International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Lecture Transcription and Content Summarization using Core Speech Recognition and AI-Agents

Dr. S. Vidya Sagar Appaji
Department of Computer Science
Raghu Institute of Technology, Visakhapatnam

S Hemanth Srinivas
Department of Computer Science
Raghu Institute of Technology, Visakhapatnam

Metta Ritesh Kumar
Department of Computer Science
Raghu Institute of Technology, Visakhapatnam

Vajrapu Murali
Department of Computer Science
Raghu Institute of Technology, Visakhapatnam

V Manasa Gummudu
Department of Computer Science
Raghu Institute of Technology, Visakhapatnam

I. INTRODUCTION

In the era of rapid digital transformation, educational institutions are increasingly adopting technology to enhance learning experiences. This research presents an AI-driven lecture transcription and summarization system designed to convert spoken lectures into concise, well-structured PDF summaries, bridging the gap between lengthy lecture content and efficient knowledge retention. The proposed system leverages state-of-the-art speech-to- text models and a multi-layered intelligent agent architecture, encompassing perception, decision-making, and action layers.

The perception layer captures and processes raw audio signals, extracting essential features and refining speech data for accurate transcription. The decision-making layer employs a large language model (LLM) [1] to distill key concepts, generate coherent summaries, and identify relevant references, ensuring contextual integrity and knowledge preservation. The action layer dynamically formats the refined content into a structured, accessible PDF document, ready for seamless distribution.

This approach not only streamlines knowledge acquisition but also reduces the cognitive load associated with reviewing extensive lecture recordings. It further enhances accessibility for students with diverse learning needs, promoting equitable access to educational resources. The system is engineered for real-time processing and iterative learning, continuously improving through feedback loops and model optimization. Our experimental evaluation indicates significant improvements in learning efficiency, comprehension, and content accessibility.

II. RELATED WORK

A. TEXT-TO-SPEECH (TTS)

Text-to-Speech (TTS) systems are intricate, multi-stage pipelines designed to convert written text into natural, human-like speech. This process involves several critical stages, each leveraging advanced algorithms, linguistic models, and deep learning techniques.

1. Text Preprocessing and Normalization
The first stage involves cleaning and normalizing the input text to make it suitable for speech generation. Tokenization splits the text into words or subwords, while text normalization expands abbreviations, converts numbers into words, and standardizes symbols. Phonetic conversion is another crucial step, where words are mapped to phonemes (distinct speech sounds) using lexicons or predictive models. Tools like the CMU Pronouncing Dic- tionary and models like Grapheme-to-Phoneme (G2P) converters are commonly used here. For tokenization, algorithms like Byte-Pair Encoding (BPE) help manage rare words by breaking them into subword units, which makes learning easier for neural networks.

2. Linguistic Analysis and Prosody Prediction
After normalization, the system extracts linguistic and prosodic features to enhance speech quality. Part-of-Speech (POS) tagging assigns syntactic roles, while syn- tax and semantic parsing help understand sentence struc- ture. Prosody prediction estimates pitch, duration, and intensity, adding rhythm and natural variation to speech. Models like BiLSTM-CRF excel at POS tagging, while sequence-to-sequence models such as Tacotron and Fast- Speech predict prosody with remarkable accuracy.

3. Acoustic Feature Generation
Next, linguistic and prosodic features are converted into acoustic representations like mel-spectrograms, which visually represent sound frequencies over time. Tacotron 2 generates high-quality spectrograms directly from text, while models like FastSpeech 2 offer faster, non-autoregressive alternatives, significantly reducing inference time while preserving quality.

3. Vocoder Waveform Synthesis
The mel-spectrogram is then fed into a vocoder, which synthesizes the final audio waveform. Early systems used concatenative or parametric synthesis, but modern TTS systems leverage neural vocoders like WaveNet and HiFi-GAN [4]. WaveNet uses dilated convolutional layers to model raw audio, while HiFi-GAN employs GANs [6] for fast, high-fidelity waveform generation.

3. PostProcessing and Optimization
The final stage involves refining the generated speech to enhance quality and adapt to specific requirements. Techniques like noise reduction and GAN-based enhancers remove artifacts and improve clarity. Voice adaptation methods, such as SV2TTS [5], enable speaker identity preservation, allowing systems to generate speech in different voices with minimal training data.

B. Mel-Frequency Cepstral Coefficients (MFCCs) in Speech Processing

Introduction to MFCCs Mel-Frequency Cepstral Coef- ficients (MFCCs) [8] are a widely used feature extraction technique in speech and audio processing. They capture the spectral properties of audio signals, making them crucial for applications like speech recognition, speaker identification, and Text-to-Speech (TTS) [9] synthesis. MFCCs [8] help represent audio signals in a way that closely aligns with human auditory perception, making them an indispensable tool in modern speech systems.

Theoretical Background MFCCs are based on the Mel scale, which approximates the human ear's sensitivity to different frequencies. The Mel scale is a perceptual scale where equal distances correspond to equal perceptual differences in pitch. By transforming the frequency domain into the Mel scale, MFCCs [8] allow speech models to focus more on perceptually significant frequency components.

Steps to Compute MFCCs The process of comput- ing MFCCs [8] involves several steps, each contributing to capturing the essential characteristics of the speech sig- nal: The process of computing MFCCs [8] involves several steps, each contributing to capturing the essential charac- teristics of the speech signal:

- **Pre-Emphasis**: The audio signal is passed through a high-pass filter to amplify higher frequencies, compensating for the natural attenuation of high- frequency components in human speech.
- **Framing and Windowing**: The signal is divided into short overlapping frames (typically 20-40ms) to capture stationary properties. A Hamming window is applied to each frame to reduce spectral leakage.
- **Fast Fourier Transform (FFT)**: Each windowed frame undergoes an FFT to convert the time-domain signal into the frequency domain.
- **Mel Filter Bank**: The frequency spectrum is passed through a series of triangular filters spaced according to the Mel scale, mimicking the non-linear frequency resolution of the human ear.
- **Discrete Cosine Transform (DCT)**: The logarithmic Mel spectrum is converted to the cepstral domain using a DCT, resulting in a set of coefficients that represent the speech signal's envelope.

C. Spatiotemporal Convolutional Neural Networks (STCNNs)

Spatio-Temporal Convolutional Neural Networks (STCNNs) [10] are widely used in speech-to-text conversion due to their capability to capture both spatial and temporal dependencies within audio signals. After the voice input is recorded and preprocessed to remove background noise, the system converts the raw waveform into a spectrogram or Mel Frequency Cepstral Coefficients (MFCC) [8] representation. These representations provide valuable information about the frequency and amplitude over time. STCNNs [10] apply 3D convolutions to these inputs, extracting meaningful spatio-temporal features that reveal phonetic patterns and speech characteristics. By leveraging this capability, STCNNs [10] improve the accuracy of phoneme recognition and reduce the impact of environmental noise. The foundational research by Tran et al. (2015) on learning spatiotemporal features using 3D convolutional networks supports the use of this architecture in speech processing tasks.

Speech recognition systems must model the inherent sequential nature of speech, as phonemes and syllables are interdependent over time. STCNNs [10] excel in capturing these temporal relationships by applying convolutional operations across both the frequency and time dimensions. Each layer captures increasingly abstract representations, identifying both short-term phonetic cues and long-term linguistic patterns. This multi-layered feature extraction process enables the system to achieve robust speech recognition, even in challenging acoustic conditions. Zhao et al. (2021) demonstrated the effectiveness of STCNNs [10] in noisy environments, where conventional CNNs [12] often fail to maintain temporal coherence.

Pooling layers further enhance the recognition accuracy of STCNNs [10] by reducing computational complexity while preserving essential features. In the context of speech recognition, temporal pooling consolidates information over time, reducing redundancy and emphasizing the most important phonetic features. Max-pooling and average-pooling techniques are commonly applied to retain high-value information, ensuring that the network focuses on significant speech characteristics. Research by Karpathy et al. (2014) illustrates how temporal pooling in 3D CNNs [12] can improve the performance of time-sequential tasks, a concept that is equally applicable to speech recognition systems.

Once the STCNN [10] has extracted spatio-temporal features, the output is passed to a Long Short-Term Memory (LSTM) [11] network for further sequential modeling. LSTMs are particularly effective at capturing long-term dependencies in speech, maintaining contextual information across extended sequences. This combination of STCNNs [10] for feature extraction and LSTMs [11] for temporal modeling results in highly accurate speech-to-text conversion. Graves et al. (2013) demonstrated the advantages of combining convolutional and recurrent neural networks for end-to-end speech recognition, validating the effectiveness of this architecture in modern speech processing applications.

D. RECURRENT NEURAL NETWORK

Recurrent Neural Networks (RNNs) [13] are fundamental in the speech-to-text pipeline of this project, providing robust temporal modeling by capturing sequential dependencies across audio frames. Unlike feedforward neural networks, RNNs [13] utilize recurrent connections to maintain a dynamic hidden state, effectively modeling the temporal correlations inherent in speech signals. This recurrent nature allows the network to preserve contextual information, making it particularly effective for recognizing phonetic patterns and predicting subsequent sounds. Following feature extraction using Spatio-Temporal Convolutional Neural Networks (STCNNs)] [10], the high-dimensional spectrogram or Mel Frequency Cepstral Coefficients (MFCC) [8] representation is fed into the RNN [13] for temporal feature learning. The network's ability to maintain memory over long sequences is crucial for capturing phonetic transitions and understanding coarticulation effects, where neighboring phonemes influence each other.

To mitigate the vanishing and exploding gradient problems commonly observed in standard RNNs, this project leverages Long Short-Term Memory (LSTM) [11] networks. LSTMs introduce a gating mechanism consisting of the input gate, forget gate, and output gate, which dynamically regulate the information flow through the network. This adaptive gating mechanism ensures effective gradient propagation over long sequences, enabling the network to retain essential information while discard- ing irrelevant data. Additionally, the use of Bidirectional LSTMs (BiLSTMs) [11] further enhances the model's contextual understanding by processing the input sequence in both forward and backward directions. This bidirectional processing facilitates improved phoneme disambiguation and word boundary detection, especially in scenarios involving homophones or ambiguous acoustic signals. Research by Graves et al. (2013) demonstrated the efficacy of BiLSTMs [11] in automatic speech recognition (ASR),
establishing them as a state-of-the-art choice for sequence-to-sequence tasks.

Furthermore, the RNN's [13] capacity for temporal alignment is enhanced using Connectionist Temporal Classification (CTC) [14] loss, which optimizes the model with- out requiring pre-aligned input-output pairs. CTC [14] employs a dynamic programming algorithm to align pre- dicted phoneme sequences with the actual target transcriptions, allowing for flexible sequence generation. This is particularly advantageous in real-time speech transcription, where variable-length audio inputs are processed efficiently. Additionally, attention mechanisms are integrated into the LSTM [11] architecture to focus on relevant portions of the speech signal, dynamically weighting the importance of different time steps. Such mechanisms, as demonstrated by Bahdanau et al. (2015), have significantly improved the performance of end-to-end ASR systems.

The integration of RNNs with LSTMs [11] in this project not only enhances speech recognition accuracy but also contributes to downstream tasks such as prosody prediction. By capturing variations in pitch, rhythm, and stress patterns, the network generates semantically coherent and acoustically natural outputs. These tem- poral patterns are then used in the subsequent AI agent processing stage for content generation and PDF synthesis. Overall, the application of RNNs [13] in the project ensures a robust and scalable solution for speech-to-text conversion, aligning with advancements in deep learning-based speech recognition research.

E. LONG SHORT TERM-MEMORY (LSTM)

LSTM networks [15], introduced by Hochreiter and Schmidhuber in 1997, were designed to address the vanishing gradient problem in traditional RNNs [13]. LSTMs [11] achieve this through a memory cell and three gating mechanisms:

- **Input Gate**: Controls how much new information is stored in the memory cell.
- **Forget Gate**: Determines which information to discard from the memory cell.
- **Output Gate**: Regulates how much information from the memory cell is used to compute the output.

The operations of an LSTM [15] unit at time step are defined as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \cdot \tanh(C_t)$$

Where:

- f_t, i_t, and o_t are the forget, input, and output gates, respectively.
- C_t is the memory cell state.
- h_t is the hidden state.
- σ is the sigmoid activation function.
- W and b are learnable weights and biases.

Limitations of LSTMs

- **Computational Complexity**: LSTMs [15] are computationally expensive due to their complex architecture.
- **Sequential Processing**: They process data sequentially, limiting their parallelizability.

F. AI AGENTS

The rise of Large Language Model (LLM)-based [1] AI agents has transformed various domains, including content generation, decision-making, and automation. AI agents are intelligent entities capable of perceiving, reasoning, and taking actions in a given environment. By leveraging LLMs [1] as their core computational framework, these agents demonstrate advanced natural language processing (NLP) [16] capabilities that facilitate autonomous content creation. This research explores how AI agents can be effectively utilized to generate structured and well-organized PDF summaries, particularly in educational and academic applications.

Conceptual Framework of AI Agents AI agents typically consist of three primary components: the brain, the perception module, and the action module. The brain, often powered by an LLM [1], serves as the core processor for reasoning, planning, and decision-making. The perception module enables the agent to interpret multimodal inputs such as text, speech, and images. The action module executes tasks, including content generation and document formatting.

Application of AI Agents in PDF Content Generation In the context of lecture transcription and summarization, AI agents automate the process of converting spoken words into structured, readable text, which is then formatted into a PDF document. The workflow involves several stages:

- **Speech Recognition and Transcription** – AI agents use Automatic Speech Recognition (ASR) models such as Whisper and Google Speech-to-Text to convert audio input into textual data.

- **Contextual Understanding and Summarization** – The transcribed text is processed by an LLM [1], which extracts key points, eliminates redundant information, and structures the content into coherent paragraphs.

- **Content Formatting and Structuring** – The AI agent applies formatting rules, organizing the content with headings, bullet points, and references, creating a visually structured summary.

Advantages of AI-Generated PDF Summaries AI-driven summarization enhances accessibility, efficiency, and accuracy. It reduces the need for manual note-taking, offers concise yet comprehensive overviews, and improves comprehension by structuring information logically. Additionally, AI agents [1] can personalize summaries based on user preferences, making them highly adaptable to diverse learning needs.

III. METHODOLOGY

The proposed system follows a structured pipeline that ensures efficient and accurate conversion of voice input into a summarized and formatted PDF document. The methodology consists of the following stages:

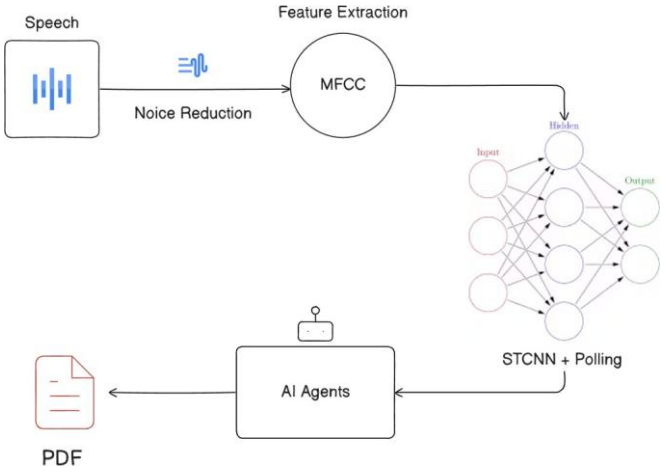


Figure 1: Architecture

The proposed system follows a structured pipeline that ensures efficient and accurate conversion of voice in- put into a summarized and formatted PDF document. The methodology consists of the following stages:

Voice Recording and Preprocessing: The process begins with capturing the user's voice input through a microphone. The recorded audio undergoes noise reduction techniques such as Spectral Subtraction or Wiener Filtering to enhance clarity by removing background disturbances. This step ensures that unwanted noise does not interfere with subsequent processing.

Feature Extraction using MFCC: Once the noise is reduced, the system extracts Mel-Frequency Cepstral Coefficients (MFCCs) [8], which are critical features rep- resenting the spectral properties of the speech signal. The

MFCC algorithm captures phonetic information by applying a series of transformations, including the Fourier Transform and Mel filter banks, ensuring that the features align with human auditory perception. These extracted features serve as inputs to deep learning models for speech-to-text conversion.

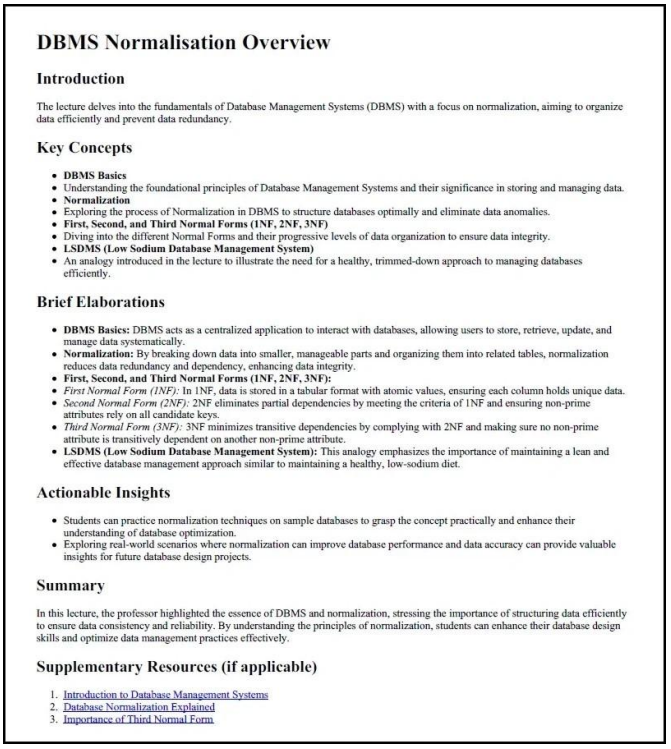
Speech-to-Text Conversion: The MFCC [8] features are fed into a Speech Temporal Convolutional Neural Network (STCNN) [10], which enhances local feature extraction while maintaining sequential information. The output is then passed through a Long Short-Term Memory (LSTM) [11] network, which specializes in handling temporal dependencies in speech data. The LSTM [11] processes the time-sequenced data and converts it into textual representations using a Linear [17] + Softmax [18] layer to map features to corresponding words.

AI Agent-Based Summarization: Once the speech is transcribed into text, the system employs AI agents powered by Large Language Models (LLMs) [1] such as **GPT-4, LLaMA, or PaLM-2** to summarize and struc- ture the content. The summarization process follows multiple stages:

- **Context Understanding**: The AI agent applies Named Entity Recognition (NER) and Dependency Parsing to extract key concepts and relationships..
- **Summarization Strategies**: The system leverages TextRank, BART (Bidirectional and Auto-Regressive Transformers), or PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization) to condense the information while preserving key insights.
- **Formatting and Coherence Optimization**: The AI agent structures the text into well-organized paragraphs, bullet points, and headings for improved readability and comprehension.

Content Structuring and Formatting: The summarized text undergoes further processing to ensure a structured format suitable for professional and academic use. This stage involves applying predefined templates for consistent formatting, enhancing readability using section headings, bullet points, and numbered lists, and incorporating citations and references to maintain academic integrity.

PDF Generation and Export: The final structured content is converted into a PDF document using tools such as LaTeX, ReportLab, or Pandas. Post-processing ensures alignment, font consistency, and accessibility, producing a high-quality, professional document ready for distribution.



DBMS Normalisation Overview

Introduction

The lecture delves into the fundamentals of Database Management Systems (DBMS) with a focus on normalization, aiming to organize data efficiently and prevent data redundancy.

Key Concepts

- **DBMS Basics**
- Understanding the foundational principles of Database Management Systems and their significance in storing and managing data.
- **Normalization**
- Exploring the process of Normalization in DBMS to structure databases optimally and eliminate data anomalies.
- **First, Second, and Third Normal Forms (1NF, 2NF, 3NF)**
- Diving into the different Normal Forms and their progressive levels of data organization to ensure data integrity.
- **LSDMS (Low Sodium Database Management System)**
- An analogy introduced in the lecture to illustrate the need for a healthy, trimmed-down approach to managing databases efficiently.

Brief Elaborations

- **DBMS Basics:** DBMS acts as a centralized application to interact with databases, allowing users to store, retrieve, update, and manage data systematically.
- **Normalization:** By breaking down data into smaller, manageable parts and organizing them into related tables, normalization reduces data redundancy and dependency, enhancing data integrity.
- **First, Second, and Third Normal Forms (1NF, 2NF, 3NF):**
- *First Normal Form (1NF):* In 1NF, data is stored in a tabular format with atomic values, ensuring each column holds unique data.
- *Second Normal Form (2NF):* 2NF eliminates partial dependencies by meeting the criteria of 1NF and ensuring non-prime attributes rely on all candidate keys.
- *Third Normal Form (3NF):* 3NF minimizes transitive dependencies by complying with 2NF and making sure no non-prime attribute is transitively dependent on another non-prime attribute.
- **LSDMS (Low Sodium Database Management System):** This analogy emphasizes the importance of maintaining a lean and effective database management approach similar to maintaining a healthy, low-sodium diet.

Actionable Insights

- Students can practice normalization techniques on sample databases to grasp the concept practically and enhance their understanding of database optimization.
- Exploring real-world scenarios where normalization can improve database performance and data accuracy can provide valuable insights for future database design projects.

Summary

In this lecture, the professor highlighted the essence of DBMS and normalization, stressing the importance of structuring data efficiently to ensure data consistency and reliability. By understanding the principles of normalization, students can enhance their database design skills and optimize data management practices effectively.

Supplementary Resources (if applicable)

1. Introduction to Database Management Systems
2. Database Normalization Explained
3. Importance of Third Normal Form

Figure 2: Sample Result

IV. EVALUATION METRICS

Evaluating the performance of the AI-driven lecture transcription and summarization system requires a combination of quantitative and qualitative metrics. The evaluation process focuses on assessing the accuracy, efficiency, and quality of both the speech-to-text (STT) and text summarization components.

The accuracy of the speech-to-text conversion is typically measured using Word Error Rate (WER), which is a widely accepted metric in automatic speech recognition (ASR) tasks. WER is calculated using the formula:

$$\text{WER} = \frac{S + D + I}{N}$$

N is the total number of words in the reference transcript. A lower WER indicates higher transcription accuracy. Additionally, Character Error Rate (CER) can be used for languages with complex morphology, providing a more granular evaluation by assessing character-level accuracy. These metrics are essential for understanding how accurately the system transcribes spoken content, especially in challenging environments with background noise or diverse speaker accents.

The quality of the summaries generated by the system is evaluated using a mix of automated metrics and hu- man assessments. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measures the overlap of n-grams, word sequences, and word pairs between the generated summary and a reference summary. It is particularly

useful in assessing the recall of key information. BLEU (Bilingual Evaluation Understudy), although originally developed for machine translation, is also applied to measure the precision of the generated summaries by compar- ing them against human-written references. Additionally, BERTScore utilizes contextual embeddings from BERT (Bidirectional Encoder Representations from Transformers) to evaluate the semantic similarity between the predicted and reference summaries, offering a more nuanced understanding of the quality of content generation.

The validation and testing process involves using diverse datasets that include lectures with varying topics, accents, noise levels, and speaking speeds. Benchmark datasets such as LibriSpeech, TED-LIUM, and AMI Meeting Corpus are commonly used for evaluating the speech-to-text module, while summarization models are evaluated using datasets like CNN/DailyMail, XSum, or the Scientific Papers Dataset. By using these well-established datasets, the system's performance can be fairly compared with existing solutions. This ensures a comprehensive evaluation of its effectiveness across different scenarios.

In addition to these quantitative evaluations, user feedback plays a critical role in the validation phase. End users, including students and educators, provide feedback on the usefulness, clarity, and accuracy of the generated summaries. This feedback is used to further fine-tune the models through iterative learning processes. Active learning strategies are implemented, allowing the system to identify and prioritize areas where improvements are needed. This continuous feedback loop ensures that the system evolves to meet user expectations while maintaining high standards of performance.

Through the combination of multiple evaluation metrics, extensive dataset testing, and user feedback, the proposed system demonstrates robust performance. It generates accurate transcripts and coherent summaries with minimal errors, reducing information overload and enhancing accessibility. This comprehensive evaluation validates the system's effectiveness in providing educational support and improving knowledge retention.

V. CONCLUSION

AI agents, powered by LLMs, offer a robust solution for automated lecture transcription and PDF summarization. Their ability to process natural language, extract key insights, and format structured documents makes them invaluable in academic and professional environments. Future research should focus on improving contextual accuracy, integrating multimodal data processing, and enhanc- ing adaptability for personalized summarization. This advancement will further streamline information management and knowledge dissemination, addressing the grow- ing demand for efficient, AI-driven content generation.

REFERENCES

[1] Xi et al. (2023) in The Rise and Potential of Large Language Model-Based Agents: A Survey discuss how LLMs enhance AI agents' reasoning and decision- making capabilities, making them ideal for automated content generation.
[2] Rabiner, L., & Juang, B. H. (1993). Fundamentals of Speech Recognition. Prentice Hall.
[3] Ren, Y., et al. (2020). "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech." NeurIPS 2020
[4] Kong, J., et al. (2020). "HiFi-GAN: Generative Adver- sarial Networks for Efficient and High Fidelity Speech Synthesis." NeurIPS 2020.
[5] Jia, Y., et al. (2018). "Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthe- sis." NeurIPS 2018.
[6] Goodfellow, I., et al. (2014). "Generative Adversarial Nets." NeurIPS 2014.
[7] Oord, A. v. d., et al. (2016). "WaveNet: A Generative Model for Raw Audio." SSW 2016.
[8] Davis, S., & Mermelstein, P. (1980). "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." IEEE Transactions on Acoustics, Speech, and Signal Processing.
[9] Shen, J., et al. (2018). "Natural TTS Synthesis by Con- ditioning WaveNet on Mel Spectrogram Predictions." ICASSP 2018.
[10] Tran, D., et al. (2015). "Learning Spatiotemporal Features with 3D Convolutional Networks." ICCV 2015.
[11] Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." Neural Computation.
[12] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). "ImageNet Classification with Deep Convolutional Neural Networks." NeurIPS 2012.
[13] Mikolov, T., et al. (2010). "Recurrent neural network based language model." Interspeech 2010.
[14] Graves, A., Fernández, S., Gomez, F., & Schmidhu- ber, J. (2006). "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks." Proceedings of the 23rd Interna- tional Conference on Machine Learning (ICML).
[15] Alex Sherstinsky. Fundamentals of recurent neural network(rnn) and long short-term memory (lstm)network. Physica D:Nonlinear Phenomena, 404:132306,2020.
[16] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). "Efficient Estimation of Word Representations in Vector Space."
[17] Bridle, J. S. (1990). "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition." In Neurocomputing.
[18] Bishop, C. M. (2006). "Pattern Recognition and Machine Learning." Springer.