International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Generating Meeting Transcription Using Natural Language Processing

Swati Tilkar[1] Prof. Ruchika Pachori[2]
Department of Information Technology[1,2]
MIT, Ujjain, India[1,2]

ABSTRACT: Natural Language Processing plays a pivotal role in automating the transcription of meetings. It enables machines to understand, interpret, and generate human language. In meeting transcription, NLP components such as Automatic Speech Recognition (ASR), speaker diarization, entity recognition, summarization, and sentiment analysis work together to produce accurate and readable transcripts. ASR converts spoken words into text, while NLP refines the raw output by correcting grammatical errors, identifying speakers, and structuring dialogue for readability and comprehension. Ethical considerations, including data privacy and bias in multilingual environments, are also examined. The findings suggest that NLP-driven transcription tools not only simplify record-keeping but also offer valuable insights for decision-making and knowledge management. As technology advances, such tools are poised to become essential in professional settings, providing reliable, scalable solutions with minimal human intervention. This research focusses on leveraging NLP models for generating machine transcripts. The word error rate (WER) and character error rate (CER) have been chosen as the performance metrics.

Keywords: Natural Language Processing (NLP), Meeting Transcriptions, wav2vec, Whisper, Word Error Rate (WER), Character Error Rate (CER).

I. Introduction

Meetings are integral to decision-making and collaboration in modern organizations. However, manually recording meeting minutes or transcriptions is time-consuming, error-prone, and inefficient [1]. With advancements in Artificial Intelligence (AI), particularly Natural Language Processing (NLP), it is now possible to automate meeting transcription with high accuracy and real-time capability. NLP-powered transcription systems can convert speech into structured text, capture key topics, and summarize important discussion points. This essay explores the various aspects of generating meeting transcriptions using NLP, highlighting technologies, methodologies, applications, and challenges [2].

Automated meeting transcription has wide applications across industries. In corporate environments, it aids in creating meeting minutes, compliance documentation, and project tracking. In healthcare, it assists doctors by transcribing patient consultations. In education, it provides accessible lecture transcripts and learning materials. Legal firms use it to document proceedings and testimonies. In order to effectively generate transcripts, it is mandatory to understand the fundamentals of speech recognition [3].

Fig.1. Illustration of Speech Recognition

Figure 1 depicts the speech recognition technology.

Speech recognition, also known as automatic speech recognition (ASR), is the process of transforming spoken language into text using computational models. It involves several stages: signal processing, feature extraction, acoustic modeling, language modeling, and decoding. Modern ASR systems use deep learning algorithms such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer-based models to improve recognition accuracy across different languages, accents, and environments [4]. These models learn patterns in speech and are trained on vast datasets to transcribe spoken words into coherent text [5].

II. Background of Speech Recognition.

Typically speech is sampled at twice the maximum frequency of audio (in case audio is mixed with speech), based on Shannon's sampling theorem stating [6]:

$$f_s \geq 2 * f_m \quad (1)$$

Here,
f_s is the speech sampling frequency
f_m is the maximum audio frequency

Without loss of generality, considering the human range of audio to extend from 20Hz to 20kHz, the sampling frequency is typically kept at 44.1kHz for most practical scenarios for speech processing. The additional 4.1kHz are for oversampling for the digital audio tape (.dat) files stored on digital systems [7].

Speech recognition plays a central role in generating meeting transcriptions by enabling real-time or post-meeting conversion of dialogue into text. It eliminates the need for manual note-taking, allowing participants to focus on the discussion. With speech recognition, transcriptions can capture every word spoken, identify speakers, and provide searchable records for future reference. This is particularly valuable in business meetings, educational settings, legal proceedings, and remote work environments, where accurate documentation is crucial for follow-up, compliance, and knowledge sharing.

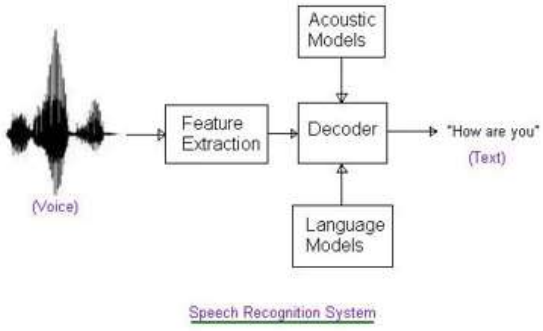

Fig.2 Parts of Speech Recognition System

Figure 2 depicts the parts of speech recognition system which are [8]:

1. Voice capturing mechanism (transducer)
2. Feature Extraction Module
3. Acoustic Module
4. Decoder
5. Language Module

Audio Acquisition: The first step in a speech recognition system is audio signal acquisition. This involves capturing the spoken input using a microphone or recording device. The captured audio is usually in the form of a continuous waveform, which contains variations in amplitude and frequency over time. This raw audio signal is digitized and often processed to remove noise and enhance quality. The sampling rate and bit depth determine how accurately the sound is captured for further processing.

Feature Extraction: Once the audio signal is captured, it is preprocessed to extract meaningful features. Preprocessing may include noise reduction, silence removal, and normalization of volume. Feature extraction is the process of identifying relevant acoustic patterns in the speech signal. One of the most common techniques is the extraction of Mel-Frequency Cepstral Coefficients (MFCCs), which represent the power spectrum of sound. These features reduce the complexity of the data and highlight the characteristics that differentiate various phonemes [9].

Acoustic Modeling: The next stage is acoustic modeling, where the system attempts to map the extracted features to basic sound units or phonemes. Acoustic models are typically built using machine learning techniques and are trained on large datasets of spoken language. Traditionally, Hidden Markov Models (HMMs) were used for this purpose, but modern systems use deep learning models such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers for more accurate acoustic predictions. These models learn how specific sounds are represented in the feature space.

Language modeling: This helps the system understand the context in which words are spoken. While acoustic models identify the likely phonemes or words based on sound, language models determine the most probable word sequences based on grammar, syntax, and usage frequency. For instance, if the system hears “I scream,” it uses a language model to avoid transcribing it as “ice cream” unless the context fits. Modern language models are based on probabilistic approaches (like n-grams) or deep learning-based approaches [10].

Decoding and Hypothesis Generation: In this step, the system combines the outputs of the acoustic and language models to determine the most likely transcription. A decoder evaluates multiple possible sequences (called hypotheses) and selects the one with the highest probability. This process involves search algorithms such as the Viterbi algorithm to find the optimal path through the combined model space. The

final output is a textual representation of the spoken input that is as close as possible to what was actually said.

Post-processing: Once the transcription is generated, the system performs post-processing to improve readability and user understanding. This may include adding punctuation, capitalizing proper nouns, correcting grammar, and identifying named entities like people or organizations. In some systems, speaker diarization is applied to differentiate between multiple speakers. Finally, the clean and formatted transcription is output to the user or integrated into downstream applications such as subtitles, meeting summaries, or voice commands.

Existing Challenges [11]:

While speech recognition has significantly advanced, several challenges still impact its effectiveness in meeting transcription. Background noise, multiple speakers talking simultaneously, accents, speech clarity, and domain-specific vocabulary can reduce transcription accuracy. Recognizing different speakers (speaker diarization) and maintaining context in long conversations remain complex tasks. Moreover, privacy and security concerns arise when meetings are transcribed using cloud-based services, especially when sensitive information is involved.

Despite its advantages, automated transcription using NLP faces several challenges. Background noise, overlapping speech, multiple accents, and domain-specific jargon can significantly reduce transcription accuracy. Speaker diarization, especially in large group meetings, remains an ongoing challenge. Furthermore, privacy and data security concerns arise when sensitive information is transcribed and stored. Ethical considerations about data ownership and consent must also be addressed when deploying transcription services in regulated industries.

III. Existing Statistical Models

The existing machine learning and deep learning models employed for speech recognition and transcription are [12]:

Hidden Markov Models (HMMs)
Hidden Markov Models were among the earliest and most widely used models in speech recognition. Figure 2 depicts a Hidden Markov Model.

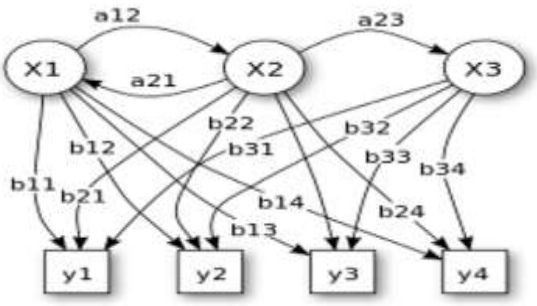

Fig.3 Hidden Markov Model

HMMs are statistical models that represent sequences of observable events (like spoken words) with underlying hidden states (such as phonemes). They assume that the current state depends only on the previous state, making them suitable for modeling temporal sequences like speech. HMMs work well when combined with Gaussian Mixture Models (GMMs) to model acoustic signals, but they struggle with capturing long-range dependencies and complex variations in speech.

Deep Neural Networks (DNNs)
The introduction of Deep Neural Networks marked a major improvement in speech recognition performance. DNNs can learn complex patterns from large datasets and are effective at classifying speech features. When used in combination with HMMs (in hybrid models), DNNs significantly improved acoustic modeling. However, they still had limitations in handling sequential data over time, which led to the development of more specialized architectures like RNNs [13].

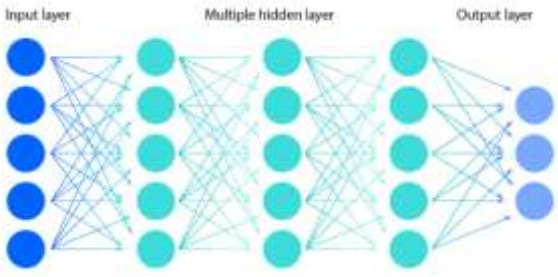

Fig.4 A Deep Neural Network

The input-output relation of a neural network is given by:

$$y = f(\sum_{i=1}^{n} x_i w_i + b) \quad (1)$$

Here,
x denote the parallel inputs
y represents the output
w represents the bias
f represents the activation function

Recurrent Neural Networks (RNNs) and LSTMs:
Recurrent Neural Networks (RNNs) were developed to handle sequential data like audio signals by maintaining a memory of previous inputs. Long Short-Term Memory networks (LSTMs), a type of RNN, address the problem of vanishing gradients, allowing the model to remember information over longer time intervals. LSTMs became widely used in speech recognition systems due to their ability to model context and dependencies in spoken language, greatly improving the natural flow and accuracy of transcriptions.

The LSTM networks are a specialized type of recurrent neural network (RNN) designed to process and predict data sequences by learning long-term dependencies. Unlike traditional RNNs, which suffer from vanishing or exploding gradient problems during training, LSTMs incorporate a unique architecture with gates and memory cells that help retain important information over long periods [14].

The LSTM primarily has 3 gates:
1) Input gate: This gate collects the presents inputs and also considers the past outputs as the inputs.
2) Output gate: This gate combines all cell states and produces the output.
3) Forget gate: This is an extremely important feature of the LSTM which received a cell state value governing the amount of data to be remembered and forgotten.

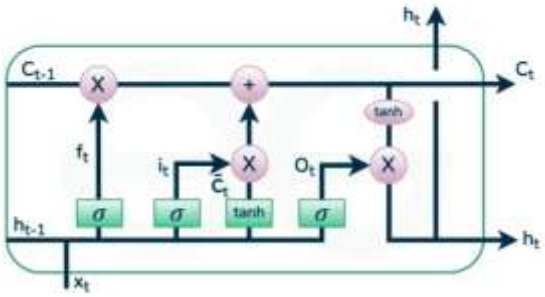

Fig.5 The LSTM Model

Figure 5 depicts the LSTM model. The relation to forget by the forget gate is given by:

$$\boldsymbol{f = \sigma(W_f[h_{t-1}, x_t] + b_i)} \qquad \textbf{(2)}$$

Here,
f denotes forget gate activation
w_f are forget gate weights.
h_{t-1} Denotes Hidden state from the previous time step
x_t is present input.
b_i is the bias

The advantages of LSM are:
Capturing Long-Term Dependencies: LSTMs maintain long-term memory using the cell state, unlike traditional RNNs.
Mitigating Vanishing/Exploding Gradients: Gates help regulate gradient flow, enabling stable training over long sequences.
Versatility: Useful for several time series prediction problems.

Connectionist Temporal Classification (CTC):
One of the key innovations in end-to-end speech recognition is Connectionist Temporal Classification (CTC). CTC allows models to learn the alignment between input audio frames and output text sequences without requiring pre-segmented training data. It is especially useful for training speech-to-text models directly. CTC is often used with RNNs or LSTMs and forms the basis for systems like DeepSpeech, an open-source speech recognition engine developed by Mozilla.

Sequence-to-Sequence Models and Attention Mechanisms:
Sequence-to-sequence (seq2seq) models revolutionized speech recognition by treating it as a translation problem—translating audio sequences into text. These models use an encoder-decoder architecture where the encoder processes the input audio, and the decoder generates the text output. Attention mechanisms further enhance seq2seq models by allowing the decoder to focus on relevant parts of the input at each step. This led to more accurate and context-aware transcriptions.

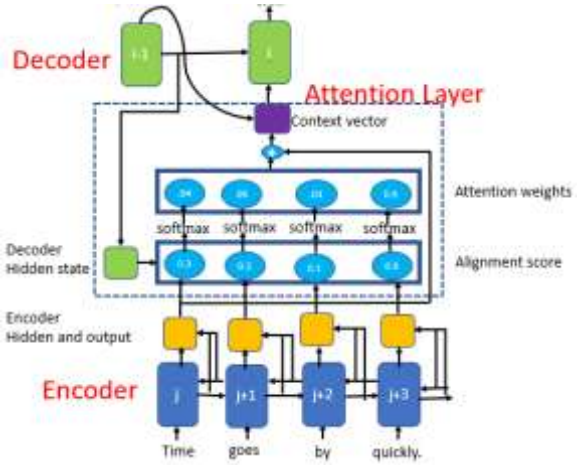

Fig.6 Sequence to Sequence Attention Model

Figure 6 depicts a Sequence-to-Sequence Models and Attention Model.

Transformer-Based Models:
Transformers, particularly models like Wav2Vec 2.0 by Facebook AI and Whisper by OpenAI, represent the current state-of-the-art in speech recognition [15].

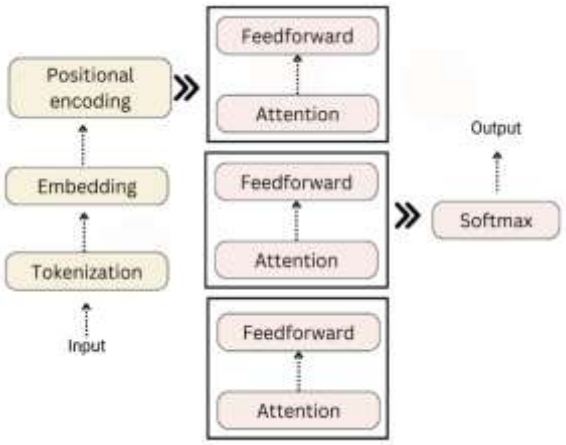

Fig.7 Transformer Model for NLP

Figure 7 depicts the transformer model for NLP. Unlike RNNs, transformers process entire sequences in parallel, allowing them to capture long-range dependencies more efficiently. Wav2Vec 2.0 is a self-supervised model that learns speech representations from raw audio without labeled data, making it highly adaptable. Whisper, on the other hand, is a multilingual, multitask model capable of transcription, translation, and language identification. These models outperform previous architectures in terms of accuracy, speed, and robustness to noise.

End-to-End vs. Hybrid Systems:
Modern speech recognition systems can be broadly categorized into hybrid and end-to-end models. Hybrid systems combine various components (like acoustic, language, and pronunciation models), often using DNNs and HMMs. End-to-end models, such as those using CTC or transformers, simplify the pipeline by directly mapping audio to text. While end-to-end models are more elegant and easier to maintain, hybrid systems still offer competitive accuracy in specific domains and are widely used in commercial applications

IV. Proposed Methodology

The proposed methodology explores two different models [16]:

- Wav2Vec
- Whisper

Each of them is discussed:

Wav2Vec:
The wav2vec model, developed by Facebook AI (now Meta AI), stands out as a powerful self-supervised learning approach for processing raw audio waveforms. Unlike traditional speech recognition pipelines that rely on hand-crafted features like MFCCs, wav2vec learns meaningful speech representations directly from audio data. This essay provides a comprehensive overview of how the wav2vec model works, its components, advantages, and its transformative role in modern speech recognition systems

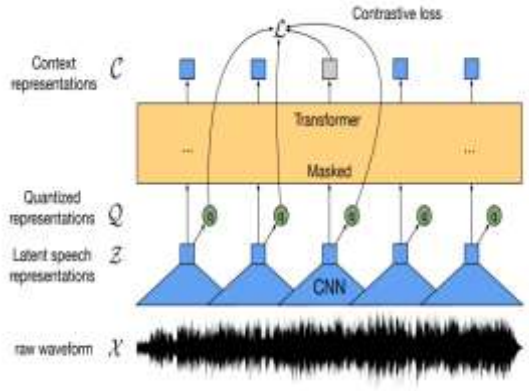

Fig.8 The wav2vec model

The core architecture of wav2vec 2.0, the most popular version, consists of three main components: a feature encoder, a contextualized transformer-based network, and a quantization module (used in pre-training). The feature encoder is a stack of convolutional layers that processes raw audio into latent speech representations. These representations are then fed into a Transformer network that captures long-range dependencies and contextual information. During pre-training, the model learns to predict masked parts of the audio sequence, similar to how BERT masks words in NLP. A key innovation in wav2vec 2.0 is its use of self-supervised learning. The model is trained on unlabeled speech by masking segments of the encoded audio and tasking the network with predicting them from surrounding context. This approach enables the model to learn phonetic and linguistic structures without needing transcripts. Once pretrained, the model can be fine-tuned on a relatively small labeled dataset to perform downstream tasks like automatic speech recognition (ASR), making it both data-efficient and powerful.

Whisper Model:
The Whisper model is built on a transformer-based encoder-decoder architecture, similar to those used in natural language processing. The encoder takes raw

audio that has been converted into log-Mel spectrograms, a time-frequency representation of the sound. The decoder then generates the corresponding text, one token at a time, while attending to the encoder's output. The model supports multiple tasks, such as transcription, translation, and voice activity detection, through the use of special tokens that guide the decoder to perform the desired operation [17].

Whisper was trained using a supervised learning approach on a massive multilingual and multitask dataset. The training data included both audio and the corresponding text in various languages, with some paired with English translations. This diversity enabled the model to learn cross-lingual mapping and generalize across languages and dialects. Unlike self-supervised models like wav2vec, which require separate fine-tuning, Whisper's supervised approach provides high accuracy out-of-the-box and requires minimal customization for real-world tasks.

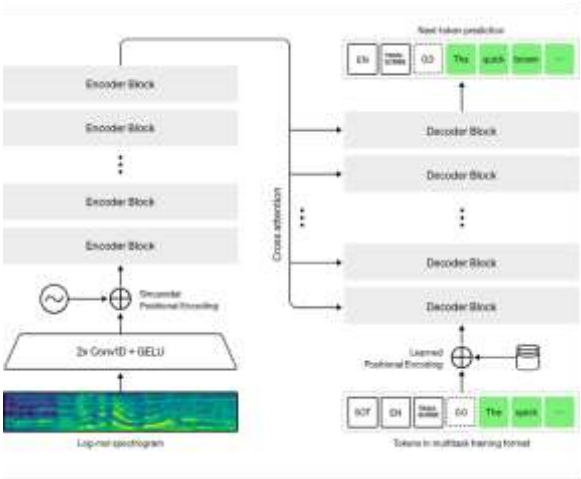

Fig.9 The Whisper Model

Whisper excels in several key areas. First, it offers high accuracy, even with background noise, accents, or informal speech. Second, it requires no fine-tuning to operate effectively in many environments, making it user-friendly. Third, it supports zero-shot translation, meaning it can translate speech directly from one language to another without needing paired training data. Furthermore, as an open-source model, Whisper can be integrated into various platforms, customized for specific needs, and used in privacy-conscious offline environments [18].

The training loss for the models is computed as [19]:

$$mse = \frac{\sum_{i=1}^{n} e_i^2}{n} \quad (3)$$

The word error rate (WER) is defined as [20]:

$$WER = \frac{S+D+I}{N} \quad (4)$$

Here,
S is Number of substitutions (wrong words)
D is Number of deletions (missing words)
I is Number of insertions (extra words)
N is Total number of words in the reference (ground truth).

WER measures the minimum number of word-level edits (insertions, deletions, substitutions) needed to match the system output to the reference, normalized by the number of words in the reference [21].

The character error rate (WER) is defined as:

$$CER = \frac{S+D+I}{N} \quad (5)$$

Here,
S is Number of substitutions (wrong characters)
D is Number of deletions (missing characters)
I is Number of insertions (extra characters)
N is Total number of characters in the reference (ground truth).

CER is used when fine-grained evaluation is needed, especially in languages where word boundaries are unclear (e.g., Chinese, Japanese) or in noisy data conditions.

V. Experimental Results

The Open SLR language dataset is used for this project.

Fig.10 Pre-Processing

The first step is the pre-processing step which includes normalization and removal of background noise.

Fig.11 Model Selection

The model selected is the Wav2Vec2-XLSR-Large-Hindi model from Hugging Face. The hyperparameter tuning details were:

Learning rate of 1e-4, a batch size of 8, and gradient accumulation steps of 2, number of training epochs is 30 and evaluation steps.

Fig.12 Training Model

The next phase is the model training and testing. The values of the evaluation metrics were the CER and AER.

Fig.13 Model Training Parameter

The model training parameters in terms of training loss, validation loss, WER and step is depicted in figure above. The WER obtained is 72.6%.

Due to the large WER of the wav2vc model, the next approach tested was the Whisper model.

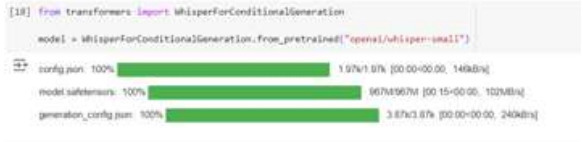

Fig.14 Model Selection: Whisper

The Whisper model was selected from the Hugging face API. Whisper is a transformer-based sequence-to-sequence model trained on 680,000 hours of supervised multilingual audio collected from the web.

Fig.15 Hyperparameters Tuning

Hyperparameters tuning details are:

Learning rate of 1e-5, a batch size of 16, and gradient accumulation steps of 1, number of training epochs is 3 and evaluation steps.

Fig.16 Performance Evaluation

The figure depicts the model's performance. The model attains:

WER of 19.83% , training loss of 0.122% and validation loss of 0.29%.

It can be observed that the Whisper Model clearly beats the wav2vec model in terms of WER.

The difference in results stem from the fundamental difference between the two models lies in their training methodology. Wav2vec 2.0 is primarily a self-supervised model, trained on large unlabeled audio datasets and later fine-tuned on specific transcribed data. In contrast, Whisper is trained in a fully supervised manner using a large paired audio-text data, covering a wide range of languages, accents, and background noise conditions. This extensive and diverse training allows Whisper to perform more reliably out-of-the-box, without requiring task-specific fine-tuning, giving it a practical advantage in real-world deployment.

Whisper is designed to handle noisy audio, overlapping speech, and diverse speaking styles, thanks to the varied and realistic training dataset it was built on. Wav2vec 2.0 performs well under controlled conditions but tends to be more sensitive to background noise, poor microphone quality, and accented speech unless fine-tuned. Whisper's robustness to audio imperfections makes it more effective in real-world settings such as interviews, phone calls, lectures, and live events.

The final deployment is done using the Gradio platform.

Fig.17 Deployment through Gradio

The above figure depicts the web application developed in the research work.

V. CONCLUSION

It can be concluded that effective documentation of meetings is essential for collaboration, accountability, and decision-making. Manual note-taking, however, is time-consuming, error-prone, and often incomplete. Automated meeting transcription systems offer a solution by converting speech to text in real time or post-processing. Leveraging advanced deep learning models like Wav2Vec 2.0 and Whisper, such systems can provide accurate, multilingual, and context-aware transcriptions. Wav2Vec 2.0 offers customization and strong acoustic modeling, while Whisper delivers multilingual, multitask performance with ease of deployment. Together, they enable the creation of intelligent transcription systems that are scalable, accessible, and suited to the complexities of real-world meeting environments. In the proposed work, it has been shown that the Whisper model clearly outperforms the wav2vec model in terms of training loss and WER, making it more suitable for practical utility.

References:

[1] T. Yoshioka et al., "Advances in Online Audio-Visual Meeting Transcription," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding (ASRU)*, 2019, pp. 1–8

[2] A. Baevski, H. Zhou, A. Mohamed and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1–6

[3] S. Schneider, A. Baevski, R. Collobert and M. Auli, "wav2vec: Unsupervised Pre-training for Speech Recognition," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding (ASRU)*, 2019, pp. 1–6

[4] Y. Li et al., "Self-Supervised Learning-Based Source Separation for Meeting Data," arXiv preprint arXiv:2304.00871, 2023.arXiv

[5] C. Zhu, R. Xu, M. Zeng and X. Huang, "A Hierarchical Network for Abstractive Meeting Summarization with Cross-Domain Pretraining," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1–7

[6] H. He, R. Zhao, J. Li, L. Lu and Y. Gong, "Exploring Pre-Training with Alignments for RNN Transducer Based End-to-End Speech Recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 7079-7083

[7] "Deep Learning Enabled Semantic Communications With Speech Recognition and Synthesis," IEEE Trans. Wireless Commun., vol. 22, no. 3, pp. 1234–1245, Mar. 2023

[8] G. Saon, Z. Tuske, D. Bolanos and B. Kingsbury, "Advancing RNN Transducer Technology for Speech Recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Canada, 2021, pp. 5654-5658

[9] A. Sharma and B. Kumar, "Minutes of Meeting Generation for Online Meetings Using NLP & ML Techniques," in Proc. IEEE Conf. on Intelligent Systems, 2023, pp. 1–6.

[10] X. Shi, F. Yu, Y. Lu, Y. Liang, Q. Feng, D. Wang, Y. Qian, L. Xie, "The Accented English Speech Recognition Challenge 2020: Open Datasets, Tracks, Baselines, Results And Methods," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2021..

[11] P. Nascimento, J. C. Ferreira, and F. Batista, "Automatic Transcription System for Parliamentary Debates in the Context of the Assembly of the Republic of Portugal," International Journal of Speech Technology, vol. 27, pp. 613–635, 2024..

[12] L. Zhang et al., "Dialogue Acts Enhanced Extract–Abstract Framework for Meeting Summarization," Information Processing & Management, vol. 60, no. 1, 2023, Art. no. 103372.

[13] S. Ali et al., "Meeting the Challenge: A Benchmark Corpus for Automated Urdu Meeting Summarization," Information Processing & Management, vol. 60, no. 2, 2024, Art. no. 103694.

[14] M. Johnson et al., "Natural Language Processing Techniques Applied to the Electronic Health Record: A Case Study," Computer Methods and Programs in Biomedicine, vol. 230, 2025, Art. no. 107158.

[15] J. Zhang et al., "Speech Dereverberation with Frequency Domain Autoregressive Modeling," IEEE/ACM Trans. Audio, Speech, and Language Process., vol. 32, pp. 123–135, 2024.

[16] M. Auli, A. Baevski, H. Zhou, and A. Mohamed, “wav2vec: Unsupervised Pre-training for Speech Recognition,” in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2019, pp. 1–6.

[17] N. Kanda, G. Ye, Y. Wu, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, “Large-Scale Pre-Training of End-to-End Multi-Talker ASR for Meeting Transcription with Single Distant Microphone,” in *arXiv preprint*, 2021.

[18] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, “Unsupervised Cross-lingual Representation Learning for Speech Recognition,” in *Proc. Interspeech*, 2021, pp. 2426–2430.

[19] S. Cornell, M. Omologo, S. Squartini, and E. Vincent, “Detecting and Counting Overlapping Speakers in Distant Speech Scenarios,” in *Proc. Interspeech*, 2020, pp. 3107–3111

[20] G Mehendale, C Kale, P Khatri, H Goswami, “Multilingual Meeting Management with NLP: Automated Minutes, Transcription, and Translation”, Proceeding of International Conference on Communication and Intelligent Systems, Springer, 2023, pp.309-323.

[21] A. Silnova, N. Brümmer, J. Rohdin, T. Stafylakis, and L. Burget, “Probabilistic Embeddings for Speaker Diarization,” in *Odyssey Workshop*, 2020.