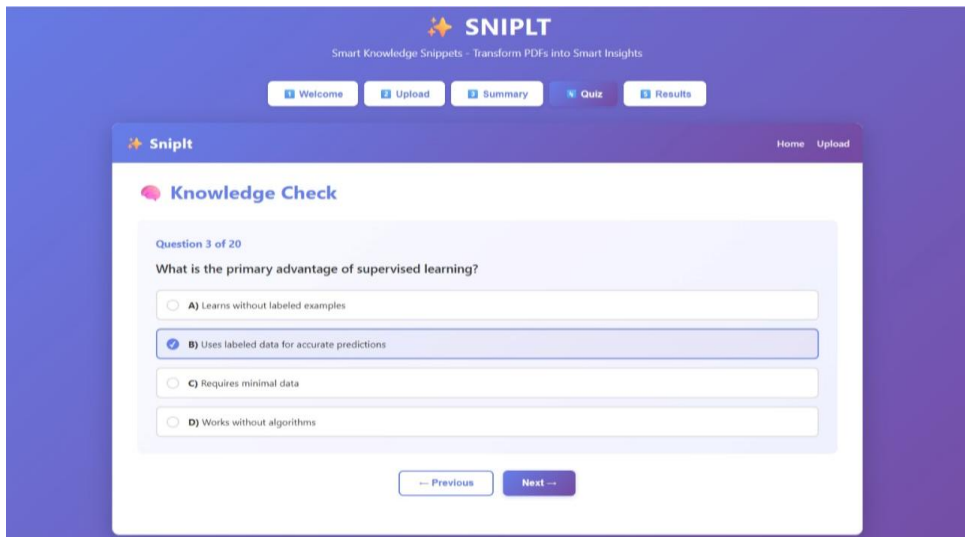
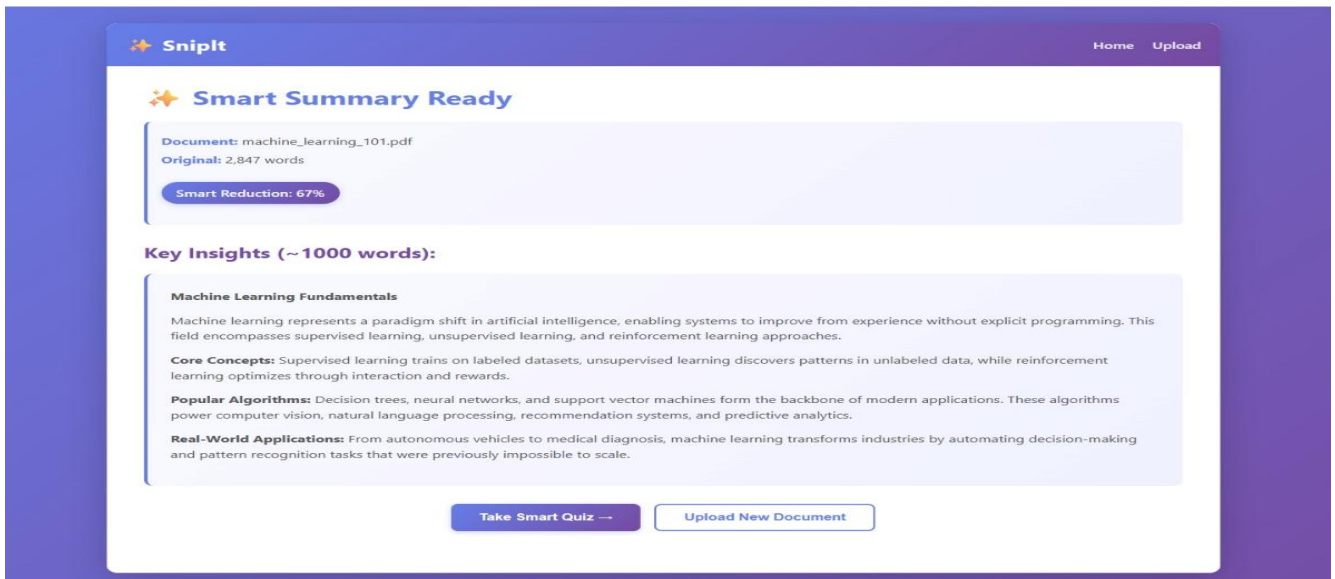
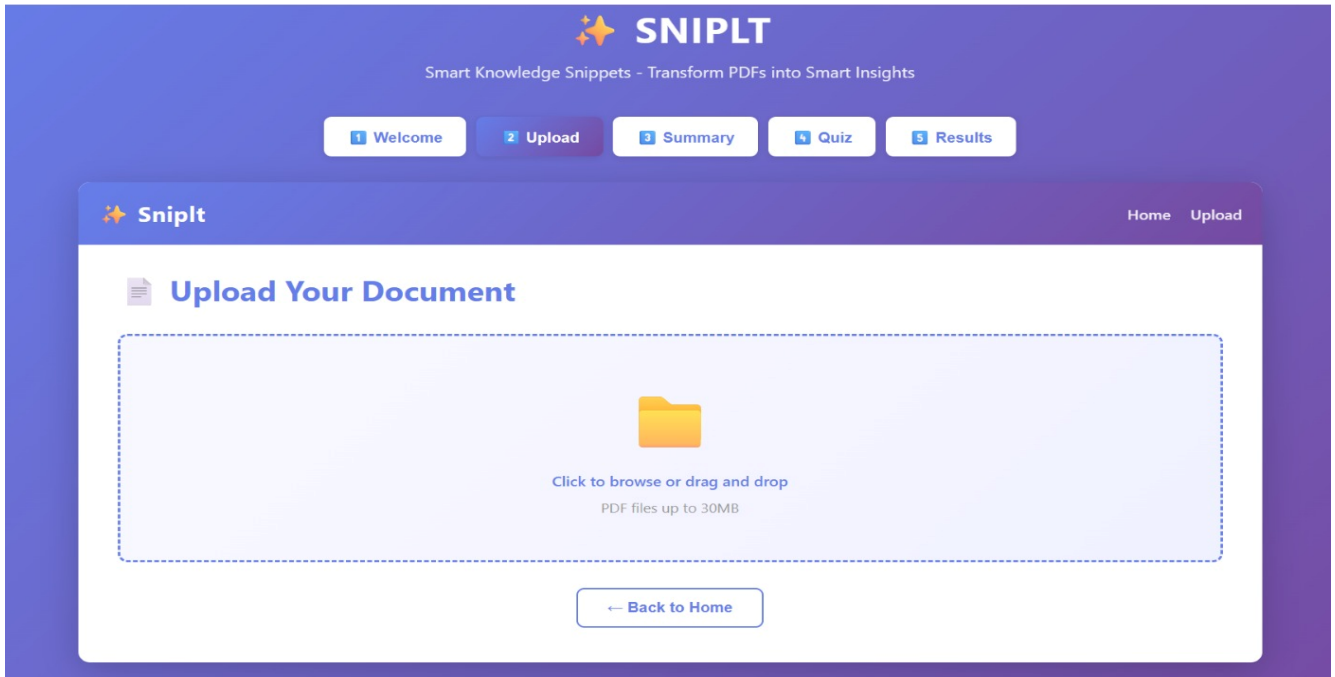
The Deep-Translator API translates the quiz question from English to Hindi so that students from regional backgrounds are also able to take part in the quiz without any hassle..

### 4.7 Implementation

The backend is done using Flask, and the data is stored in MongoDB. REST APIs are applied in handling interactions between the frontend and backend. The frontend is developed using responsive HTML, CSS, and JavaScript.

## 5. RESULTS





## 6. CONCLUSION AND FUTURE WORK

In this study, the researcher introduced SNIPLT (Smart Neural Interactive Platform for Learning and Testing), an intelligent Web platform. SNIPLT was conceptualized and developed with the aim of addressing the two major challenges posed by the digital era in the field of higher education. These two challenges are the issue of ‘information overload’ and the challenge of ‘inefficient revision.’ SNIPLT uses advanced natural language processing concepts. SNIPLT has the capability to summarize automatically from PDF documents consisting of 3000 words. This application will be able to reduce the workload of students in turning 3000 words of raw data into a summarized version of 1000 words.

Along with summarization and evaluation, SNIPLT integrates other accessibility-oriented functionalities, such as speech interaction and language support. The speech interpreter component helps learners listen to the questions and respond orally, hence facilitating auditory learners and the visually impaired. The explanation component enhances understanding of concepts by providing a satisfactory explanation for the answers given. This will ensure that students comprehend, rather than memorizing. In addition to that, the students will benefit from the feature that will allow the translation of English to Hindi since the students will be able to understand and navigate the system easily.

By experimental analysis and user feedback, it has been concluded that SNIPLT increases the efficiency of learning by reducing the revisiting time, actively engaging the learners, and increasing the levels of comprehension. Moreover, its broad modular design makes it scalable and flexible too, which would play an important part in future upgrades.

In the future, the work will be done on extending SNIPLT in various ways Adaptive difficulty levels, which will adapt the complexity level of quizzes based on the usage patterns of the users, is one such. This will enable personalized learning paths to be created. Other major areas would be its multilinguality, adding more regional and international languages, and then its compatibility with LMS systems such as Moodle and Google Classroom. This will enable it to be seamlessly integrated into a learning environment. The research also looks to Domain-specific fine-tuning of summarization and question-generation models for better accuracy in technical and specialized domains.

Overall, SNIPLT demonstrates the potential of AI-driven educational tools to transform traditional learning methods into intelligent, interactive, and inclusive digital learning experiences.

## REFERENCES

- [1] R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, “Abstractive text summarization using sequence-to-sequence recurrent neural networks and beyond,” in Proc. 20th Conf. Empirical Methods in Natural Language Processing (EMNLP), Austin, TX, USA, 2016
- [2] A. M. Rush, S. Chopra, & J. Weston, "A neural attention model for abstractive sentence summarization," in Proc. Conf. on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, 2015, pp. 379-389

- [3] M. Lewis, Y. Liu, N. Goyal, et al., “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in Proc. 58th. Meeting ACL, Online, 2020,
- [4] C. Raffel, N. Shazeer, A. Roberts, et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020
- [5] V. Kumar, S. Jain, and M. Dutta, “Automatic question generation using natural language processing,” *International Journal of Artificial Intelligence*, vol. 15, no. 2, pp. 45–55, 2017.
- [6] S. Mazidi and R. Tarau, “Infusing NLP into automatic question generation,” in *Proc. ACL Workshop on Natural Language Processing for Education*, Berlin, Germany, 2016, pp. 51–60.
- [7] J. Glass, H. Zen, and C. S. Lee, “Spoken dialogue systems,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 116–131, Nov.
- [8] D. Jurafsky and J. H. Martin. *Speech and language processing*. 3rd ed. Pearson, 2019.
- [9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp.
- [10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. “Distributed representations of words and phrases.” Proc. Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 2013, 3111 –
- [11] H. Zhang, Y. Gong, and C. Liu, "Text Rank: Bringing order into text," *IEEE Intelligent Systems*, vol. 19, no. 3, pp. 23–29,
- [12] A. Vaswani, et al., “Attention is all you need,” Proc. NIPS, Long Beach, CA, USA, 2017, pp. 5998
- [13] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. ICASSP, Vancouver, BC, Canada, 2013, pp. 6645
- [14] Google AI. “Speech-to-Text API Documentation.” 202
- [15] S. Bird, E. Klein, and E. Loper, \*Natural Language Processing with Python\*, Sebastopol, CA, USA: O’Re
- [16] K. Papineni, S. Roukos, T. Ward, & W. Zhu, “ROUGE: A package for automatic evaluation of summaries,” in Proc. ACL Workshop, Philadelphia, PA, USA, 2002, pp. 74-
- [17] J. Allen. *Natural Language Understanding*. Pearson. Redwood City, CA, USA. 1995
- [18] M. Honnibal and I. Montani, “spaCy: Industrial-strength natural language processing in Python,”.
- [19] P. Resnik and J. Lin, “Evaluation of multilingual machine translation systems,” *Computational Linguistics*, vol. 36, no. 2, pp. 151–177, 2010.
- [20] MongoDB Inc., “MongoDB Documentation,” 2022. Available online: <https://www.mongodb.com/docs>

- [21] Flask Community, “Flask Web Framework Documentation,” 2021.
- [22] Hugging Face, “Transformers Library,” 2022. [Online]. Available: <https://huggingface.co/transformers>
- [23] L. Rabiner, “A tutorial on hidden Markov models and selected applications,” IEEE Proc., vol. 77, no. 2, pp. 257–286, 1989.
- [24] Y. Goldberg, Neural Network Methods in Natural Language Processing. Morgan & Claypool, 2017.
- [25] R. Mihalcea and P. Tarau, “Text Rank: Bringing order into texts,” IEEE Intelligent Systems, vol. 19, no. 3, pp. 23–29, 2004.
- [26] D. Jurafsky, “The language of food,” Cultural Analytics, vol. 1, no. 1, 2014.
- [27] E. Loper and S. Bird, “NLTK: The Natural Language Toolkit,” in Proc. ACL Workshop on Effective Tools and Methodologies, Philadelphia, PA, USA, 2002, pp. 63–70.