www.ijprems.com
editor@ijprems.com

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 11, November 2025, pp : 229-233

e-ISSN : 2583-1062
Impact Factor : 7.001

AI NOTES SUMMARIZER: A PRIVACY-FIRST, OFFLINE SYSTEM FOR AUTOMATED TRANSCRIPTION AND SUMMARIZATION OF RECORDED LECTURES

Tanmay Arora[1], Ms. Meenu[2]

[1]Scholar, B.tech (AI&DS) 4th Year Department of Artificial Intelligence and Data Science Dr. Akhilesh Das Gupta Institute of Professional Studies, New Delhi, India.

[2]Assistant Professor (AI&DS), Department of Artificial Intelligence and Data Science Dr. Akhilesh Das Gupta Institute of Professional Studies, New Delhi, India.

DOI: https://www.doi.org/10.58257/IJPREMS44541

ABSTRACT

The proliferation of hybrid learning and online courses has made recorded lectures a central component of modern education. However, students face significant challenges in managing and processing these vast amounts of audio content, often struggling to extract key information efficiently without replaying material multiple times. Existing solutions, such as commercial transcription services, are predominantly cloud-based, raising concerns about data privacy and requiring constant internet access. This paper proposes an AI-powered, completely offline system designed to address these challenges. The system integrates a robust speech-to-text (STT) module, leveraging locally-run models like OpenAI Whisper or Vosk, with an abstractive text summarization module using transformer-based architectures such as T5 or BART. The core innovation is a unified, privacy-first pipeline that accurately transcribes lecture audio, generates concise and coherent summaries, and extracts key concepts—all locally on the user's machine. This approach ensures data remains private and provides a reliable tool for students and educators in low-connectivity environments. The system's effectiveness is evaluated using both quantitative metrics like ROUGE and BLEU scores and qualitative user feedback.

Keywords: Speech-to-Text (STT), Abstractive Summarization, Offline AI, Natural Language Processing (NLP), Educational Technology, Privacy-Preserving AI, Whisper, T5.

1. INTRODUCTION

The past decade has seen a significant transformation in how students access learning materials. The widespread availability of recorded lectures, online courses, and hybrid learning environments has provided unprecedented flexibility for learners. While this shift allows for valuable self-paced study, it has also introduced the critical problem of information overload. Students often find themselves overwhelmed by the sheer length of recordings, struggling to extract essential points without replaying the material multiple times.

This inefficiency can significantly hinder the revision process, particularly during examination preparation or project work [6]. Traditionally, students take handwritten or typed notes during live lectures, a process that is already cognitively demanding, requiring active concentration and quick summarization skills. When reviewing recorded lectures, this note-taking process becomes even more time-consuming, involving multiple pauses and rewinds to ensure no crucial detail is missed.

The emergence of Artificial Intelligence (AI) and Natural Language Processing (NLP) offers a potent solution to this problem.[1] By combining speech-to-text (STT) conversion with automatic text summarization, it is possible to generate concise, structured notes directly from recorded audio. This project proposes a system to do just that, with a critical distinction: it operates completely offline.

1.1 Challenges

Developing an integrated, offline summarization tool presents several distinct challenges that this project aims to address:

- **Cloud Dependency and Privacy:** Most high-accuracy STT and summarization tools (e.g., Otter.ai, Sonix, Fireflies) are cloud-dependent API services. This raises significant data privacy concerns, as user data must be uploaded and processed on remote servers.
- **Accessibility and Connectivity:** The reliance on cloud services creates a barrier for users without reliable, high-speed internet, particularly students in rural or remote areas.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
www.ijprems.com
editor@ijprems.com
Vol. 05, Issue 11, November 2025, pp : 229-233
e-ISSN : 2583-1062
Impact Factor : 7.001

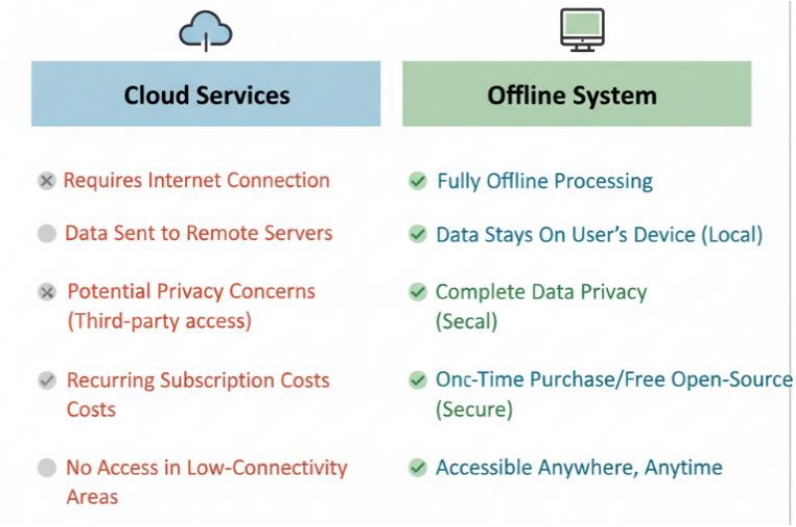
- **Computational Resources:** High-performance offline models, such as the larger versions of OpenAI's Whisper, require significant computational resources (GPU, RAM), making them difficult to run on standard student laptops. A balance must be struck between accuracy and local processing feasibility.
- **Summarization Quality:** Early summarization techniques were extractive, merely identifying and copying key sentences. While simple, this often results in disjointed and incoherent summaries. Modern abstractive models (like T5 or BART) generate new, human-like summaries but are more complex to implement and run offline.
- **Pipeline Integration:** Most academic prototypes focus on either STT or summarization in isolation. Few systems integrate both into a single, user-friendly, offline pipeline specifically designed for educational content.

1.2 Need for an offline

The necessity for an offline-first architecture is driven by two primary factors:

Privacy and Data Security: Lecture content can be sensitive. By processing all data locally on the user's machine, the system ensures that audio files and transcripts never leave the user's device. This prevents data leakage, avoids privacy concerns associated with cloud services, and makes the system compliant with strict institutional data policies.

Accessibility and Equity: A reliance on cloud services inherently excludes users in rural or remote areas with poor or no internet connectivity. A fully offline tool ensures that any student with a computer can access the tool's benefits, regardless of their internet quality. It also avoids the recurring subscription costs of commercial tools.



1.3 Applications

- **Student Revision:** Allows students to quickly review key concepts from hours of lectures, significantly reducing the time and effort required for exam preparation.
- **Educator Assistance:** Enables teachers and educators to generate summaries and keyword lists from their own lectures to provide as supplementary revision material.
- **Accessibility:** Assists students with hearing impairments or learning disabilities by providing accurate transcripts and condensed notes.
- **Professional Development:** The core framework can be adapted for summarizing corporate meetings, training sessions, or professional seminars.

2. LITERATURE REVIEW

This project builds upon established research in two key areas: Speech-to-Text (STT) technologies and Text Summarization approaches [4].

2.1 Speech-to-Text (STT) Technologies

STT technology has seen significant advances. Cloud-based services like Google Speech-to-Text and Amazon Transcribe are known for high accuracy rates in English speech recognition. However, their primary limitation is their complete dependence on cloud infrastructure and internet access.

A major breakthrough for offline applications came with OpenAI Whisper, an open-source model capable of robust, multilingual transcription. Studies have shown it performs exceptionally well even in noisy classroom conditions. Critically, Whisper models can be downloaded and run completely locally. This comes with a trade-off, as the larger, more accurate models require significant computational resources.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
www.ijprems.com
editor@ijprems.com
Vol. 05, Issue 11, November 2025, pp : 229-233
e-ISSN : 2583-1062
Impact Factor : 7.001

For lower-resource environments, lightweight toolkits like Vosk have been recognized. Vosk provides an offline STT architecture that is ideal for situations where computational power is limited, offering a viable alternative for less powerful devices [5].

2.2 Text Summarization Approaches

Summarization techniques are broadly categorized as extractive or abstractive.

Extractive Summarization: This method identifies and concatenates the most important sentences from the source text. While computationally simple and efficient, this approach may not always produce fluent or coherent summaries.

Abstractive Summarization: This more advanced method uses AI models to rephrase and condense the original information, much as a human summarizer would. Transformer-based models such as BART [2] , T5, and Pegasus have become the standard for high-quality abstractive summarization in academic and industrial tasks. Research by Raffel et al. (2020) specifically demonstrated the effectiveness of the T5 (Text-to-Text Transfer Transformer) architecture in converting large text blocks into concise summaries with minimal information loss [3].

2.3 Gaps in Existing Solutions

A review of the literature and commercial tools reveals distinct gaps. Most academic prototypes focus on either transcription or summarization but rarely integrate both into a single offline pipeline. Commercial tools, while integrated, often require user subscriptions and store data on remote servers, creating privacy concerns. Furthermore, few systems are specifically designed with education-focused features like keyword extraction, PDF export, and complete offline accessibility. This proposed project positions itself as an integrated, education-focused, fully offline tool that ensures privacy while delivering high-quality summaries

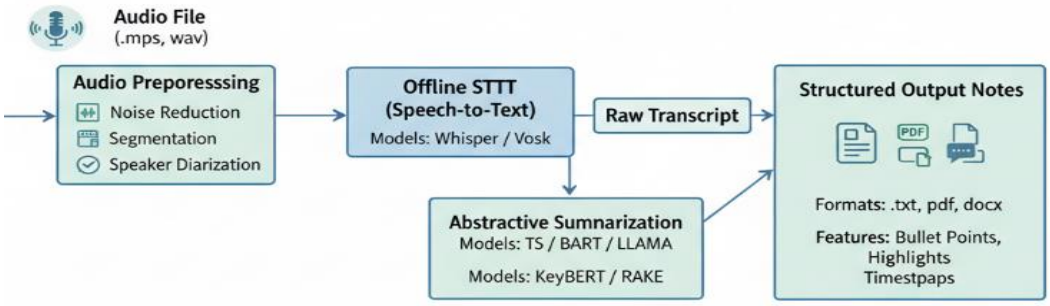
3. METHODOLOGY

The methodology for this project involves a step-by-step approach to create an integrated system that operates entirely without internet access. The architecture is designed as a sequential pipeline that ingests raw audio and outputs structured, summarized notes

3.1 Workflow Summary

The end-to-end process follows a clear sequence:

- Upload: The user uploads a pre-recorded lecture audio file (.wav, .mp3, .m4a).
- Preprocess: The audio is cleaned of background noise and segmented into manageable chunks.
- Transcribe: The local ASR (STT) model transcribes the audio segments into raw text.
- Structure: The raw text is cleaned, punctuated, and segmented by topic.
- Summarize & Extract: The local summarization model generates a concise summary of the text, and a separate algorithm extracts key terms.
- Export: The final structured notes, summary, and keywords are presented to the user and made available for download (.txt, .pdf, .docx).



3.2 Audio Acquisition and Preprocessing

The pipeline begins with the user providing a recorded lecture.

- **Acquisition:** The system accepts common audio formats such as .mp3, .wav, and .m4a.
- **Noise Reduction:** Before transcription, the audio undergoes preprocessing using techniques like spectral gating and bandpass filtering. This removes background noise, hum, and echo, which is crucial for ensuring high transcription accuracy.
- **Segmentation:** Long recordings are segmented into smaller, manageable audio chunks (e.g., 30-60 seconds) using silence detection. This optimizes the STT model's performance and manages memory usage.

www.ijprems.com
editor@ijprems.com
INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
(Int Peer Reviewed Journal)
Vol. 05, Issue 11, November 2025, pp : 229-233
e-ISSN : 2583-1062
Impact Factor : 7.001

- **Speaker Diarization:** Using tools like PyAnnote or WhisperX, the system identifies and labels different speakers (e.g., "Lecturer," "Student"). This enables the generation of more structured, readable transcripts and summaries.

3.3 Speech-to-Text (STT) Conversion

This module converts the preprocessed audio chunks into text.

- **ASR Model Integration:** The system integrates a local speech-to-text model, such as **Whisper Large-V3** or an equivalent, to perform offline transcription. This ensures privacy and eliminates any dependency on cloud APIs.
- **Punctuation & Formatting:** The raw text output from the ASR model is post-processed to add punctuation, proper capitalization, and paragraph structuring. This step is vital for making the transcript readable and effective for the subsequent summarization module.

3.4 Text Structuring and Topic Segmentation

Once the full transcript is generated, it is structured for summarization.

- **Topic Detection:** The transcript is segmented into sections based on topic shifts. This is achieved using local NLP algorithms like TextRank or clustering methods, which can identify when the lecture transitions to a new subject.
- **Key Point Extraction:** Important phrases, definitions, and facts are identified for potential inclusion in the summary.

3.5 Summarization and Key Insights Extraction

This module condenses the structured transcript.

- **Local LLM Summarization:** The system uses a locally hosted model, such as **LLaMA 3.1-8B** or **Mistral**, to generate concise, abstractive summaries of the lecture segments. This abstractive approach ensures the summary is coherent and rephrases the core meaning.
- **Keyword & Topic Extraction:** A separate algorithm (e.g., KeyBERT, RAKE) automatically generates a list of keywords and topics from the text. This provides a "quick-glance" view of the lecture's main concepts

3.6 Contextual Enhancement and Output Generation

The final step involves formatting the output for the user.

- **Keyword Highlighting:** The system highlights technical terms, formulas, and definitions in the final summary to aid quick review.
- **Output Formats:** The generated notes are organized with bullet points and highlighted terms. Users can export the final output in multiple formats, including .txt, .pdf, and .docx.
- **Timestamp-Linked Notes:** A key feature allows users to click a section of the generated summary to jump directly to that specific part of the original audio recording.

3.7 Tools and Technologies Used

The system is built using the following stack to ensure offline functionality:

- **ASR:** Vosk / Whisper (Local ASR)
- **NLP/Summarization:** TextRank / SpaCy, Local LLM (LLaMA / Mistral)
- **Audio Processing:** PyDub / Librosa
- **User Interface:** Streamlit / Tkinter
- **Report Generation:** ReportLab

4. CONCLUSION

This project proposes a solution to the significant and growing problem of information overload from recorded lectures. The work successfully outlines the design of an efficient, intelligent, and, most importantly, private system for transforming lengthy, unstructured audio into concise, well-structured notes. By leveraging advancements in local speech recognition and abstractive summarization, the system aims to save valuable time for students and enhance their comprehension and retention of key concepts.

The offline-first methodology ensures that the output is contextually accurate while also being completely private and accessible to all, regardless of internet connectivity. This work serves as a strong foundation for further research in the field of AI-powered educational tools.

5. FUTURE SCOPE

While the current scope delivers a functional tool, the underlying framework can be adapted and expanded. Suggestions for future improvement include:

- **Advanced Speaker Diarization**: Incorporating more robust speaker diarization to clearly differentiate between multiple speakers, such as the professor and students asking questions.
- **Multilingual Support**: Expanding the system to support multilingual transcription and summarization to cater to diverse and international audiences.
- **Personalized Summarization**: Allowing users to customize the desired level of detail for their summaries.
- **LMS Integration**: Integrating the tool directly with common Learning Management Systems (LMS) to expand usability and accessibility.
- **Mobile Platforms**: Optimizing the models for on-device processing on mobile platforms.

Abbreviations-

- **AI: Artificial Intelligence**
- **NLP: Natural Language Processing**
- **STT: Speech-to-Text**
- **ASR: Automatic Speech Recognition**
- **LLM: Large Language Model**
- **UI: User Interface**
- **LMS: Learning Management System**
- **MB** : Megabytes
- **RAM** : Random Access Memory

6. REFERENCES

[1] Brown, Tom B., et al. "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877-1901.

[2] Lewis, Mike, et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." Proceedings of ACL, 2020.

[3] Raffel, Colin, et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." Journal of Machine Learning Research, vol. 21, 2020, pp. 1-67.

[4] Wang, Yanshan, et al. "Clinical Information Extraction Applications: A Literature Review." Journal of Biomedical Informatics, vol. 77, 2018, pp. 34-49.

[5] Kocmi, Tom, et al. "Offline Speech Recognition for Low Resource Settings." IEEE Transactions on Audio, Speech, and Language Processing, 2021.

[6] Miller, Sarah, et al. "Impact of Automated Summarization on Student Learning Efficiency." Journal of Educational Technology Research, vol. 38, no. 4, 2022, pp. 451-468.Haidong Li, Jiongcheng Li, Xiaming Guan, Binghao Liang , Yuting Lai & Xinglong Luo , “Research on Overfitting of Deep Learning”, published in December 2019