International Journal of Cognitive Computing in Engineering 6 (2025) 201–237

Contents lists available at ScienceDirect

International Journal of Cognitive Computing in Engineering

journal homepage: http://www.keaipublishing.com/en/journals/international-journal-of-cognitive-computing-in-engineering/

Automatic Speech Recognition: A survey of deep learning techniques and approaches

Harsh Ahlawat *, Naveen Aggarwal, Deepti Gupta

University Institute of Engineering and Technology, Panjab University, Chandigarh, India

ARTICLE INFO

Keywords:
Automatic Speech Recognition
Deep Neural Networks
Conformer
Transformer
Datasets
Multilingual
Deep learning

ABSTRACT

Significant research has been conducted during the last decade on the application of machine learning for speech processing, particularly speech recognition. However, in recent years, deep learning models have shown promising results for different speech related applications. With the emergence of end-to-end models, deep learning has revolutionized the field of Automatic Speech Recognition (ASR). A recent surge in transfer learning-based models and attention-based approaches on large datasets has further given an impetus to ASR. This paper provides a thorough review of the numerous studies conducted since 2010, as well as an extensive comparison of the state-of-the-art methods that are now being used in this research area, with a special focus on the numerous deep learning models, along with an analysis of contemporary approaches for both monolingual and multilingual models. Deep learning approaches are data dependent and their accuracy varies on different datasets. In this paper, we have also analyzed the various models on publicly accessible speech datasets to understand model performance across diverse datasets for practical deployment. This study also highlights the research findings and challenges with way forward that may be used as a beginning point for academicians interested in open-source Automatic Speech Recognition (ASR) research, particularly focusing on mitigating data dependency and generalizability across low resource languages, speaker variability, and noise conditions.

1. Introduction

Recently, the speech recognition community has made great progress toward building Deep Neural Networks (DNNs) for speech recognition by utilizing enormous amounts of training data and high-quality test sets (Ghoshal, Swietojanski, & Renals, 2013; Veselỳ, Karafiát, Grézl, Janda, & Egorova, 2012). While high-resource languages like French, English and Mandarin have benefited from newly created technologies (Huang, Li, Yu, Deng, & Gong, 2013; Toshniwal et al., 2018), a large number of the world's languages still lack a significant amount of training data or even modest test sets. Surprisingly, many languages lack standardized orthographies. With over 7100 languages (Lewis, 2009) in the world, one of the most pressing concerns for the speech and language community is to swiftly design and implement speech processing systems in unsupported languages at an acceptable cost and to bridge the gap between linguistic and technological expertise.

Human accents and speech begin to vary considerably after a few miles, and this slight change in speech characteristics is one of the most critical obstacles in constructing an intelligent speech recognition system. Over the past three decades, speech recognition systems have made significant strides in various areas such as speaker identification, diarization, and emotion recognition (Singh, Puri, Aggarwal, & Gupta, 2020; Singh, Singh, & Aggarwal, 2018a, 2018b; Singh, Singh, Aggarwal, Singh, & Singla, 2021). Recently, there has been a surge in the development of multilingual speech and language technologies, that have gained popularity as an effective method of expanding ASR coverage for every language across the globe. Multilingual systems have demonstrated a dominance over the monolingual systems with the help of shared learning of model components in several languages, particularly for those languages with limited data. Furthermore, by supporting multiple languages with just a single speech model instead of various distinct models, multilingual systems considerably simplify infrastructure. However, knowledge distillation (Cui et al., 2017), layered bottleneck features (Cui et al., 2015; Sercu et al., 2017; Thomas, Ganapathy, & Hermansky, 2012; Tüske, Pinto, Willett, & Schlüter, 2013), multitask learning (Chen & Mak, 2015) and shared hidden layers (Ghoshal et al., 2013; Heigold et al., 2013), are a few of the successful ways for developing multilingual acoustic models. Although only the acoustic model is multilingual in most state-of-the-art multilingual systems, separate language-specific models are still necessary. Multilingual End-2-End (E2E) models have recently acquired popularity because they combine language, acoustic, and pronunciation models from multiple languages into a single model that outperforms

* Corresponding author.
E-mail addresses: ahlawatharsh24@gmail.com (H. Ahlawat), navagg@gmail.com (N. Aggarwal), deeptigupta@pu.ac.in (D. Gupta).

https://doi.org/10.1016/j.ijcce.2024.12.007
Received 2 March 2024; Received in revised form 16 December 2024; Accepted 20 December 2024
Available online 6 January 2025

monolingual models (Cho et al., 2018; Karafiát et al., 2018; Toshniwal et al., 2018; Watanabe, Hori, & Hershey, 2017). These E2E multilingual models are producing excellent results, but it remains to be seen whether these models can compete with existing state-of-the-art traditional models while performing within real-time constraints.

Numerous surveys have been conducted over the last few decades to analyze and evaluate various aspects of ASR models developed over time. In a recent survey (Karmakar, Teng, & Lu, 2021), authors examined the various attention models that were utilized in the development of ASR models. The survey examines how Recurrent Neural Networks (RNNs) and Transformer architectures are used to develop and evolve attention models for offline and streaming speech recognition, and how self-attention could be used to replace the need for recurrence in ASR models, making them more effective and efficient. The authors in Aldarmaki, Ullah, Ram, and Zaki (2022) reviewed numerous deep learning models and identified strategies that could lead to fully unsupervised ASR. By unsupervised ASR authors refer to the problem of generating text transcriptions from raw speech input that does not have any form of manual labeling generated by humans, such as pre-transcribed spoken utterances and pronunciation dictionaries. The authors in Wali et al. (2022) analyzed various speech Generative Adversarial Networks (GANs) that influenced speech processing. The authors also addressed several application areas such as speech enhancement, speech synthesis, and data augmentation in ASR and emotion speech recognition systems. The authors also compiled a list of several data sets and evaluation metrics for speech GANs. The authors also explored several feature extractions, sub-word modeling, and segmentation strategies that can be employed with unsupervised ASR. In an another, paper (Alharbi et al., 2021) the authors evaluated various deep learning models based on DNNs that were introduced between the year 2015 and 2020. The authors also addressed several datasets, notably English datasets, that were used by researchers in the papers, that the authors evaluated. The authors also outlined several research gaps and gave future recommendations in the research domain of ASR. Another survey (Malik, Malik, Mehmood, & Makhdoom, 2021) conducted a detailed review of numerous feature extraction methodologies, classification models, and their impact on the performance of an ASR system. In addition, the survey covered numerous speech recognition resources such as datasets, toolkits, and language models, in brief, all of which can aid in the development of an ASR. In Harish and Rangan (2020) a variety of methodologies and approaches for processing Indian regional languages were discussed. The authors reviewed various aspects such as sentiment analysis, named entity recognition, machine translation, and parts of speech tagging using Rule, Statistical, and Neural Nets based techniques. Various dataset sources for Indian languages were also discussed. Another survey paper (Nassif, Shahin, Attili, Azzeh, & Shaalan, 2019) addressed several RNN and CNN-based hybrid and stand-alone DNN models. The authors also looked at some of the most common feature extraction and classification approaches, as well as ASR evaluation metrics. Paper (Singh, Kadyan, Kumar, & Bassan, 2020) reviewed the various feature extraction strategies as well as many classification methods for Indian languages in depth. Several Indian language datasets and resources were also thoroughly examined by the authors. The authors of Karita et al. (2019) compared and evaluated Transformer and conventional RNNs on a total of fifteen ASR benchmarks, including two Text-to-Speech (TTS) benchmarks, one multilingual ASR benchmark, and one speech translation benchmark. The authors discovered a variety of training tips and significant performance increases while employing Transformer for every benchmark, including Transformer's astonishing dominance against RNN in 13/15 ASR benchmarks. In another survey (Singh, Fayjie, & Kachari, 2015), the authors reported that the most common technique used for speech classification was the Hidden Markov Model (HMM) and Mel Frequency Cepstral Coefficients (MFCC) was commonly used for extraction of speech features. Furthermore, ASR systems implemented with Hidden Markov Toolkit (HTK) were more efficient than ASR implemented with other toolkits. Paper (Gaikwad, Gawali, & Yannawar, 2010) examined numerous feature extraction techniques in conjunction with classification models. Additionally, the authors discussed speech analysis methodologies and various types of speech, as well as their effect on the performance of speech recognition systems. Similarly, Besacier, Barnard, Karpov, and Schultz (2014) concentrated entirely on ASR for underserved languages. The authors gave definitions for under-resourced languages and explained why their preservation is critical. The methods used to collect data for under-resourced languages, as well as the basic framework of an ASR for an under-resourced language, were also reviewed. Similarly, Trentin and Gori (2001) provided a thorough explanation of various hybrid HMM/ANN based ASRs. Another research work, Lekshmi and Sherly (2016) discussed various types of ASRs as well as neural network-based speech recognition methodologies, whereas (Singh, Nath, & Kumar, 2018; Vadwala, Suthar, Karmakar, Pandya, & Patel, 2017) provided a review of different ASRs and various approaches which helps in recognizing speech. The authors reviewed previous literature as well and discussed several types of speech recognition techniques. In Latif et al. (2023) the authors presented an extensive survey conducted within various sub-fields of speech recognition. The primary focus of the authors was directed towards exploring various types of transformers employed in ASR, while concurrently identifying the array of challenges and issues that necessitate careful consideration when working with transformer models. The authors in Bhable, Deshmukh, and Kayte (2023) provide an overview of distinct speech technologies. Moreover, potential challenges, prospects, and methodologies specifically in the context of Indian languages were also discussed. The study also incorporates a presentation of different datasets and toolkits utilized for ASR system development and training. The paper (Prabhavalkar, Hori, Sainath, Schlüter, & Watanabe, 2023) presents an extensive survey on various End-to-End models implemented in the development of speech recognition systems. The authors primarily concentrated on the different types of End-to-End models, including their architecture, taxonomy, and relationship with conventional Hidden Markov Model (HMM)-based ASR systems. The authors tried to cover all pertinent aspects of End-to-End (E2E) ASR, ranging from modeling, encoding and decoding, model training and integration of external language models. Additionally, performance evaluation, deployment opportunities, and potential future developments were also discussed by the authors. A systematic review of recent advancements in speech recognition from 2015 to 2021 (Dhanjal & Singh, 2023) and from 2010–2022 (Mehrish, Majumder, Bharadwaj, Mihalcea, & Poria, 2023) were discussed by authors, with a focus on neural network-based approaches, datasets, toolkits, and evaluation metrics. They also integrated findings from previous studies to propose practical solutions for improving accuracy.

Table 1 summarizes the key highlights from the past survey in a concise and easy-to-understand format, and Table 2 highlights the same with reference to deep learning. The column's fields highlight critical aspects either included or omitted in previous survey papers.

Many prior surveys have overlooked the inclusion of diverse deep learning-based methodologies, such as "Transformers", "Conformers", and "Language Models". These approaches play a crucial role in advancing the ASR models, yet only a handful of surveys recognizing their significance. Further, there is a need to provide a comparison of several online toolkits, datasets, metrics and language models which can help in better training and evaluation of different ASR models.

This survey paper seeks to highlight the improvements in ASR over the last decade, as well as the analysis of contemporary approaches and models for both monolingual and multilingual languages with the following goals:

- To comprehend the core architecture of an ASR system and explore the impact of alternative methods used across different phases on overall performance.
- To analyze the various deep learning models that are currently being employed in ASR development.

Table 1
Key outlines and limitations of the past surveys for speech recognition.

Ref No.	Year	Key Outline	Discussion of various steps			Discussion of various speech resources		
			Feature Extraction	Classification	Deep Learning methods	Evaluation Metrics	Toolkits and Datasets	Language Models
Trentin and Gori (2001)	2001	Discussed Hybrid HMM/ANN ASRs.	✗	✓	✗	✓	✗	✗
Gaikwad et al. (2010)	2010	Discussed different speech recognition techniques.	✓	✓	✗	✓	✗	✗
Besacier et al. (2014)	2013	Discussed ASRs for under-resourced languages.	✓	✓	✗	✓	✓	✗
Padmanabhan and Johnson Premkumar (2015)	2015	Discussed Hybrid HMM/ANN and DNN/HMM ASRs.	✗	✓	✓	✗	✗	✗
Singh et al. (2015)	2015	HMM and MFCC were discussed and their implementation using HTK.	✓	✓	✗	✗	✗	✗
Lekshmi and Sherly (2016)	2016	Discussed different types of neural nets used for ASRs.	✗	✓	✓	✗	✗	✗
Vadwala et al. (2017), Singh, Nath, and Kumar (2018)	2017, 2018	Overview on ASR and past literature.	✗	✗	✗	✗	✗	✗

- To highlight various online toolkits, datasets, and language models used in the development of ASR models.
- To discuss in detail recent state-of-the-art approaches for low-resourced languages in a multilingual context.

The remainder of the paper is structured in the following manner. Section 2 outlines the purpose and methodology of this research study. Section 3 summaries an ASR system. Section 4 discusses the information regarding several databases that have been used in the development of ASR models in past decades. It also discusses whether or not their distribution is open source or protected by a license. Section 5 describes the metrics that are widely used to measure the results of various ASR models on diverse datasets. Section 6 discusses several online speech recognition toolkits, and Section 7 discusses various language models which help with the development of modern ASR systems. Section 8 reviews a few state-of-art models that have previously been employed and are currently used with monolingual and multilingual ASR systems. Section 9 discusses some of the recent transformer based ASR models recently being used in the ASR development. Section 10 discusses the computational, ethical, and environmental considerations that need to be addressed during the development of ASR systems. Section 11 discusses the impact of reinforcement learning on the development and advancement of ASR systems. Section 12 concludes our survey findings, while Section 13 addresses challenges and outlines future directions in the field of ASR. A few tables to provide a brief overview of the various

methodologies employed by various researchers and to demonstrate the best model accuracy over a certain database as comparisons are also included.

2. Nature of study

The goal of the Research Methodology is to investigate, identify, and evaluate articles that use DNNs in the domain of speech recognition. The survey in this research is based on a Systematic Literature Review (SLR) which follows general criteria proposed by Kitchenham and Brereton (2013). The following are the essential steps in designing this SLR:

- Formulating research questions in order to craft SLR.
- Describing the research strategy used to obtain relevant research papers.
- Defining an appropriate study selection criterion, including inclusion/exclusion criteria.
- Conducting a comprehensive relevant literature search and analysis by doing a backward and forward citation check.

2.1. Objective of study

Main objective of the paper is to present a comprehensive review and detailed analysis of recent speech recognition model. For effective analysis, we have addressed the key questions such as the types of studies reviewed, datasets used for training and testing different DNN models, and the languages associated with these datasets. The paper also examines evaluation metrics for ASR models and the language models employed. Additionally, it explores the application of deep learning models in low-resource languages and discusses future challenges and directions. This review aims to enhance understanding and development of deep learning-based speech recognition technologies and serve as a starting point for new researchers in the field.

2.2. Research questions

The key objective of this survey paper is to identify and evaluate articles that use DNNs in the domain of speech recognition. The following research questions were identified as a result of this:

- RQ1: What were the various types of papers included in the study?
- RQ2: Which deep neural network (DNN) models were used?
- RQ3: Which datasets were utilized for testing and training of models in each paper?
- RQ4: What were the various datasets languages identified in the published papers?
- RQ5: Which evaluation metrics were employed in the research papers?
- RQ6: Which language models were employed in the research papers?
- RQ7: Which Deep learning models are currently being employed with low-resourced languages?
- RQ8: Potential future challenges and directions?

2.3. Search strategy

The objective is to list search keywords to determine relevant literature. The keywords are searched in the title, abstracts, and metadata of research publications. The research questions were utilized to determine the main search words. “Speech Recognition”, “Automatic Speech Recognition”, “Speech Recognition Using Deep Neural Networks”, “Monolingual Automatic Speech Recognition”, “Multilingual Speech Recognition”, “Speech Recognition Using Transformers”,

Table 2
Key outlines and limitations of the past surveys using deep learning for speech recognition.

Ref No.	Year	Key Outline	Discussion of various steps			Discussion of various speech resources		
			Feature Extraction	Classification	Reviewed Trans-former/RNN	Evaluation Metrics	Toolkits and Datasets	Language Models
Karita et al. (2019)	2019	Analyzed Transformer and RNN models side by side on various speech datasets.	✓	✓	✓	✗	✓	✗
Singh, Kadyan, et al. (2020)	2019	Analyzed in detail the different feature extraction techniques and multiple classification methods for Indian languages.	✓	✓	✗	✗	✓	✗
Nassif et al. (2019)	2019	Analyzed in detail the different deep learning models.	✓	✓	✗	✓	✗	✗
Harish and Rangan (2020)	2020	Analyzed in detail the different Rule, Statistical and Neural based approaches for Indian languages.	✓	✓	✗	✓	✓	✓
Malik et al. (2021)	2020	In depth review of ASR along with different feature extraction techniques and multiple classification methods.	✓	✓	✗	✓	✓	✓
Karmakar et al. (2021)	2021	Analyzed in detail transformer and RNN based methods for offline and streaming ASR.	✓	✓	✓	✗	✓	✗
Alharbi et al. (2021)	2021	Evaluated numerous deep learning models based on DNNs that were introduced between 2015 and 2020.	✓	✓	✓	✗	✗	✗
Aldarmaki et al. (2022)	2022	Reviewed numerous deep learning models and identified strategies that could lead to fully unsupervised ASR.	✓	✓	✓	✓	✗	✗
Wali et al. (2022)	2022	Analyzed various speech GANs and addressed several application areas such as speech enhancement, speech synthesis, and data augmentation in ASR.	✓	✓	✓	✓	✓	✗
Yadav and Sitaram (2022)	2022	Reviewed various multilingual ASR models trained on multiple languages along with cross-lingual transfer learning setting.	✓	✓	✓	✗	✗	✗
Latif et al. (2023)	2023	A comprehensive and in-depth survey of transformer applications in the audio domain was provided.	✓	✓	✓	✗	✓	✗
Bhable et al. (2023)	2023	Analyzed various ASR techniques and approaches for multilingual Indian languages	✓	✓	✓	✗	✓	✗
Kaur, Singh, Sachdeva, and Kukreja (2023)	2023	Discussed ASR techniques across major languages, highlighting scarcity of ASR systems for minority languages.	✓	✓	✓	✗	✓	✗
Prabhavalkar et al. (2023)	2023	An in-depth review of end-to-end models and methodologies currently being employed in ASR was presented.	✓	✓	✓	✗	✗	✗
Mehrish et al. (2023)	2023	Presented an in-depth review of deep learning ASR models, different speech-processing tasks, datasets, and benchmarks employed in evaluation of ASR.	✓	✓	✓	✗	✓	✗
Dhanjal and Singh (2023)	2023	A comprehensive review of recent DNN ASR models. Datasets and metrics employed in ASR were also discussed.	✓	✓	✓	✓	✓	✗

"Speech Recognition Using Conformers", and "Indian Language Automatic Speech Recognition". were among the search queries used. To find and access papers related to this research topic *ACM Digital Library, IEEE Xplore, Science Direct, Google Scholar, Springer Link*, and *arxiv.org* were explored and the distribution of conference and journal research papers obtained are summarized in Fig. 1 as well. The general words including associated synonyms related to the domain were used to identify the majority of the relevant research seed words. Afterwards, distinct seed words gathered from a variety of sources were used to make sure all keywords were included in the names of research publications and articles. The use of AND operator ensured that almost all selected terms were included in the titles.

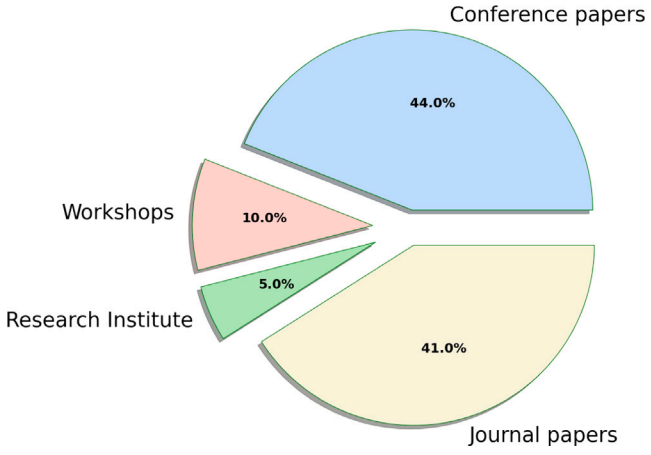

Fig. 1. Percentage of the paper type studied.

2.4. Study selection

Initially, 264 papers were acquired based on the search conducted using the search keywords. Further screening was carried out to ensure that only relevant publications were included in this review, and the findings were discussed in regularly scheduled meetings. The following are the selection and filtration steps that were used:

- Remove all duplicate research publications received from various digital libraries.
- Using inclusion/exclusion criteria to assure that only relevant publications are included.

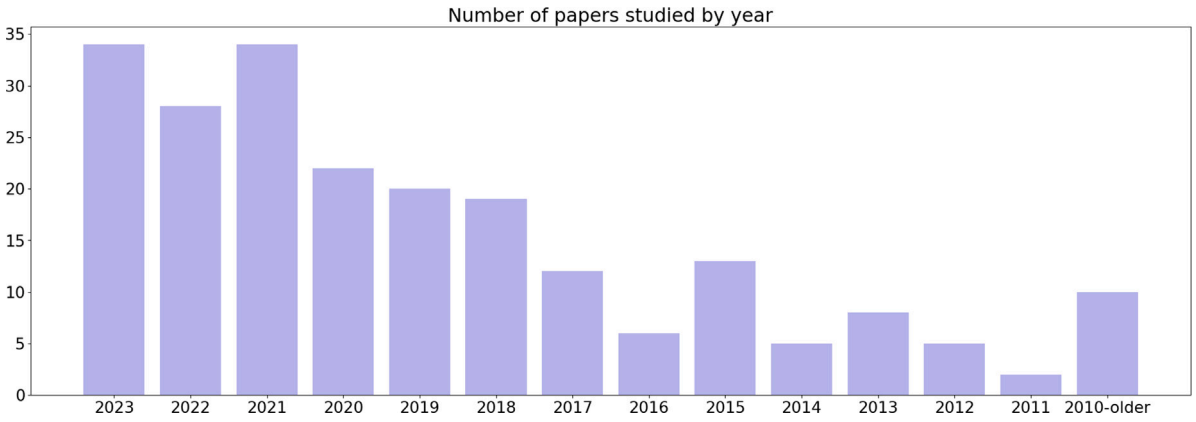

Fig. 2. A year-by-year breakdown of the number of papers studied.

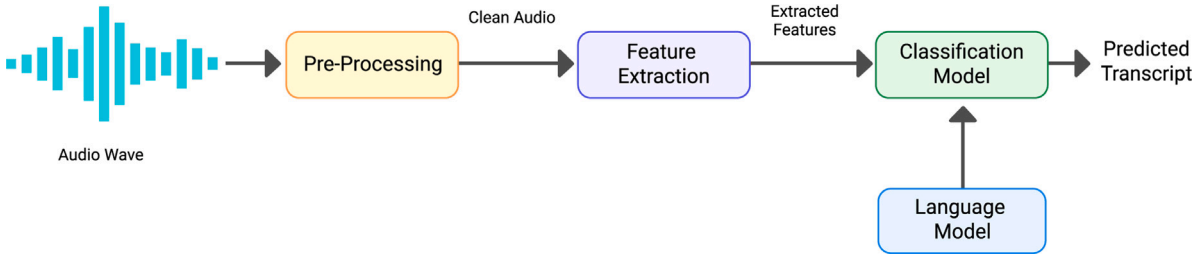

Fig. 3. Architecture of a traditional ASR (Papastratis, 2021).

The following are the inclusion/exclusion criteria followed in this review paper:

Inclusion criteria: Papers were collected from the year 2010 till January 2024. All papers retrieved that use DNNs and deep learning in the domain of speech are included.

Exclusion criteria: Papers which used DNNs in areas other than speech were exempted. Short papers were excluded because the majority of them constituted preliminary work. Papers with no apparent publication information were also excluded.

Methods such as citation chaining and reference mining were also utilized to broaden the scope of the research. By evaluating cited publications, it assisted in the retrieval of more relevant papers connected to our domain. The number of papers studied year wise is shown in Fig. 2.

3. Overview of Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR) refers to the process of automatically recognizing and translating a human voice into text format. This research area has drawn a lot of attention for decades. It is currently a critical area of research for human–machine communication. Manual feature extraction and standard approaches such as Gaussian Mixture Models (GMM), Support Vector Machines (SVMs), Dynamic Time Warping (DTW), and HMM have been used in the early stages. These architectures were best suited for solving basic or confined problems, as their constraints might create complications in large-scale, complex real-world problems. Recently, models such as RNNs and CNNs, as well as Transformers, were applied to ASR with considerable success.

The basic objective of an ASR system is to convert an input acoustic signal $x = (x_1, x_2, \ldots, x_T)$ of length T into a series of words or characters $y = (y_1, y_2, \ldots, y_T)$, $y_n \epsilon V$ where V is the vocabulary. The labels could be letters, words, or sentences. Fig. 3 shows an overall flow of a classical ASR system.

3.1. Processing steps

The following are the processing steps in a generic ASR system:

- Pre-Processing: — The purpose of the pre-processing stage is to enhance the audio signal by lowering the signal-to-noise ratio, eliminating the noise, and filtering the signal in order to increase the quality of the signal.
- Feature Extraction: — ASR features are retrieved using a set of values or coefficients derived from the input using different algorithms. This method must be dependable in terms of a variety of quality criteria, including noise and the echo effect. The majority of ASR systems use the following techniques to extract features: (a) Feature extraction using MFCC. (b) Discrete Wavelet Transform (DWT).
- Classification: — This step seeks to detect or locate the spoken text in the input signal. The output text is generated depending on the features retrieved during the preprocessing step.
- Language model (LM): — This module maintains language's grammar rules or semantic information. Language models are necessary to recognize the output of a classification model and apply alterations to the output text.

3.2. Practical implementation

The following are the steps to get input features and target labels while working with ASR systems.

- The initial step is to load audio files of spoken speech into the system.
- Convert to uniform dimensions: The audio data clips could contain a lot of variety. Clips may contain a distinct number of channels and varying sampling rates or can certainly be of varying lengths. As a result, the size of each audio clip will vary. Due to the fact that deep learning models require all input clips to be the same size, various data cleaning operations are required

to normalize the dimensions of audio data which significantly improves the performance of models by creating tensors of fixed shapes. Re-sampling the audio ensures the same sampling rate for each clip. Each item must be converted to almost the same number of channels and audio length. It is accomplished through the padding of smaller sequences and the truncation of longer ones. When working with models that consider the elastic nature of speech, such as RNN, audio clips of varying lengths can also be used. Whereas if audio quality is bad, it can be enhanced by employing a noise-removal technique that suppresses background noise.

- Enhancement of raw audio data: — Several data augmentation methods can be employed to diversify the input data and aid the model in learning to generalize to a larger range of inputs. This can be performed by randomly moving the audio right or left by a fine margin, or by slightly altering the pitch or tempo of the audio.
- Mel Spectrograms: — This stage converts original audio to Mel Spectrograms. A spectrogram shows how audio is made by breaking it down into the different frequencies that make it up.
- Mel Frequency Cepstral Coefficients: — When analyzing human speech, it might be helpful to do an additional step in which the Mel Spectrogram is converted to MFCC. MFCC produces a condensed version of the Mel Spectrogram via extracting just the most significant frequency coefficients that correspond to human voice frequency ranges.
- Data Augmentation of Spectrograms: — This process involves artificially expanding the dataset size by applying transformations, such as SpecAugment, to mel spectrogram images. This method involves Time and Frequency masking, which randomly mask horizontal (frequency mask) or vertical (time mask) bands of information from the spectrogram. It helps in improving the generalization ability of machine learning models by exposing them to a broader range of variations in the input data.

Following data cleaning and augmentation, the initial raw audio file is converted to Mel Spectrogram pictures in order to create the transcript's target labels. Given that the transcript consists of plain text comprising sentences and words, it is essential to extract a vocabulary out of each character in the transcript, which is then translated into character IDs. This gives us both the input and target labels. Following that, the data is ready to be loaded into the deep learning model.

Several preprocessing techniques and data augmentation strategies are used to enhance performance of speech recognition models. Feature normalization, MFCC feature selection improves the model convergence and generalization. Using Audio data as spectrograms with frequency masking, and time masking enhances model robustness to noise. Spectrogram can be considered as images and can be enhanced using image augmentation approaches. Similarly in time domain, Speed perturbation randomly adjusts speech speed, improving the model's ability to adapt to variable speaking rates. Adding background noise during training also helps the model adapt to noisy environments by mixing clean speech with various noise types and levels. These techniques collectively contribute to improved model robustness and generalization capabilities in speech processing tasks.

Fig. 4 presents a timeline from 2010 to 2023, which saw some notable improvements in speech recognition models (discussed in the following sections) ranging from hybrid neural network architectures to more end-to-end models, seq-to-seq encoder–decoder models with self-attention mechanism, transducer, transformer, and conformer-based speech recognition.

4. Datasets

For effective training of ASR models, researchers have used different monolingual and multilingual datasets, which have also been released in the public domain. This section discusses in detail a few of the most frequently used open-source and licensed datasets. Table 3 lists the monolingual speech datasets that are available, whereas Table 4 lists the multilingual speech datasets in a concise and convenient manner. The license type, total duration (in hours), and speech languages, as well as the best accuracy model of various datasets, are also listed.

4.1. LibriSpeech

The LibriSpeech corpus (Panayotov, Chen, Povey, & Khudanpur, 2015) contains around 1000 h of English speech that has been properly segmented and aligned from read audiobooks from the LibriVox project. The majority of the audiobooks come from Project Gutenberg. Vassil Panayotov prepares it with the assistance of Daniel Povey. The training data for LibriSpeech is broken down into three different sets: 100 h, 360 h, and 500 h. The development and test data sets, on the other hand, are broken down into 'clean' and 'other' sets. There are about 5 h of audio in each set of dev and test tools. There are 803 million tokens and 977 thousand unique words in this corpus, as well as n-gram language models with associated texts which are all excerpted from Project Gutenberg books.
Dataset link: — https://www.openslr.org/12

4.2. Libri-light

Another dataset in which the audio files are collected from open-source read audiobooks made accessible by the LibriVox project. Libri-Light (Kahn et al., 2020) is a spoken English audio library that may be used to train speech recognition systems in an unsupervised manner. It is one of the largest freely-available corpus of speech, with over 60K hours of audio recordings. A small training set (10 h, 1 h, or 10 m) of labeled speech with very little supervision is also provided. Voice activity detection was used to separate the audio, and it is labeled with Signal-to-Noise Ratio (SNR), speaker ID, and genre descriptions.
Dataset link: — https://github.com/facebookresearch/libri-light

4.3. TIMIT

The TIMIT (Linguistic Data Consortium, 2022) is an acoustic-phonetic continuous speech corpus that provides speech data for acoustic-phonetic research, as well as the creation and testing of ASR systems. This dataset consists of recordings of 630 speakers in eight accents of American English, where each of them is reading 10 phonetically rich sentences. Thirty per cent of the speakers are female, while the remainder are male. A 3.14 h of recording time is allocated to the training set; the remainder is shared between the test and development sets. It also includes time-aligned orthographic transcriptions of speech at the word and phone levels. The corpus was created in partnership with Texas Instruments, Inc. (TI), Massachusetts Institute of Technology (MIT), and SRI International (SRI), with recordings made at TI and transcriptions made at MIT. To ensure phonetic and dialectal coverage, the TIMIT corpus transcriptions were thoroughly checked.
Dataset link: — https://catalog.ldc.upenn.edu/LDC93S1

4.4. Switchboard

The Switchboard (Linguistic Data Consortium, 2022) telephone speech corpus (LDC97S62) consists of about 260 h of telephone conversations in English between 543 speakers (302 men and 241 women) from all around the United States. It was initially recorded by Texas Instruments through DARPA sponsorship in 1990–1. In 1992–3, the Linguistic Data Consortium (LDC) distributed the initial version of this corpus, which was published by NIST.
Dataset link: — https://catalog.ldc.upenn.edu/LDC97S62

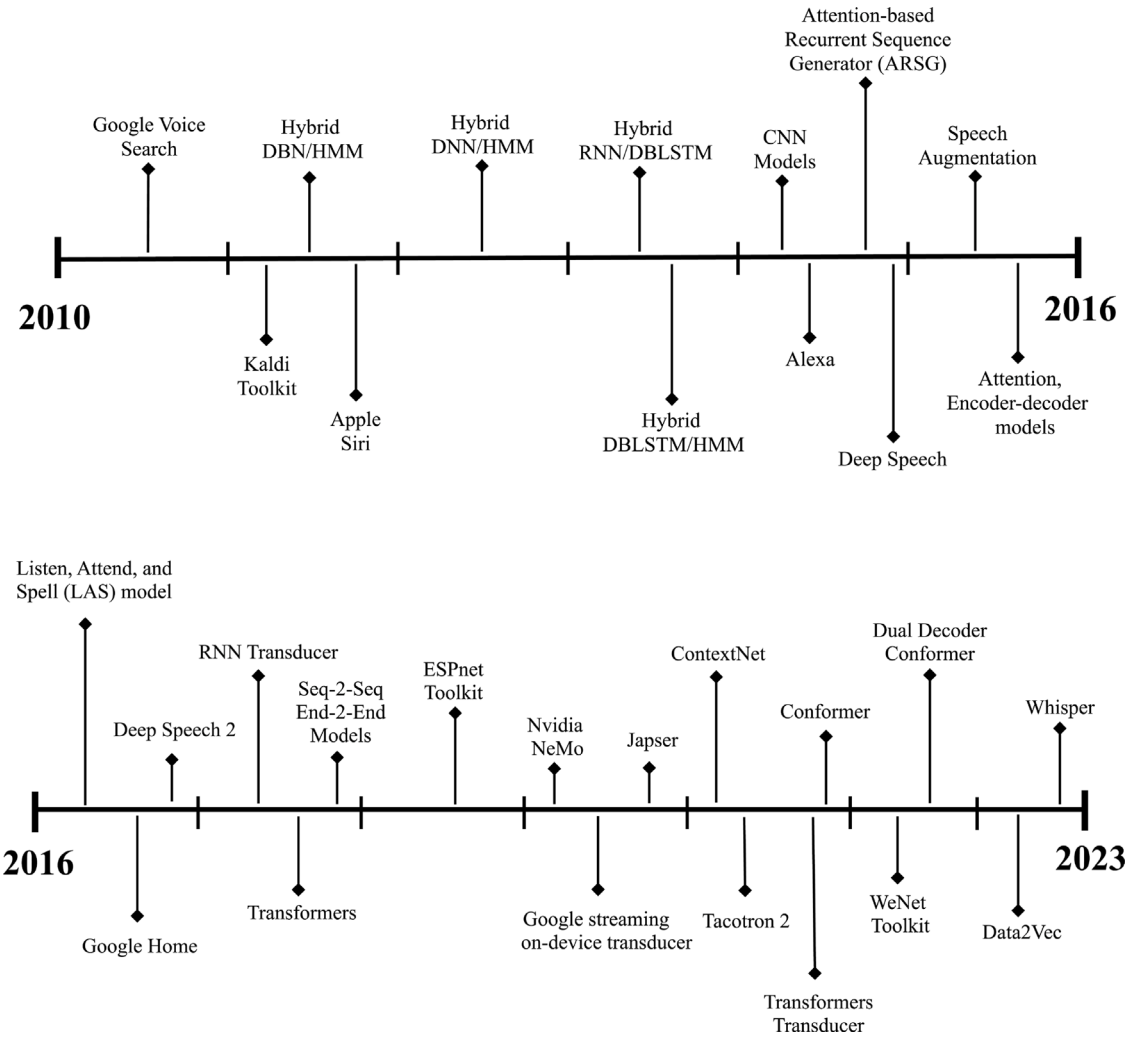

Fig. 4. A timeline of some of the significant advances in speech recognition from 2010 to 2023.

Table 3
List of monolingual datasets available for speech recognition.

Name	License Type	Language	Hours/Sentence pairs (Approx)	Accuracy (year)
LibriSpeech	Open Source	English	1000 h	WER 1.4 (2020)
Libri-light	Open Source	English	60K h	WER 2.5 (2020)
TIMIT	Licensed by LDC	English	420 h	PE 8.3 (2020)
Switchboard	Licensed by LDC	English	260 h	PE[1] 4.3 (2021)
2000 HUB5	Licensed by LDC	English	11 h	
CHiME-5	Licensed	English	50.12 h	WER 31.9 (2021)
TED-LIUM	CC BY-NC-ND 3.0	English	452 h	WER 3.94 (2023)
GigaSpeech	Open Source	English	40K h	WER 10.9 (2021)
CSTR VCTK	Open Source	English	9 h	No proper benchmarks
SPGISpeech	Proprietary	English	5000 h	WER 3.11 (2023)
Libri-trans	Open Source	English	236 h	BLEU[2] 16.3 (2020)
Aishell-1	Open Source	Mandarin	170 h	WER 1.29 (2023)
Aishell-2	CC BY-NC-ND	Mandarin	1000 h	WER 2.85 (2023)
Fisher-CallHome	Licensed by LDC	Spanish	160 h	WER 9.6 (2016) WER 9.1 (2017)
How2	Open Source	English	2000 h	WER 13 (2021)

[1] PE: Percentage Error (Switchboard + Hub5 combined).
[2] BLEU (case sensitive sacre).

4.5. 2000 HUB5

The NIST-sponsored 2000 HUB5 (Linguistic Data Consortium, 2022) evaluation series focused on conversational speech over the phone, with the specific objective of transcribing conversational voice into text. The LDC then created the 2000 HUB5 Evaluation Transcripts in English, which contains transcripts of 40 English telephone conversations. Its objectives were to investigate interesting new areas in conversational speech recognition, to develop advanced technologies incorporating those concepts, and to assess the performance of such technology.
Dataset link: — https://catalog.ldc.upenn.edu/LDC2002S09

Researchers combined the Switchboard and 2000 HUB5 datasets for the development and training of various new ASR models, providing them with about 300 h of English speech data. This merged dataset

Table 4
List of multilingual datasets available for speech recognition.

Sr No.	Name	License Type	Languages	Hours/Sentence pairs (Approx)
1	MuST-C	CC BY-NC-ND 4.0	15 Languages	385-504 h
2	Europarl	Unknown	21 Languages	24M sentences
3	MediaSpeech	Open Source	4 Languages	10 h
4	CoVoST	CC0	16 Languages	2880 h
5	IWSLT TED	CC BY-NC-ND 4.0	109 Languages	Variable Dataset Size
6	VoxPopuli	CC0	23 Languages	400K h
7	Common Voice[1]	CC BY-SA 3.0	112 Languages	18.6K h
8	LibriVoxDeEn	CC BY-NC-SA 4.0	De-En	100 h
9	Microsoft Speech	C-UDA 1.0	Gujarati, Telugu, Tamil	Insufficient information
10	IIT Bombay	CC BY-NC 4.0	En-Hi	1.66M segments
11	MultiIndicMT	Open Source	11 Languages	11M sentence pair
12	IndicCorp	CC BY-NC 4.0	12 Languages	9B tokens
13	Dhwani	MIT License	40 Languages	17k h
14	MUCS 2021	Various licenses	6 Languages	145 h
15	Shrutilipi	CC0	12 Languages	6457 h
16	Vistaar	MIT License	12 Languages	10736 h
17	NITK-IISc	Open Source	6 Languages	Insufficient information
18	Voxforge	GNU General Public License	6 Languages	Insufficient information
19	Tatoeba	Various CC licenses	Over 100 Languages	Variable Dataset Size
20	FLEURS	CC BY 4.0	102 Languages	1.4k h
21	Multilingual LibriSpeech	CC BY 4.0	8 Languages	44.5k h (Eng), 6k h (other)

[1] Dataset constantly being updated and information provided above is up-to date until Dec 2023.

includes variants such as Switchboard (300hr), Switchboard Hub5'00 full/swb-hub500-full, and Switchboard + Hub500. A small easy subset of this combined dataset called as Switchboard + Hub500 was taken by a few where they train on Switchboard conversational telephone speech and perform testing on Hub5'00. Others, on the other hand, used the entire dataset, i.e. Switchboard Hub5'00 full, for testing and training of their ASR model and reported varying results.

4.6. CHiME-5

The CHiME-5 (Barker, Watanabe, Vincent, & Trmal, 2018) dataset was created to help with the development of a reliable speech recognition system. It is made up of 50.12 h of recorded discussions in actual homes. The dataset's training set, development set and testing set of this dataset consist of around 40.33 h of data with nearly 80,000 utterances, 4.27 h of data with over 7000 utterances and 5.12 h of data with 11,000 utterances respectively.
Dataset link: — http://spandh.dcs.shef.ac.uk/chime_challenge/CHiME5/data.html

4.7. Common voice

Mozilla's Common Voice is an open-source initiative in which users can lend their voices to read a certain text or can offer their time to assess if a given audio recording fits its transcription. They have collected 18.6k h of data from various languages so far, of which 14.1k h have been validated and currently offering speech datasets in 112 languages. This dataset is constantly being updated, and the information provided above is accurate until May 2022.
Dataset link: — https://commonvoice.mozilla.org/en/datasets

4.8. TED-LIUM

The TED-LIUM Corpus (Hernandez, Nguyen, Ghannay, Tomashenko, & Esteve, 2018) is a free to use English speech corpus derived from audio briefings and interview transcripts that were made accessible on the TED website. It contains 452 h of TED speeches and subsequent transcriptions. Out of which 316 h comprises of male recordings and 134 h is of female with an average duration of 11 m. There are 2028 unique speakers in the corpus with 268k sentence segments and 4.9M words.
Dataset link: — https://www.openslr.org/51/

4.9. GigaSpeech

GigaSpeech (Chen et al., 2021) contains around 40k h of audio transcribed from many sources, including YouTube, audiobooks and podcasts, making it a multi-domain English speech corpus. A total of 40k hours of audio is provided, including 10k h of high-quality labeled audio suitable for supervised training, making it appropriate for semi-supervised and unsupervised training. GigaSpeech provides 5 subsets of various sizes: 10 h, 250 h, 1000 h, 2.5k h, and 10k h for system training.
Dataset link: — https://github.com/SpeechColab/GigaSpeech

4.10. CSTR VCTK corpus

In this dataset (Veaux, Yamagishi, & MacDonald, 2017) 400 newspaper sentences were picked and recited by a total of 110 distinct people. All of the speakers were native English speakers, ranging in accent, age and gender. Approximately 9 h of audio data is included in this dataset.
Dataset link: — https://datashare.ed.ac.uk/handle/10283/2651.

4.11. SPGISpeech

SPGISpeech (O'Neill et al., 2021) is a large-scale transcription dataset that is open to academic researchers. It contains 5k h of recorded and transcribed corporate earnings calls. The original calls were divided into slices ranging from 5 to 15 s in duration to facilitate speech recognition system training. SPGISpeech comprises a diverse range of worldwide speakers (around 50k), one of the greatest numbers of any speech corpus, and provides a variety of L1 and L2 English accents. Each file is a 16 kHz, 16-bit audio file with a single channel. Each transcript has been double-checked for accuracy and presented correctly.
Dataset link: — https://datasets.kensho.com/datasets/spgispeech

4.12. Libri-trans

Libri-trans is another corpus that was created using read audiobooks from LibriVox, and it was carefully divided and automatically aligned e-books in French with LibriSpeech utterances in English. This resulted in 236 h of English speech that was utterance-by-utterance synchronized with French translations.
Dataset link: — https://github.com/alicank/Translation-Augmented-LibriSpeech-Corpus

4.13. LibriVoxDeEn

LibriVoxDeEn (Beilharz, Sun, Karimova, & Riezler, 2019) is a speech translation corpus that makes it simple to train E2E speech translation systems from German to English. This corpus consists of triplets of German audio and text, as well as English translations based on German audiobooks. This corpus is comprised of more than 100 h of audio and more than 50k parallel sentences. A manual evaluation was used to check the audio quality and sentence alignment.

Dataset link: — https://heidata.uni-heidelberg.de/dataset.xhtml?persistentId=doi:10.11588/data/TMEDTX

4.14. Aishell-1

Aishell (Bu, Du, Na, Wu, & Zheng, 2017) is one of the largest datasets used for speech recognition research and system development in Mandarin. The dataset consists of 170 h Mandarin speech data which includes audio from 400 distinct speakers of various genders and ages. The recordings were made in a calm indoor setting with a high-fidelity microphone and were down-sampled to 16 kHz. The dataset was made robust by including speech on a variety of themes, including business, science and technology, entertainment, and sports.

Dataset link: — https://www.openslr.org/33/

4.15. Fisher-CallHome

Fisher-CallHome (Linguistic Data Consortium, 2022) was made up of transcribed phone calls between Spanish speakers (mainly native) who spoke in different dialects. The Fisher Spanish data set includes 819 transcribed talks with 1.5 million tokens and approximately 160 h of speech aligned at the utterance level. The CALLHOME Spanish corpus has 200k tokens and 120 transcripts i.e. about 20 h of speech aligned at the utterance level.

Dataset link: — https://catalog.ldc.upenn.edu/LDC2014T23

4.16. How2

The How2 (Sanabria et al., 2018; Yu, Jiang, & Hauptmann, 2014) corpus contains around 80,000 instructional videos (almost 2000 h with an average length of 1 m 30 s) with English subtitles and summaries. About 300 h have also been translated into Portuguese using crowd-sourcing.

Dataset link: — https://github.com/srvk/how2-dataset

4.17. MuST-C

MuST-C (Cattoni, Di Gangi, Bentivogli, Negri, & Turchi, 2021) is a multilingual corpus that facilitates the training of E2E speech translation systems from English to 14 additional languages (Spanish, German, Turkish, Italian, Persian, Russian, French, Romanian, Portuguese, Czech, Dutch, Arabic, Vietnamese, Chinese). MuST-C consists of 385 h (varying from 385 to 504 h) of speech recordings from English TED Talks that have been automatically matched at the sentence level with their translations and manual transcriptions in each target language. The 40-channel log-mel filterbank coefficients were used to pre-process speech features of this corpus.

Dataset link: — https://ict.fbk.eu/must-c/

4.18. Europarl

The Europarl (Iranzo-Sánchez et al., 2020) Corpus contains the European Parliament's proceedings from 1996 to 2012. It featured 11 official languages (Swedish, Dutch, German, Italian, English, French, Greek, Portuguese, Danish, Spanish, and Finnish) of the European Union (EU) when it first came out in 2001. As a result of the EU's political expansion, the corpus data has been expanded to include the national languages of the 10 new EU member states. The most current version (2012) contains up to 60M words for each language, with newly added languages severely under-represented owing to data availability beginning in 2007. There are now 21 European languages in the new corpus: Portuguese, Czech, Danish, Estonian, Slovak, French, Latvian, German, English, Greek, Dutch, Hungarian, Finnish, Italian, Bulgarian, Lithuanian, Spanish, Polish, Romanian, Slovene, and Swedish.

Dataset link: — https://www.statmt.org/europarl/

4.19. MediaSpeech

MediaSpeech (Kolobov et al., 2021) is a media speech dataset created to evaluate the performance of ASR systems. The dataset is made up of brief speech segments that were automatically retrieved from YouTube videos and hand transcribed after some pre-and post-processing. For each language, the dataset contains 10 h of speech. The current release of the dataset is a part of a bigger private dataset that includes audio files in French, Arabic, Turkish, and Spanish. Each audio recording was converted to WAV files encoded in single-channel 16 kHz 16-bit PCM.

Dataset link: — https://github.com/NTRLab/MediaSpeech

4.20. CoVoST

CoVoST (Wang, Pino, Wu & Gu, 2020; Wang, Wu, & Pino, 2020) is a multilingual, large-scale ST corpus based on Mozilla's Common Voice initiative. Its most recent version includes Arabic, Catalan, Estonian, German, Indonesian, Tamil, Latvian, Turkish, Mongolian, Persian, Welsh, Slovenian, Japanese, Swedish, and Chinese translations from English. It has a total of 2880 h. of speech audio and 78K different speakers.

Dataset link: — https://github.com/facebookresearch/covost.

4.21. IWSLT TED corpus

WIT3 (Web Inventory of Transcribed and Translated Talks) (Cettolo, Girardi, & Federico, 2012) provides the IWSLT TED and MT Experiment Corpus available for research purposes on multilingual transcriptions of TED talks. TED talks are accessible in video format with English translations and subtitles in more than a hundred other languages. More information about the dataset is available on the WIT3 website.

Dataset link: — https://wit3.fbk.eu/home

4.22. VoxPopuli

VoxPopuli (Wang et al., 2021) is a multilingual speech corpus for semi-supervised learning, unsupervised representation learning and interpretation. It contains around 400K h of unlabeled speech data in approximately 23 languages, including 1800 h of transcribed speech in fifteen languages and their associated oral interpretations into fifteen target languages, all totaling around 17300 h.

Dataset link: — https://github.com/facebookresearch/voxpopuli

4.23. IIT Bombay English-Hindi corpus

The dataset (Anoop, Pratik, Pushpak, et al., 2018) contains a parallel English-Hindi corpus in addition to a monolingual Hindi corpus gathered from a variety of current corpora and sources developed over the years at IIT Bombay's Center for Indian Language Technology.

Dataset link: — https://www.cfilt.iitb.ac.in/~parallelcorp/iitb_en_hi_parallel/

4.24. Microsoft speech corpus (Indian languages)

This corpus release mainly includes 3 Indian languages (Tamil, Telugu, Gujarati) phrasal and conversational speech for test and training data. The audio and transcripts are also included in the dataset. The dataset's data may not be utilized for commercial purposes and can only be used for research purposes as stated by Microsoft.
Dataset link: — https://msropendata.com/datasets/7230b4b1-912d-400e-be58-f84e0512985e

4.25. MultiIndicMT

This corpus generally contains ten Indian languages (Hindi, Punjabi, Malayalam, Telugu, Bengali, Gujarati, Oriya, Tamil, Kannada, and Marathi) as well as English. The training corpus contains over 11M sentence pairs in English and Indian languages.
Dataset link: — http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/index.html

4.26. IndicCorp

IndicCorp (Kakwani et al., 2020) is a large monolingual corpus consisting of 9 billion tokens and comprising of twelve Indian languages (Assamese, Marathi, Oriya, Punjabi, Gujarati, Hindi, Kannada, Malayalam, Bengali, English, Tamil, Telugu). The corpus was created over several months by locating and scraping thousands of websites, mainly news, magazines, and books. It includes a single huge text file with one sentence per line. The version made accessible is randomly shuffled, reduplicated and untokenized.
Dataset link: — https://ai4bharat.iitm.ac.in/corpora

4.27. IndicNLP corpus

The IndicNLP (Kunchukuttan et al., 2020) corpus contains 2.7 billion words from ten Indian languages (Marathi, Oriya, Punjabi, Gujarati, Hindi, Kannada, Malayalam, Bengali, Tamil, Telugu) and is a large-scale general domain dataset. The sentences were collected from the domains of business, entertainment, lifestyle, sports, politics, and technology.
Dataset link: — https://ai4bharat.iitm.ac.in/datasets

4.28. Dhwani

Dhwani is a large and diverse ASR corpus compiled from two major sources: YouTube and News On AIR news broadcasts covering a wide range of topics such as finance, education, news, and technology. The dataset includes 17000 h of raw speech in 40 distinct Indian languages.
Dataset link: — https://ai4bharat.iitm.ac.in/dhwani

4.29. MUCS 2021

The MUCS 2021 (Diwan et al., 2021) challenge dataset contains testing and training data for 6 Indian languages (Hindi, Telugu, Odiya, Gujarati, Marathi, and Tamil). Audio files in Telugu, Gujarati and Tamil were sampled at 16 KHz, but audio files in Hindi, Odiya and Marathi were sampled at 8 KHz each having 40 h of training and 5 h of testing, whereas Hindi, Marathi, and Odiya each have 95 h of training and 5 h of testing. Additionally, the dataset includes two code-switched language pairings (Hindi–English and Bengali–English) with durations of around 90 and 46 h, respectively. The dataset spans across a variety of domains, including general topics, stories, agriculture, finance, and healthcare.
Dataset link: — https://navana-tech.github.io/MUCS2021/data.html

4.30. Shrutilipi

Shrutilipi (Bhogale, Raman, et al., 2023) is a large labeled and diverse ASR corpus, sourced from news headlines aired on All India Radio. The dataset consists of 6457 h of transcribed speech obtained by extracting audio and text pairing in 12 Indian languages (Marathi, Oriya, Punjabi, Gujarati, Sanskrit, Hindi, Kannada, Urdu, Malayalam, Bengali, Tamil, Telugu).
Dataset link: — https://ai4bharat.iitm.ac.in/shrutilipi/

4.31. Vistaar

Vistaar (Bhogale, Sundaresan, et al., 2023) is a comprehensive collection of 59 benchmarks covering diverse language and domain combinations. Vistaar-Trainset brings together 13 publically available datasets in Indian languages, including popular ones such as Common Voice, Shrutilipi, NPTEL, MUCS, IIT Bombay, Google TTS, and Vakyasancayah. This massive dataset contains 10736 h of transcribed speech obtained by extracting audio and text pairing in 12 Indian languages (Marathi, Oriya, Punjabi, Gujarati, Sanskrit, Hindi, Kannada, Urdu, Malayalam, Bengali, Tamil, Telugu). Dataset link: — https://github.com/AI4Bharat/vistaar

4.32. NITK-IISc

This is a multilingual and multi-accent speaker corpus (Kalluri et al., 2021) that includes five Indian languages (Telugu, Malayalam, Tamil, Hindi, Kannada) and Indian English. Speech data from 345 bilingual speakers in India are included in the sample. Each speaker has contributed approximately 4-5 min of data, which comprises recordings in both English and their native language. The dataset also includes speaker metadata such as L1, medium of instruction, their native location, and current residence for each speaker. Furthermore, the corpus also includes the physical characteristics of the speakers such as their weight, age, height and shoulder size.
Dataset link: — https://github.com/iiscleap/NISP-Dataset

Apart from these, some more open-source multilingual dataset sources include **FLEURS** (Conneau et al., 2023), **Multilingual LibriSpeech (MLS)** (Pratap, Xu, Sriram, Synnaeve, & Collobert, 2020), **Voxforge**, which was created to gather transcribed speech which can be used with open source and free speech recognition Engines and is accessible through http://www.repository.voxforge1.org/downloads/SpeechCorpus/Trunk/. **Tatoeba**, is another massive collection of phrases, translations, sentences and spoken audio that may be utilized for language learning purposes. This dataset contains community-generated recordings of spoken English, which can be accessed at https://tatoeba.org/en/downloads while the Indian Language Technology Proliferation And Deployment Center provides speech data in a variety of Indian languages for research purposes and is accessible at https://tdil-dc.in/index.php?lang=en

5. ASR accuracy evaluation metrics

Various research publications have used a variety of evaluation approaches to analyze the ASR system's overall performance. To determine the accuracy of an ASR, the following approaches can be used:

Word Error Rate (WER) (Morris, Maier, & Green, 2004) is drawn using the Levenshtein distance and is a frequently used metric for estimating the accuracy of machine translation or speech recognition systems since it estimates the error rate at the word level instead of the phoneme level. The value of the WER may be calculated by using the equation below:

$$\text{WER} = \frac{S + D + I}{N} \tag{1}$$

Where S is the total number of substitutions within output text, D denotes the total number of deletions, I denotes the total number of insertions, and N denotes the overall number of words within the source.

The **Word Recognition Rate (WRR)** (Morris et al., 2004), sometimes known as the Word Accuracy Rate, is a variant of the WER which can be used to measure the efficiency of an ASR. The following equations can be used to compute it:

$$\begin{aligned} \text{WRR} &= 1 - \text{WER} \\ &= \frac{N - S - D - I}{N} \\ &= \frac{H - I}{N} \end{aligned} \tag{2}$$

Where $H = N - (S + D)$ is the total number of words correctly predicted.

The **Phone error rate (PER)** (Morris et al., 2004) is computed by dividing the total number of phonemes by the number of phoneme errors (inserted (I), deleted (D), and modified phonemes (M)).

$$\text{PER} = \frac{I + D + M}{N} \tag{3}$$

The **Character error rate (CER)** (Morris et al., 2004) is another metric for assessing the accuracy of an ASR system. It is comparable to WER, except that it measures character errors rather than words.

$$\begin{aligned} \text{CER} &= \frac{S + D + I}{N} \\ &= \frac{S + D + I}{S + D + C} \end{aligned} \tag{4}$$

where S denotes substitutions, I denotes insertions, D denotes deletions, C denotes correct characters, and N = total number of characters in the source ($N = S+D+C$). CER's output is not always a value between zero and 1, especially when a large amount of insertions occurs. This value is mostly related to the percentage of characters that were anticipated inaccurately. The lower the value, the more efficient the ASR system is, with a CER of zero being the optimal score.

Translation Error Rate (TER) (Post, 2018; Snover, Dorr, Schwartz, Micciulla, & Makhoul, 2006) computes the amount of post-editing needed after machine translation tasks. This automated metric counts the number of steps needed to change a translated segment in accordance with one of the reference translations. It is simple to use, is language-independent and correlates to post-editing work.

Frame Error Rate (FER) in speech recognition, represents the model's accuracy at the individual frame level. Each feature in speech recognition is calculated by a specific model over a set time interval known as the frame duration. The recognizer then analyzes these features to generate the desired output. The frame duration typically lasts for a very short period, for example, 10 ms. Furthermore, features calculated over several seconds are aggregated into a window, which may have a duration of 25 ms. The FER evaluates the percentage of frames that are incorrectly predicted by the recognizer.

$$\text{FER} = \text{FF} - \text{FM} \tag{5}$$

FF represents the percentage of frames where a voice is detected when there is none, whereas FM represents the percentage of frames in which a voice is present but not detected.

MAE (Mean Absolute Error) and **RMSE (Root Mean Squared Error** are metrics used to measure the discrepancy (in duration) between predicted and true sequences. They help quantify the average deviations between predictions and true values. Lower MAE and RMSE values indicate a smaller difference between predicted and actual values, signifying more accurate predictions.

BLEU (Bilingual Evaluation Understudy) (Papineni, Roukos, Ward, & Zhu, 2002) is a method for determining the efficiency with which a machine can translate a reference text between two natural languages. It was one of the first measures to demonstrate a substantial correlation between human quality judgements. BLEU always returns a value between zero and 1. This number indicates the degree to which the candidate text is comparable to the reference texts, with larger values indicating greater similarity. A machine translation is better if it is as close as possible to a quality human translation. Only a small percentage of human translations are given a score of 1, indicating that the candidate text is exactly the same as one of the reference translations. As a result, obtaining a score of 1 is not required. Additional reference translations will improve the BLEU score since there are more possibilities to match. Despite the fact that BLEU provides significant advantages, it has been proposed that an increase in BLEU score does not always imply an improvement in translation quality.

BERTScore (Zhang, Kishore, Wu, Weinberger, & Artzi, 2019) is a machine learning algorithm that leverages BERT's pre-trained contextual embeddings to detect words in candidate and reference sentences based on cosine similarity among them. It has been demonstrated that it correlates with human judgement on both the sentence and system levels. Additionally, BERTScore calculates accuracy, recall, the F1 measure and precision, which may be useful for evaluating a variety of language generation tasks.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004)is a software package and collection of metrics for analyzing NLP technologies such as machine translation and automated summarization. The metric compares an autonomously generated translation or summary against a human-authored summary or translation to a collection of references. ROUGE is case insensitive, which means that it treats uppercase and lowercase letters equally.

The **General Language Understanding Evaluation (GLUE)** (Napoles, Sakaguchi, Post, & Tetreault, 2016) measure is a BLEU variant devised for assessing grammatical error corrections based on n-gram overlap with a group of reference sentences instead of recall or precision of individual annotated errors. Its evaluation is more similar to human assessments than those provided by metrics such as I-measure and MaxMatch. The current version of GLUE is the second iteration of the metric, which has been updated to solve concerns that develop as a result of the increasing number of reference sets being used. The updated metric does not require any tuning and should be used in place of the original metric wherever possible.

IndicGLUE (Kakwani et al., 2020) is an assessing criterion of natural language understanding for the Indian languages. It can do a wide range of operations and is available in 11 main Indian languages, including Assamese, Marathi, Oriya, Punjabi, Gujarati, Hindi, Kannada, Malayalam, Bengali, Tamil, and Telugu.

In a recent work on ASR evaluation in healthcare, researchers introduced **Clinical BERTScore (CBERTScore)** (Shor et al., 2023), a metric prioritizing clinically relevant accuracy over other metrics like WER, METEOR and BLUE scores. Supported by the Clinician Transcript Preference (CTP) dataset, comprising 149 medical sentences evaluated by 18 clinicians, their research aims to improve ASR metrics tailored to clinical needs, addressing a crucial gap in evaluation within medical contexts.

In ASR, various combinations of metrics are used to evaluate the performance of different deep learning techniques and models. Certain measures evaluate systems better depending on the language dataset for which they are utilized. Common combinations include WER, CER and PER. Additionally, traditional classification metrics like accuracy, precision, and recall evaluate ASR systems at the word or phoneme level. Language model metrics such as perplexity or cross-entropy can be used to evaluate the quality of language models used alongside ASR systems. End-to-end ASR systems may employ specific metrics like Connectionist Temporal Classification (CTC) loss, attention alignment loss, or Sentence Error Rate (SER). Robustness metrics like Signal-to-Noise Ratio (SNR) and Signal-to-Interference Ratio (SIR) evaluate ASR system resilience to environmental factors such as background noise or reverberation.

Table 5
Toolkits available for Automatic Speech Recognition.

Toolkit Name	Open-Source	Description	Programming Language	Operating System	Language
Kaldi	Yes	Neural Net based	C++	Cross-Platform	English
PyTorch-Kaldi	Yes	Neural Net based	Python, Perl	Cross-Platform	English
HTK	No	HMM Neural Net based	C	Cross-Platform	English
Julius	Yes	HMM Trigrams based	C	Cross-Platform	Multilingual
ESPnet	Yes	Hybrid CTC/attention E2E Neural Net based	Python	Cross-Platform	Multilingual
NeMo	Yes	E2E Neural Net based	Python	Cross-Platform	Multilingual
WeNet	Yes	E2E Neural Net based	C++, Python	Cross-Platform	Multilingual
CAT	Yes	E2E Neural Net based	Python, C++	Cross-Platform	Multilingual
Sphinx	Yes	HMM based	Java	Cross-Platform	Multilingual
RWTH	No	Hybrid HMM/Neural Net based	C++	Linux, macOS	English

6. ASR toolkits

Significant amount of time and work has been expended on improving the process of speech recognition. Several online resources and toolkits have been made publicly available throughout the last decade for the effective processing of speech with ease. These toolkits provide an optimized implementation of state-of-the-art algorithms and are very popular among application developers. Open-source toolkits can be retrained in languages other than English from the scratch. However, if the toolkits are proprietary and only available in English, retraining in languages other than English is usually not feasible. In a concise and simple manner, Table 5 summarizes the key features of the various toolkits covered in the preceding paragraph.

Kaldi (Povey et al., 2011) is a free and open-source voice recognition application programmed in C++ which is distributed under the Apache License v2.0 and supports cross-platform. This toolkit initially supported the English language, but recognizers for a wide range of languages can be built depending on the availability of speech data in the respective language. In addition to performing classification tasks with DNNs, Kaldi can perform feature extraction as well. Multiple techniques can be used to extract the features, including the most frequently used MFCC, i-vectors and Cepstral Mean and Variance Normalisation (CMVN). Kaldi was named after the legendary Ethiopian goat herder Kaldi, who is associated with discovering the coffee plant.

PyTorch-Kaldi (Ravanelli, Parcollet, & Bengio, 2019) is another open-source toolkit for developing DNN/HMM-based ASR systems. It fills the gap between the Kaldi and PyTorch toolkits, attempting to inherit Kaldi's efficiency and PyTorch's flexibility. PyTorch manages the DNN component, while the Kaldi toolkit handles feature extraction, decoding and label computation. It supports a number of pre-implemented models, including RNN, MLP, LSTM, CNN, GRU, Li-GRU, SincNet, and multi-GPU training.

The **Hidden Markov Model Toolkit (HTK)** (Zhao, Wakita, & Zhuang, 1991) is a proprietary software toolkit that was primarily used to manipulate and generate HMMs. Speech recognition was the primary focus of this toolkit's development, but it can also be applied to other pattern recognition tasks that require HMMs, such as speech synthesis, character recognition, and DNA sequencing. HTK, which was created at the Cambridge University Engineering Department's Machine Intelligence Laboratory is employed widely by researchers working on HMMs. Recently researchers shifted their focus towards Kaldi, as it appears to be the most popular option nowadays in place of HTK.

Julius (Lee, Kawahara, & Shikano, 2001) is an openly accessible speech recognition toolkit which was initially developed for Japanese recognition. Additionally, with the assistance of the VoxForge project, a feasible model for the English language was built over time. Using Julius, it is possible to build recognizers for a wide variety of languages that has both a language model and an auditory model. Julius makes use of a pronunciation dictionary and acoustic models in a format similar to HTK, and ARPA standard word 3-gram language models.

Another open-source speech recognition toolkit developed at Carnegie Mellon University is **CMU Sphinx** (Lamere et al., 2003). This Java-based toolkit may provide pre-trained models for a variety of languages, including English, German, Mandarin, Russian, and French. Sphinx extracts features with MFCC and performs classification with an HMM-based model and include an online tool for creating language models. Sphinx involves a series of speech recognizers (Sphinx 2 - 4) which are developed and modified over the years and an acoustic model trainer (SphinxTrain). Sphinx now supports the following programming languages: Python, C, Ruby, Java, C++, JavaScript and C#.

RWTH ASR (short RASR) (Rybach et al., 2009) is another proprietary voice recognition toolkit released under the "RWTH ASR License", which is based on the Q Public License (QPL). Recipes for word lattice processing, speaker adaptive training, discriminative training, and unsupervised training are all available through RWTH ASR, as well as tools for developing acoustic models and decoders. It extracts features using MFCC and Perceptual Linear Prediction (PLP). GMM is used to perform acoustic modeling. The fact that this toolkit is only accessible for Linux and macOS is a major limiting factor.

ESPnet (Watanabe et al., 2018) is a hybrid CTC/attention-based E2E speech processing toolkit that focuses primarily on E2E speech recognition and text-to-speech conversion. As its core deep learning engine, ESPnet uses Pytorch and chainer and utilizes Kaldi-based feature extraction and data processing format. ESPnet also provides recipes for a full framework for ASR and other speech processing research, such as text-to-speech, speech enhancement, machine translation, and voice conversion. Unlike ESPnet1, the current version ESPnet2 does not rely on Kaldi/Chainer.

Nvidia NeMo (Kuchaiev et al., 2019) is an open-source toolkit enabling developers to create and train cutting-edge conversational AI models. The fundamental goal of NeMo is to assist researchers from industry and academia in reusing earlier work (code to pre-trained models) in order to make the development of new conversational AI models faster. NeMo contains domain-specific collections for ASR, NLP, and TTS that may be used to build cutting-edge models like Citrinet, Jasper, BERT, Fastpitch, and HiFiGAN. Each collection is composed of prebuilt modules which include all the components necessary for data training. Each module is simply customizable, extendable, and compositable in order to construct new conversational AI systems. It takes a lot of data and computation power to train conversational AI models. NeMo makes use of PyTorch Lightning to facilitate and accelerate mixed-precision training over several GPUs and nodes.

WeNet (Yao et al., 2021) is an open-source transformer-based production-ready speech recognition toolkit. WeNet differentiates itself from other open-source E2E speech recognition toolkits by facilitating the deployment of ASR applications in a number of real-world contexts. With its revolutionary two-pass method, it attempted to incorporate non-streaming and streaming E2E speech recognition into a single system. Additionally, WeNet demonstrated a dynamic chunk-based attention approach, the same as the transformer layers for allowing arbitrary changes in the right context length within a hybrid CTC/attention architecture.

CAT (An, Xiang, & Ou, 2020) is an open-source ASR toolkit that provides an E2E methodology for CTC-CRF based speech recognition. It inherits the hybrid approach's data efficiency and the E2E approach's flexibility, giving a full-fledged application of CTC-CRFs as well as

comprehensive testing and training scripts for a variety of Chinese and English benchmarks. On a finite dataset, CAT outperforms previous non-modularized E2E models, showing its data efficiency.

7. Language models

Language Models are generally used to predict the next word or character in a document. It is essentially a probability distribution word sequences. In practice, a language model indicates the likelihood with which a given word sequence is "valid", however, the term "validity" here does not refer to grammatical validity at all. It means that it is similar to how people speak (or, more precisely, write), which is what the language model learns. In contrast to RNNs and CNNs, the standard **Transformer** based language model (Vaswani et al., 2017) architecture uses an attention mechanism rather than using a recurrence to extract global dependencies between input and output, also transformer based language model can obtain considerably higher parallelization than RNNs and CNNs (Zhao et al., 2023).

Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018) improves on standard Transformers by discarding the unidirectionality constraint via a pre-training objective of a Masked Language Model (MLM). BERT utilizes a next sentence prediction task in conjunction with the MLM to jointly pre-train text-pair representations. BERT consists of 2 steps: pre-training and fine-tuning. The pre-training stage involves training on unlabeled data utilizing a range of pre-training tasks. After the BERT model has been initialized with the pre-trained parameters, it is fine-tuned on subsequent tasks using labeled data where each subsequent task has its own fine-tuned model.

ALBERT (Lan et al., 2019) is a BERT-based Transformer model but has far fewer parameters. It accomplishes this through the use of two-parameter reduction strategies. The first is a parametrization based on factorized embeddings. The hidden layers size is segmented from the vocabulary embedding by splitting the larger vocabulary matrix into two smaller matrices. This enables the hidden size to be increased without considerably increasing the parameter size of the vocabulary embeddings significantly. The second method is to share parameters across layers. This stops the parameter from increasing in size as the network depth increases. In addition, to deal with the problem of Sentence Order Prediction (SOP), ALBERT employs a self-supervised loss. SOP is primarily concerned with inter-sentence coherence and is intended to solve the inefficiency of the next sentence prediction (NSP) loss proposed in the original BERT.

DistilBERT (Sanh, Debut, Chaumond, & Wolf, 2019) is another BERT-based Transformer model which is small, fast and light. Knowledge distillation is employed while the pre-training phase to lower the size of a BERT model by 40 percent and speed it up by 60% while preserving 97 percent of its language understanding capabilities at the same time. It uses a triple loss function which combines cosine-distance, language modeling, and distillation losses to utilize the inductive biases learned by bigger models during pre-training.

RoBERTa (Liu et al., 2019) is a BERT extension with few modifications to the pre-training procedure. These changes include training the model for a longer duration, with additional data and in larger batches. Another is eliminating the BERTs objective of the next sentence prediction. The RoBERTa model can then be trained on longer sequences. In contrast to BERT, a dynamically changing masking pattern is applied to the training data in the case of RoBERTa.

GPT-3 (Brown et al., 2020) is an autoregressive transformer model with 175 billion parameters. It is built on the same architecture/model as GPT-2, with reversible tokenization, modified initialization, and pre-normalization. GPT-3, in contrast to GPT-2, makes use of the Sparse Transformer-style alternating dense and local banded sparse attention patterns in the transformer layers.

GPT-4 (OpenAI, 2023) is a large-scale pre-trained Transformer-based multimodal model that predicts the next token in a document using images and text. It falls short of human proficiency in some real-world situations, yet it performs comparably to humans on a number of professional and academic levels.

BART (Lewis et al., 2019) is a pre-training denoising autoencoder for seq-to-seq models. It is trained by first corrupting text using a random noise function and then developing a model to retrieve the original content. Its architecture is based on a standard Transformer and includes a left-to-right decoder (like GPT) and a bidirectional encoder (similar to BERT). This indicates that, like BERT, the encoder's attention mask is totally observable, whereas the decoder's attention mask is causal, as in GPT2.

XLNet (Yang et al., 2019) is an autoregressive Transformer that combines the best features of autoencoding and autoregressive language modeling while trying to address their shortcomings. Rather than employing a predefined backward or forward factorization order, as is the case with typical autoregressive models, XLNet maximizes the predicted log-likelihood of a sequence for all potential factorization orders. Additionally, XLNet combines the Transformer-XL relative encoding scheme and segment recurrence mechanism into pre-training, which results in improved performance, particularly for tasks that involve longer text sequences, as inspired by recent advances in autoregressive language modeling.

Without a language model, decoding audio is likely to produce spelling errors, this is due to the fact that the speech recognition system bases its predictions entirely on the audio input it receives, rather than the language modeling context of successive or previous anticipated letters. Spelling errors made by speech recognition systems can be greatly reduced by using a language model because a well-trained transformer-based or n-gram based language model will never predict a word with spelling errors.

8. Review of deep neural network models for ASR

The last decade has seen a significant improvement in Neural Network architecture used in the speech recognition domain. Different models have been used, including RNNs (Oruh, Viriri, & Adegun, 2022), CNNs (Li et al., 2022) and more recently, Transformer networks (Bain, Huh, Han, & Zisserman, 2023; Le et al., 2020; Wang, Mohamed, et al., 2020; Zeng, Li, & Liu, 2021) and Conformer networks (Guo, Chang, Watanabe, & Xie, 2021; Kim et al., 2022; Liu, Li, Zhang, & Yan, 2021) which have performed admirably (Papastratis, 2021). These Neural network architectures have undergone significant evolution over time, each bringing its own set of benefits and drawbacks (Qamar & Zardari, 2023). Let us delve into the details of their architecture.

8.1. Background study

Initial research in speech recognition systems faced a lot of hindrances due to large requirements of computation power and storage needs. Further non-availability of large datasets for different languages also slowed down the pace of research in this domain.

Since its introduction in 2010, the Google Voice Search app has revolutionized the field of speech recognition. Millions of people were able to utilize speech recognition as it was released as an app. Additionally, Google was gathering data from billions of searches, which could have assisted it in predicting what people were saying. Approximately 230 billion words derived from user searches were stored in Google's English Voice Search System during the time. This also paved the way for Siri, which arrived a year later.

In a difficult Large Vocabulary Continuous Speech Recognition (LVCSR) task, Dahl, Yu, Deng, and Acero (2011) demonstrated in 2011 that substituting more powerful models in place of GMMs can help in reducing recognition error. The authors presented the DBN-HMM, an HMM and a context-dependent Deep Belief Network (DBN) hybrid system that outperforms strong GMM-HMM baselines on a difficult LVCSR dataset which was derived from the Bing mobile voice search

task. After comparing their model to GMM-HMM trained models, they found their DBN-HMM system achieves a test set accuracy of 69.6%, an increase of 5.8% and 9.2% when trained with minimum PE rate and Maximum Likelihood (ML) respectively, over GMM-HMM-based state-of-the-art systems. As a result of this and other similar research, later researchers started working on a combination of HMM and neural network-based models.

Another research published in 2012, Hinton et al. (2012) showed that substituting GMM with a DNNs greatly increases the WER of an ASR system Authors claimed that DNNs trained using novel methods and with multiple hidden layers outperform GMMs on a number of speech recognition tasks, sometimes by a wide margin. On five distinct large vocabulary tasks, they compared the DNN-HMM model against the GMM-HMM-based model (YouTube, Switchboard, Bing Voice, English broadcast news and Google Voice). On every task, the models based on DNN-HMM outperformed the HMM-GMM model; even GMM baselines trained for more than 400 h had similar WER to DNN baselines trained for roughly 50 h. As a result, DNNs pre-trained as generative models leads to improved recognition results on the TIMIT dataset and, subsequently, on a range of LVCSR benchmarks. Additionally, the authors discovered that pre-training a DNNs system helps to avoid overfitting and minimizes the time needed for fine-tuning, resulting in a surge of interest in DNNs for acoustic modeling. The authors also highlighted that the major disadvantage of DNNs compared to GMMs was the computing power and cost required by DNNs, which makes it more difficult to train them on vast datasets using large cluster machines.

In another study (Deng, Hinton, & Kingsbury, 2013) by Hinton et al. published in 2013, the authors explained how DNNs transformed the development of the acoustic model during the previous year. The authors highlighted the reasons that contributed to the recent rise of NN as high-quality acoustic models which were as follows: (1) increasing the depth of networks increases their power (2) By appropriately initializing the weights and employing much faster hardware and higher GPU capacity, DNNs can be trained efficiently, and (3) by using more (context-dependent) output units, their performance can be improved significantly. Moreover, the authors also discussed various methods for optimizing DNNs, improved methods for speech pre-processing and how numerous DNN hyper-parameters can be determined, and methods for leveraging multiple dialects or languages that were more easily accomplished with DNN models rather than GMM models. As a result, the research community shifted its focus from GMM-HMM models to Deep Learning and Neural Networks (NN) (Chorowski, Bahdanau, Cho, & Bengio, 2014; Chorowski, Bahdanau, Serdyuk, Cho, & Bengio, 2015) for ASR development (Anastasopoulos et al., 2021; Ansari et al., 2020; Cettolo et al., 2012).

DNNs have emerged as powerful technique across multiple domains around that time, showcasing their versatility in applications such as speech recognition, image classification and natural language processing. One of their key strengths lies in feature learning, where they autonomously extract pertinent features from raw data, thus minimizing the necessity for manual feature extraction. Moreover, the scalability of DNNs over the years has significantly improved with advancements in both hardware and software frameworks, facilitating the training and deployment of large-scale networks. Notably, DNNs have consistently pushed the boundaries of performance across various domains, particularly with the evolution of deep learning techniques, often achieving state-of-the-art results. Their adaptability extends to tasks like classification, regression, and pattern recognition, with a capability to discern complex non-linear relationships within high-dimensional datasets. The accessibility of platforms like TensorFlow and PyTorch in recent years further democratizes the process of building and training DNNs, making them more widely applicable in artificial intelligence research and development. DNNs often require a large amount of labeled data for training, which can be time-consuming and expensive to acquire. Training these networks also requires powerful hardware like GPUs or TPUs, especially for bigger models. Overfitting is a common issue with DNNs, especially when there is not much training data, leading to compromised generalization performance on unseen data. Additionally, DNNs may encounter difficulties in effectively processing sequential data or capturing temporal dependencies. To address such issues, RNNs and CNNs were introduced, effectively handling challenges like these.

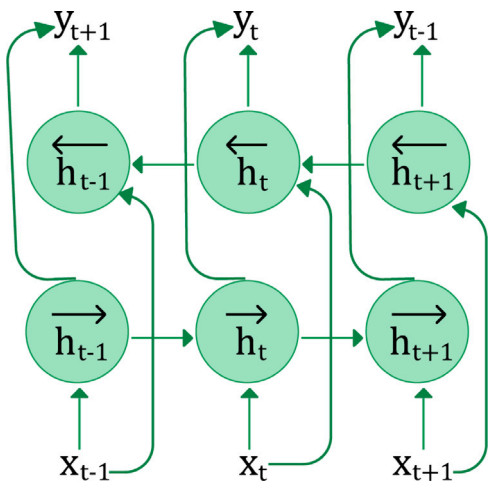

Fig. 5. Bidirectional Recurrent Neural Networks (BiRNN) (Papastratis, 2021).

8.2. Recurrent neural networks (RNN's)

RNNs perform calculations on the time sequence because their present hidden state is reliant on each of their past hidden states. RNNs are built to visualize time-series signals and to catch short and long-term dependencies between various input time-steps.

In terms of speech recognition, To compute the output sequence $y = (y_1, y_2, y_N)$ and hidden states $h = (h_1, h_2, h_N)$, the input signal $x = (x_1, x_2, x_T)$ is passed through the RNN. The simple RNN has one limitation that based on the previous context, the next output is generated. But, when it comes to speech recognition in most cases, knowledge about the future context is just as crucial as information about the past (Graves, Mohamed, & Hinton, 2013). Therefore, Bidirectional RNNs (BiRNNs) are often used in order to address this problem rather than using a unidirectional RNN. Both the backward and forward directions of the input vectors are processed by the BiRNNs, and the hidden state vectors for each direction are retained. Fig. 5 presents a general structure of BiRNNs.

RNN-Transducer (RNN-T) (Guo et al., 2020; Rao, Sak, & Prabhavalkar, 2017; Saon, Tüske, Bolanos, & Kingsbury, 2021) is a combination of one RNN with CTC and another RNN which tries to predict the next output depending on the previous one. It computes a distinct probability distribution $P(y_k \mid t, u)$ for each time step t and output time step u for each of the output kth elements. As illustrated in Fig. 6 RNN-T is composed of three networks: an encoder network, a joint network, and a prediction network. The encoder is a recurrent network that is similar to the acoustic model in a conventional ASR system, consisting of eight-stacked Long Short-Term Memory (LSTM) layers. It transforms the acoustic feature x_t at time-step t into the representation $he_t = f_{enc}(x_t)$. Two LSTM layers are used in the prediction network, followed by a joint model 640-dimensional projection layer. The prediction network accepts the previously assigned label y_{u-1} and produces a new representation $hp_t = f_p(y_{u-1})$. A joint network consists of fully connected layers that merge the two representations and give the posterior probability $P(y \mid t, u) = f_{joint}(he_t : hp_t)$.

As a result, the RNN-T can produce new words or symbols by combining data from the prediction network and the encoder, depending on whether the label predicted is blank or not. Whenever a blank label

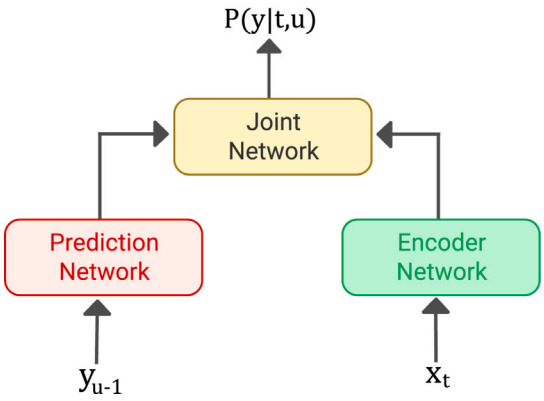

Fig. 6. General structure of RNN Transducer (Papastratis, 2021).

is generated during the final time step, the inference operation comes to a halt.

Connectionist Temporal Classification (CTC) loss is a function that determines how well the input speech signal and the output word sequence are aligned. A blank label is used by CTC to represent the silent time-step or to represent the transition between phonemes or words. The alignment path probability is computed as:

$$P\ (\alpha \mid x) = \Pi_{t=1}^{T}\ P\ (\alpha_t \mid x) \tag{6}$$

where x = input, y = output probability sequence of character or words at time-step t, and α_t is a single alignment.

There are multiple potential alignments for a particular transcription sequence because labels and blanks can be segregated in a variety of ways. As a result, the cumulative probability of all potential paths is computed as follows:

$$P\ (y \mid x) = \sum P\ (\alpha_t \mid x) \tag{7}$$

CTC optimizes the cumulative probability of accurate alignments to provide the right output word sequence. One significant advantage of CTC is that no prior data alignment or segmentation is required.

Over the past decade, several researchers have used RNN-based ASR models in a variety of ways. Graves, Mohamed, and Hinton (2013) discussed the usage of RNNs in ASR in their research work. They discovered that RNNs performed poorly in ASR when compared to DNNs, but when RNNs were trained alongside Deep Bi-directional LSTM (DBLSTM), they discovered that this combination outperformed DNNs on the TIMIT phoneme recognition benchmark, achieving 17.7% PER, which was superior to all state-of-the-art during that time. For training RNNs, the authors used two E2E training methods: (1) CTC and, (2) Sequence Transduction. Because the results of this work rely on RNNs' special objective function, they were difficult to combine with other existing large vocabulary speech recognition systems or with world level language models. Authors extended the results of this work in Graves, Jaitly, Mohamed et al. (2013), reporting that the DBLSTM and HMM hybrid system delivers equally good results on TIMIT as the prior work. On a portion of the WSJ corpus, their hybrid combination outperformed GMM and DNN benchmarks. However, the WER improvement over DNNs was not notable. As a result, the authors concluded that the hybrid technique using DBLSTM seems to be appropriate for acoustic modeling applications. In their other paper (Graves & Jaitly, 2014), authors expanded their research work by presenting an ASR that immediately translates voice to text without the need for an intermediary phonetic representation or language model. This system was a hybrid of DBLSTM, RNN, and CTC objective functions. Authors tested the system on two different evaluations, 1) 14 h and (2) 81 h, and discovered that the proposed system achieved a WER of 27.3% on the WSJ corpus without any proper linguistic information, 21.9% with allowed words lexicon, and 8.2% with a trigram language model.

Deep Speech by Hannun et al. (2014) was another model published in 2014 that had a significant impact on the research community and revolutionized the use of RNN in ASR. The RNN was utilized as the core of the model, which was trained to take speech spectrograms directly as an input and outputs text transcription. Their approach does not require background noise, reverberation, or speaker variation for training improvement, instead, it learned directly from a function that is resistant to such effects. The key to their model approach was a well-optimized RNN training system that makes use of several GPUs and data synthesis techniques such as the use of beam search algorithm, ReLU and dropout function, which allowed the authors to effectively gather a huge amount of diverse data for training. The Authors avoided using LSTM (which earlier researchers claimed outperforming RNNs) because of its high computational cost. For model training, they merged recorded utterances from different datasets, including (1) WSJ, (2) Switchboard, (3) Fisher, and (4) Baidu, to generate an extensive dataset comprising of 5000 h of read speech from 9600 speakers. The authors then used the whole Switchboard Hub5'00 dataset for evaluation and reported a 16% WER on the entire Switchboard Hub5'00 dataset, outperforming all previously reported findings on the entire dataset. The authors also compared their deep speech model to current state-of-art commercial systems of that year in a noisy environment and discovered that it outperformed all of them.

Following the success of DNNs and RNNs, researchers in the research community began to work on end-to-end (E2E) models and Seq-to-Seq (s2s) models. As these Deep neural acoustic models require phoneme lexicons, custom pronunciation dictionaries and as well as a multi-stage training approach to ensure that the components work together, as a result, Chorowski et al. (2015) proposed a framework named Attention-based Recurrent Sequence Generator (ARSG). In their framework, the encoder was implemented as deep BiRNN, while the recurrent activation unit was LSTM and GRU. On the TIMIT phoneme recognition task, the authors obtained 18.7% PER, comparable to state-of-art systems. The major drawback of their approach was that the same or comparable elements were scored similarly regardless of their location in the sequence, authors noticed this happening as a result of BiRNN. To address this drawback of both location-based and content-based mechanisms, the authors suggested a novel and generic way for augmenting attention mechanisms with location awareness by combining their ARSG framework with a convolution module that takes into account the alignment produced by ARSG and makes this framework position aware. This new method helped in developing a model with a PER of 18% in a single utterance and 20% in ten times longer (repetitive) utterances. The authors then proposed a tweak to the attention mechanism by including smoothing, which stops it from concentrating too much on a single frame, lowering the PER to 17.6%. In their attempt to use attention in ASR, their results were comparable, if not superior, than the previous state-of-art model, which used a hybrid DNN-HMM-RNN.

In another work, Miao, Gowayyed, and Metze (2015) presented an E2E system called EESEN, which utilized a single deep RNN as an acoustic model and LSTM units as RNN construction blocks. The authors used the CTC objective function to estimate the overlaps between label and speech sequences, which eliminated the necessity for pre-generated frame labels. To make acoustic modeling simpler, this EESEN model used the CTC objective function to make a single RNN learn over a pair of Context-Independent (CI) labels and speech sequences. Their EESEN system was unique because it employed an extended decoding technique based on Weighted finite-state Transducers (WFSTs). Individual modules such as lexicons, CTC labels, language models were coded into WFSTs and were combined to create a search graph. The authors used the WSJ corpus and received 81 h of transcribed speech as part of the data preparation process, from which they picked 5 per cent for cross-validation and the rest 95 per cent for the training set. The authors discovered that the WERs of EESEN were comparable to strong hybrid HMM/DNN baselines. Their E2E EESEN system achieved a WER

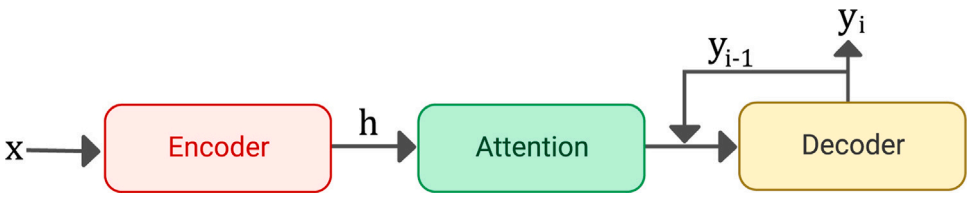

Fig. 7. General structure of LAS model (Chan, Jaitly, Le, & Vinyals, 2016).

of around 7.87% using both the language model and the lexicon while decoding. Furthermore, the use of CI modeling targets enables EESEN to accelerate decoding and reduce decoding memory utilization.

In computer vision and speech recognition, data augmentation has been extremely helpful in enhancing DNN performance. Data augmentation can be used to avoid overfitting, increase the quantity of the training data, and improve the robustness of the model. In another work by Ko, Peddinti, Povey, and Khudanpur (2015) 2015, they displayed how speech augmentation was helpful in improving the robustness of ASRs trained on large datasets LVCSR and on a subset of Switchboard Hub5 00. It became a fixture in the ASR pipeline around that time, with practically all state-of-art speech research work incorporating some sort of speech augmentation.

In 2016 Amodei et al. (2016) presented a system named Deep Speech 2, which was identical to its core to the Hannun et al. (2014) Deep Speech model. Deep Speech 2 was based on RNN with multiple recurrent (unidirectional or bidirectional) layers, one or more convolutional input layers, and one fully connected layer prior to a softmax layer. The network was trained E2E, with the CTC loss function which allowed them to predict character sequences directly from the input audio. The network was fed a series of log spectrograms of power-normalized audio clips evaluated on 20 ms windows. Each language's alphabet was produced as an output. CTC models were linked with language models trained on larger corpora of text at inference time. They experimented with the Batch Normalization (BatchNorm) method to deal with the depth of RNN layers and to train deeper networks faster. Their training data includes 12k h of labeled English speech and 9400 h of labeled Mandarin speech. Authors tested Deep Speech 2 on the different datasets and obtained WER accuracy of (1) 4.42% on the WSJ eval'93 dataset, (2) 5.15% on LibriSpeech test-clean dataset, (3) 7.94% on VoxForge American–Canadian and (4) 21.56% on CHiME eval dataset. On the Mandarin Chinese speech dataset, their best model obtained an accuracy of 7.93% on the test dataset. As a result, the authors demonstrated the effectiveness of E2E deep learning models for speech recognition in a variety of settings and on a variety of datasets.

In other work, Chan et al. (2016) presented a Listen, Attend, and Spell (LAS) model as represented in Fig. 7, a neural-based voice recognizer that transcribes audio utterances straight to text without the use of any other typical speech recognition components or HMM and pronunciation models. The neural network-based architecture of LAS incorporates the acoustic, language and pronunciation models. LAS, unlike DNN-HMM, CTC, and many other models, does not make its own independent assumptions regarding the probability distribution of the obtained output character sequence based on a given acoustic sequence. LAS system had two components: a listener and a speller. A pyramidal BiLSTM RNN encoder that receives filter bank spectra as inputs were used as the listener. The speller was a recurrent network decoder based on attention that generates each character based on all preceding characters and the whole acoustic sequence. The authors evaluated their model on the Google Voice Search dataset which consists of three million utterances (roughly 2000 h of data), with 10 h chosen at random as a validation set. LAS attained 14.1% WER without an external language model or a dictionary and 10.3% with language model rescoring. The authors claim their model outperformed the state-of-art Convolution LSTM-based DNN-HMM (Sainath, Vinyals, Senior, & Sak, 2015) model, which achieved 8.0% WER on the same dataset.

8.3. End-to-end speech recognition with RNN, language models and attention

Followed by the development of DNNs and RNNs, researchers started to focus on end-to-end models and seq-to-seq models (Bahar et al., 2020; Bérard, Besacier, Kocabiyikoglu, & Pietquin, 2018; Cho et al., 2018; Hadian, Sameti, Povey, & Khudanpur, 2018; Li & Doddipatla, 2023; Lu, Cao, Zhang, Chiu, & Fan, 2020; Nguyen, Stüker, & Waibel, 2020; Weiss, Chorowski, Jaitly, Wu, & Chen, 2017). As a result, Seq-to-Seq and end-to-end attention-based models were becoming more popular (Chung, Weng, Tong, & Glass, 2019; Li et al., 2022; Mamyrbayev, Oralbekova, Alimhan, & Nuranbayeva, 2023; Potapczyk & Przybysz, 2020; Pratap et al., 2019; Schneider, Baevski, Collobert, & Auli, 2019; Zhang, Ling, et al., 2019; Zhang, Peng, et al., 2021; Zhang & Sennrich, 2021).

Watanabe et al. (2018) published another noteworthy paper in 2018 in which they introduced ESPnet, an E2E speech processing toolkit. ESPnet was primarily concerned with E2E ASR and uses Py-Torch and Chainer, as its primary deep learning engine. For the purpose of data processing and feature extraction, it was based on the Kaldi ASR toolkit and includes many recipes, resulting in a complete environment setup for speech processing and speech recognition research. ESPnet employs two E2E ASRs. (1) CTC-based — It makes good use of Markov assumptions for solving sequential problems effectively utilizing dynamic programming. (2) Attention-based — It identified symbols using an attention mechanism and performs alignment between audio frames. ESPnet employs a hybrid of attention-based encoder–decoder models and CTC for training and decoding. The authors used the multi-objective learning framework during training to enhance robustness on irregular alignments and achieve quick convergence. To further reduce irregular alignments, the authors performed joint decoding by merging both CTC and attention-based scores in a one-pass beam search algorithm. Aside from the basic architectures mentioned above, the authors additionally used an RNN-based language model and the warp CTC library for fast CTC computation. They evaluated their model against several baselines and found more competitive results. On the WSJ dataset, they found a WER of 8.9%. Aside from that, they reported CER on CSJ and HKUST Mandarin CST tasks which were 6.8% and 28.3% respectively.

In 2019 Zhang, Ling, et al. (2019) proposed two techniques for improving their Seq2seq voice conversion model that they introduced in Zhang, Ling, Liu, Jiang, and Dai (2019). The first strategy authors adopted was to perform a secondary job of predicting linguistic labels from the model's middle layers. To do so, they added two auxiliary classifiers, one to encoder outputs and another one to the RNN decoder inputs. The linguistic labels, which correspond to the current hidden representation of decoder and encoder RNN, were the target of these two classifiers. These auxiliary classifiers improved their seq2seq VC model (Zhang, Ling, et al., 2019) by utilizing stronger text supervision and adding these classifiers to the encoder and decoder also helped the attention module in predicting correct alignments. Because these classifiers were only employed during the training stage and not the conversion stage, no additional input or computation was required during conversion. A data-augmentation strategy was proposed in the second method, in which original utterance fragments were randomly extracted at each training phase. The authors defined these fragments in

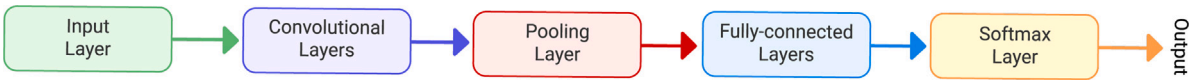

Fig. 8. A generic CNN's architecture.

silence so that they would be unaffected by the surrounding contents. The authors were able to alleviate the instability problem of mispronunciations by using multitask learning, while the data augmentation method assisted in boosting the performance of their Seq2seq VC model with limited training data.

In 2020, (Hayashi et al., 2020) proposed ESPnet-TTS, a new E2E-TTS toolkit based on their ESPnet toolkit Malik et al. (2021). The ESPnet-TTS was made up of two parts: the first was a library of E2E-TTS neural network models called Tacotron 2, Transformers, and FastSpeech, and the second was a set of recipes that contain all of the necessary steps to complete the experiment. Each model has a sequence of characters or phonemes as input and a sequence of acoustic features as output. The Tacotron 2 model was an RNN-based model with a bi-directional LSTM as an encoder and a unidirectional LSTM as a decoder. A multi-head self-attention mechanism was used in the transformer model. Its parallelizable self-attention structure substitutes RNNs, allowing for more efficient and quicker training while preserving excellent perceptual quality. FastSpeech, on other hand, was built on a feedforward Transformer architecture to ensure non-autoregressive generation. ESPnet-TTS processes data in phases similar to Kaldi. The experimental evaluation results reported by the authors in their paper indicate that their models can reach state-of-art performance which was comparable to other recent toolkits.

Inaguma et al. (2020) enhanced this toolkit by presenting the ESPnet-ST, which was developed in order to accelerate the growth of speech-to-speech translation systems in a unified model. Approaches such as multi-task learning, transfer learning, and SpecAugument were employed to fine-tune the ESPnet-model. Other additional functionalities of the system were experiment manager, large-scale training/decoding, performance monitoring, and ensemble decoding. By mid-2021, The ESPnet project has grown greatly, incorporating state-of-the-art methodologies based on community-driven development by numerous contributors. As stated in Watanabe et al. (2021), ESPnet currently incorporates Speech Translation (ST),Text-to-Speech (TTS), Speech Enhancement (SE) and Voice Conversation (VC) capabilities, as well as speech separation, dereverberation, beamforming, and denoising. The authors also discussed how the integration of the Conformer encoder with a Transformer decoder, in the ESPnet project helped in achieving better WER and CER than the present state-of-the-art models on the LibriSpeech dataset.

Specialized in processing sequential data, RNNs prove apt for applications like speech recognition, language modeling, and time series prediction. Their ability to capture temporal dependencies over time stems from the recurrent connections and memory cells embedded within their architecture. Moreover, RNNs showcase flexibility in handling input sequences of variable lengths, making them adaptable for tasks with diverse input sizes. RNNs update their internal state iteratively based on preceding inputs, allowing them to generate outputs progressively. Despite their strengths, RNNs encounter challenges such as the vanishing or exploding gradient problem, constraining their capacity to capture long-term dependencies effectively. Sequential computation inherent in RNNs may result in sluggish training times, particularly evident when processing lengthy sequences. Additionally, standard RNN architectures grapple with retaining information over extended sequences, leading to potential information loss. Furthermore, RNNs may exhibit unstable training dynamics, posing convergence issues, especially when employing gradient-based optimization methods. To address challenges, variants like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have emerged, mitigating issues like vanishing gradients and improving long-term dependency modeling. Despite these advancements, RNNs' sequential nature may not align well with tasks demanding parallel computation.

End-to-end ASR models simplify training and may improve accuracy by jointly optimizing all components at the same time, but they have limitations when compared to traditional pipeline-based techniques. One major limitation is their lack of interpretability, as these models do not allow separate analysis of components like acoustic or language models. This issue can be mitigated using attention mechanisms, and interpretable architectures. Another challenge is their high demand for large amounts of labeled data, whereas traditional models leverage custom characteristics and domain-specific knowledge for better data efficiency. Techniques such as transfer learning, domain adaptation, data augmentation, and semi-supervised learning can address this by leveraging pre-trained models or using unlabeled data. End-to-end models also tend to struggle with out-of-domain data and unexpected variations, unlike the robust traditional models with their modular design. This can be improved through domain adaptation, multi-task learning, ensemble methods, and adversarial training, which expose the models to a broader range of scenarios. Additionally, the computational intensity of end-to-end models can be a significant barrier. However, model optimization techniques like pruning, quantization, and knowledge distillation, along with the use of hardware accelerators and distributed training frameworks, can reduce complexity.

8.4. Convolutional Neural Networks (CNN's)

CNNs were initially employed to solve challenges in Computer Vision (CV). Due to their superior generation and discrimination capabilities, CNNs have been widely used in Natural Language Processing (NLP) in recent years.

A standard CNN design as shown in Fig. 8 consists of many convolutional and pooling layers for classification, as well as fully connected layers. Convolutional layers are formed by convolving kernels with the input. By using a convolutional kernel, the input signal is divided into smaller sections, which are collectively referred to as the kernel's receptive field. Additionally, the convolution process is carried out by multiplying the kernel by the portions of the input that are contained inside the receptive field. Convolutional algorithms can be classed as 1D or 2D networks.

Two-dimensional Convolutional Neural Networks (2D-CNNs) produce two-dimensional feature maps from audio signals. Acoustic characteristics (i.e., MFCC characteristics) are organized in a 2D feature map, with one axis denoting the time domain and the other frequency domain. One-dimensional CNNs, on the other hand, receive acoustic features as direct input. For Speech recognition, each input feature map $X = (X_1, X_2, \ldots, X_I)$ in 1D-CNN is connected to numerous feature maps $O = (O_1, O_2, \ldots, O_J)$ and is represented as:

$$O_j = \rho\left(\sum_{i=1}^{I} X_i w_i, j\right), \quad j \in [1, J] \tag{8}$$

where w denotes local weight. w, O are vectors in 1D-CNNs whereas in 2D-CNNs these are matrices.

In one notable work by Watanabe et al. (2017) in 2017, the authors suggested by removing the requirement for linguistic information such as pronunciation dictionaries, an E2E ASR can considerably minimize the difficulty of designing ASR systems for modern languages. They proposed a language-independent multilingual ASR system based on neural networks. The authors used a hybrid attention/CTC architecture for their model (Deng et al., 2013). Attention was used for flexible alignments and CTC was used as regularization at the time of training

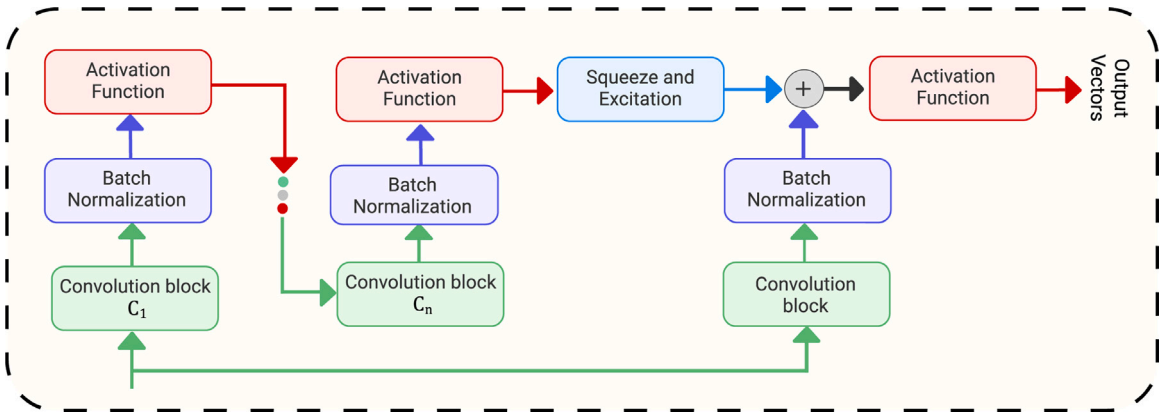

Fig. 9. Illustration of the ContextNet model (Han et al., 2020).

and as a score correction while decoding. Encoder networks used deep CNNs with BLSTMs, and decoder networks used RNN language models pre-trained with text data. For this experiment, the speech database included 10 languages [Voxforge (Portuguese, German, Italian, French, Dutch, Spanish, and Russian), Corpus of Spontaneous Japanese (CSJ), WSJ (English) and HKUST Mandarin CTS]. The authors tested their model with language-independent E2E ASR systems in a variety of experimental combinations and compared it to a language-dependent system. They achieved an average CER of 16.6% on datasets with 7 languages and 21.4% on datasets with 10 languages adopting their best configuration, which included deep CNN with BLSTM and RNN-LM, and was better than the language-dependent approach, which achieved 22.7% and 27.4% CER with 7 and 10 languages, respectively. As a result, their model performed comparable/superior to language-dependent E2E ASR systems.

Other work by Cho et al. (2018) in 2018 used a similar approach but tested on the BABEL corpus (collected from the IARPA babel program). The majority of this Conversational Telephone Speech (CTS) corpus was made of scripted and far-field recordings. The authors tried to adopt the seq-to-seq approach in this work whereas the model used was based on work by Watanabe et al. (2017). The authors picked the best system configuration of Watanabe et al. (2017) which comprised of deep CNN with BLSTM and character level RNN-LM. Then authors used that configuration in their work and trained their system with 10 multiple languages (eval set) that approximates to 600 h of training data. Then for testing authors make use of transfer learning and tested their system on 4 different languages (target set, consist 10 h of data) other than the 10 languages used while training. The end results on the Babel Speech corpus were not much significant in comparison to those of monolingual systems tested on this corpus. As reported in the paper, authors achieved an average CER rate of 37.4% on the eval set, whereas with retraining on the target set they achieved an average CER of 36.7% and an average WER was around 60.9%, as reported by them the WER was higher because of the use of character-level RRN language model instead of word level.

ContextNet (Han et al., 2020) has a CNN-RNN transducer architecture and a fully convolutional network that uses a squeeze-and-excite module to provide global context information into the layers. The CNN with K layers generates characteristics as follows:

$$h = C_K(C_{K-1}(...(C_1(x)))) \tag{9}$$

where C denotes a convolutional block proceeded by batch normalization and activation functions as illustrated in Fig. 9.

ContextNet had 23 convolution blocks (C0, ..., C22). Apart from C0 and C22, which comprises only one layer of convolution, all other convolution blocks contain five layers of convolution. In addition, the squeeze-and-excitation block produces a channel-wise weight θ along with an average pooling layer, that is then multiplied by the input x as follows:

$$\begin{aligned} \bar{x} &= \frac{1}{T}\sum_{t=0}^{T} x_t \\ \theta &= fc\ (\bar{x}) \\ SE\ (x) &= \theta * x \end{aligned} \tag{10}$$

Over the output of the squeeze-and-excitation (SE) block, a skip connection is applied with projection.

The authors evaluated ContextNet on LibriSpeech in three distinct configurations: ContextNet (Small), ContextNet (Medium), and ContextNet (Large), with and without language models along with a varying number of layers and filters. On a test-clean subset of LibriSpeech ContextNet(Large) with 112.7M parameters achieved a WER of 2.1 per cent without a language model and WER of 1.9 per cent with a language model.

CNNs excel in capturing spatial hierarchies within acoustic features, particularly beneficial for speech recognition tasks. Their efficiency is enhanced through parameter sharing across spatial positions, effectively reducing the overall model complexity. Furthermore, CNNs exhibit resilience to variations in input audio, allowing them to recognize speech patterns regardless of their location within the audio sequence. By leveraging convolutional layers, they facilitate parameter sharing and captures spatial hierarchies, resulting in improved efficiency and performance for ASR-related tasks. Transfer learning adds another layer of usefulness, as pre-trained CNN models can be adapted for ASR tasks with limited data, leveraging knowledge acquired from extensive datasets. Their effectiveness is evidenced by successful applications in speech recognition tasks including speaker identification, phoneme recognition, and speech-to-text transcription. The limited receptive field of CNNs may constrain its ability to grasp long-range dependencies in sequential data, posing a hurdle in capturing nuanced speech patterns effectively. Moreover, to enhance their generalization capabilities to diverse speech variations, CNNs may rely heavily on data augmentation techniques, to generalize across variations in input data, adding another layer of complexity to the training process. To address such challenges, the Transformer architecture emerged, effectively handling these issues. The Transformer architecture revolutionized various natural language processing tasks, including speech recognition, by introducing attention mechanisms that capture dependencies across input sequences more efficiently than traditional recurrent or convolutional architectures.

8.5. Transformer

Machine translation and speech recognition have recently witnessed considerable advancements because of the introduction of Transformer networks. Transformer models for voice recognition are commonly

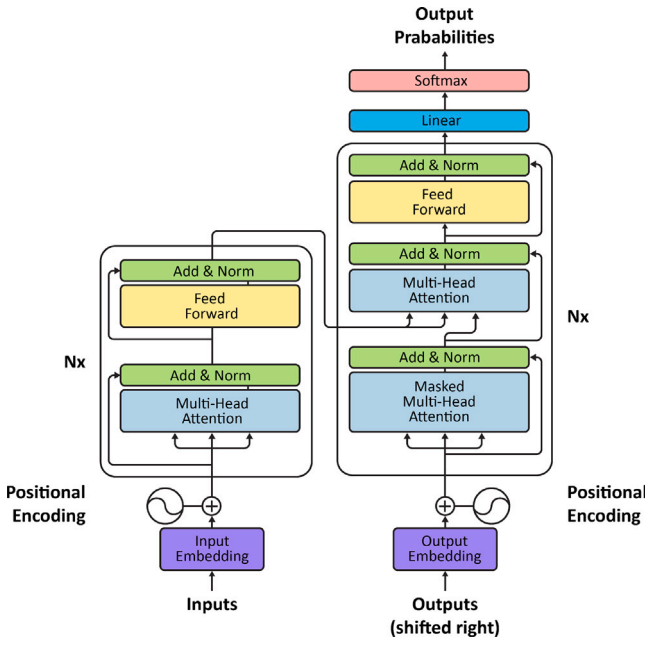

Fig. 10. Illustration of the Transformer model (Vaswani et al., 2017).

built in an encoder–decoder fashion, such as seq-2-seq models. More precisely, they are centered around the process of self-attention rather than the recurrence mechanism used by RNNs. By paying attention to different locations in a sequence, self-attention is capable of extracting meaningful representations Three types of input are accepted by the self-attention mechanism: queries Q, values V, and keys K. Self-attention outputs are computed as:

$$\text{Attention}\ (Q, K, V) = \text{softmax}\ (\frac{QK^T}{\sqrt{d_k}})V \tag{11}$$

where $(\frac{1}{\sqrt{d_k}})$ denotes scaling factor.

Multi-head attention is employed by Transformer, which determines self-attention h times, one for each head i as opposed to single-head attention. As a result, each attention module concentrates on distinct portions and learns new representations. The following equation is used to determine multi-headed attention:

$$\begin{aligned} &\text{MHA}\ (Q, K, V) = \text{concat}\ (h_1, h_2, \ldots, h_h)\ W^0 \\ &\text{where head}\ (h_i) = \text{Attention}\ (QW_i^Q, KW_i^k, V_i^V) \end{aligned} \tag{12}$$

and $W_i^Q\ \epsilon\ \mathbb{R}^{d_{\text{model}} \times d_q}$, $W_i^K\ \epsilon\ \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V\ \epsilon\ \mathbb{R}^{d_{\text{model}} \times d_v}$, $W_i^O\ \epsilon\ \mathbb{R}^{hd_v \times d_{\text{model}}}$ and d_{model} the dimensionality of the Transformer.

At last, a feed-forward network with ReLU activation functions and two fully connected networks is employed as follows:

$$\text{FFN}\ (x) = \text{ReLU}\ (xW_1\ +\ b_1)\ W_2\ +\ b_2 \tag{13}$$

where $W_1\ \epsilon\ \mathbb{R}^{d_{\text{model}} \times d_{ff}}$, $W_2\ \epsilon\ \mathbb{R}^{d_{ff} \times d_{\text{model}}}$ are the weights and $b_1\ \epsilon\ \mathbb{R}^{d_{ff}}$, $b_2\ \epsilon\ \mathbb{R}^{d_{\text{model}}}$ are the biases. In most cases, a positional encoding method is used, which is added to the input to allow the Transformer to attend to relative positions. The widely used method for this is sinusoidal encoding. Finally, to accelerate training, normalization layers and residual connections are used. Fig. 10 depicts the Transformer model's general architecture.

Attention mechanisms enhance the effectiveness of deep learning models in ASR by enabling alignment of audio features with transcript parts, thus improving accuracy. They handle variable-length sequences, crucial for ASR due to varying audio recording lengths, and capture long-range dependencies between distant audio frames and transcript characters. Attention mechanisms also improve robustness to noise and distortion by dynamically focusing on relevant input parts.

Vaswani et al. (2017) presented another significant study in 2017 that provided a new direction to the research community and paved the road for further advancement of ASR systems. In their study (Vaswani et al., 2017), the authors suggested the Transformer, a new basic network architecture that constructs global dependencies between input and output only using attention, discarding convolutions and recurrence entirely. For both the decoder and encoder, the Transformer used point-wise, fully connected layers and multilayer self-attention layers. The encoder and decoder were composed of six layers that were identical to one another. Following that, each layer was separated into two sub-layers. The first was a self-attention mechanism with several heads, whereas the second was a simple, completely linked position-wise feed-forward network. The decoder also makes use of a six-layer stack. Along with these two sublayers in each encoder layer, the decoder adds the third sublayer to each encoder layer in order to perform multi-head attention over the result of the encoder stack. Additionally, the authors employed residual connections surrounding each sublayer and then normalized both the encoder and decoder layers. The authors trained their model on the WMT 2014 English-German (En-De) dataset, which contains around 4.5 million phrase pairs, and the WMT 2014 English-French (En-Fr) dataset, which contains approximately 36 million sentence pairs. For optimization, authors used Adam optimizer (Kingma & Ba, 2014). On the WMT 2014 En-De translation test, their model scored 28.4 BLEU, outperforming the previous best, including ensembles, by more than 2 BLEU. After 3.5 days of training on eight GPUs, their model scored a new single-model all-time high BLEU score of 41.8 on the WMT 2014 En-Fr translation challenge. The authors concluded from the results of these translation tasks that the Transformer could be trained substantially quicker than models based on convolutional or recurrent layers, as they achieved a new state-of-the-art on both the WMT 2014 En-De and En-Fr translation challenges.

Alternative attention-based architectures for ASR include multi-head attention, which attends to different input parts simultaneously, improving the capture of diverse linguistic patterns. Self-attention mechanisms, like those in Transformer models, can enhance model dependencies within the audio input. Hierarchical attention mechanisms attend to larger audio segments first and then focus on smaller segments, capturing both local and global dependencies. Location-based attention mechanisms prioritize certain input positions at each decoding step, useful for tasks where temporal information is crucial. Collectively, these attention mechanisms and architectures significantly enhance ASR models' performance and adaptability.

After reviewing the research work of Vaswani et al. (2017), many researchers followed the same methodology as it became a sort of standard in the ASR pipeline around that time, with practically all state-of-art speech research work comprising models that were totally reliant on attention mechanisms. This results in the growth of seq-2-seq and E2E attention-based models.

In one such research work, Dong, Xu, and Xu (2018) presented the Speech-Transformer, a no-recurrence seq-2-seq model that depended exclusively on attention mechanisms for learning positional dependencies, and can be trained more efficiently and quickly. Speech-Transformer converts speech feature sequences to character sequences. Two-dimensional spectrograms with frequency and time dimensions were used to generate the feature sequence, which was longer than the output character sequence. CNNs were utilized to take advantage of the structural localization of spectrograms and to minimize length mismatches by striding along time. The queries, keys, and values from CNNs were extracted and provided to the two self-attention modules. 2D attention was used to attend both time and frequency dimensions. In Dong et al. (2018) authors devised a 2D-Attention mechanism that can concurrently attend to the frequency and time axes of the 2-D speech inputs, allowing the Speech-Transformer to provide more expressive representations. The authors then tested their models on the WSJ dataset, where they trained on si284 set, validated on dev93

set, and evaluated on eval92 set, where their best model achieved a competitive WER of 10.9%, while the overall training process took around 29 h on 1 GPU, which was significantly quicker than previous findings of recurrent seq-to-seq models.

In recent work, Hou et al. (2020) proposed a large-scale E2E multilingual model for ASR and language identification task. Their model was language-independent based on Transformers with a hybrid CTC/attention architecture similar to Watanabe et al. (2017) consisting of three components: a shared encoder, a CTC module and an attention decoder. Multi-task learning was also supported by the model. To ensure all target languages use the same parameters and network architecture, the authors adopted a language-independent architecture. Additionally, the authors used a shared vocabulary, which ensured that the output vocabulary contains characters or sub-words from all target languages. The authors gathered data from 11 datasets, including CHiME4, Fisher Switchboard, Aurora4, Voxforge, WSJ, Babel, Fisher CallHome Spanish, Common Voice, AISHELL, HKUST and Corpus of Spontaneous Japanese (CSJ) resulting in a large dataset with 42 languages and approximately 5000 h of training and testing data. For 42 languages, the authors achieved an average WER of 52.80 per cent, an average CER of 27.80 per cent, and average LID accuracy of 93.50 per cent. The average WER decreased from 52.80 per cent to 49.60 per cent when a large-size sub-word-level vocabulary was used, and the average CER decreased from 27.80 per cent to 27.20 per cent. Meanwhile, the average LID accuracy has been enhanced from 93.50 to 94.00, indicating that the model's performance in multilingual tasks has been further improved. The authors demonstrated that large-scale pre-training of the model significantly improves the model's performance on under-resource languages.

The authors in Schneider et al. (2019) proposed dedicated frameworks aimed at learning speech representations, notably introducing the wav2vec framework. Wav2vec employs a self-supervised training approach, utilizing the contrastive predictive coding (CPC) loss function, thereby enabling the acquisition of speech representations without reliance on transcription or segmentation. This approach has enabled wav2vec to achieve exceptional performance in various speech-processing tasks, including speaker recognition, speech recognition, and spoken language understanding. For validation of the wav2vec framework, the authors tested it on the Librispeech and WSJ datasets. The results indicated a noteworthy 2.43% reduction in WER compared to the prevailing state-of-the-art ASR system at that particular time.

The emergence of wav2vec as a successful model was largely influenced by the accomplishments of BERT, thereby opening the doors for the development of other specialized frameworks that utilize transformers to acquire representations from multi-modal data. In 2022 Baevski et al. (2022) introduced data2vec, which aims to learn multi-modal representations of various data types, including text, speech, and images, utilizing a contrastive learning objective. Similar to wav2vec, data2vec adopts a self-supervised training methodology that eliminates the need for labeled or annotated data. It achieves this by maximizing agreement between differently augmented views of the same data sample. Distinct from wav2vec, which focuses exclusively on speech signals, data2vec can process diverse data types and acquire joint representations capable of capturing crossmodal correlations and facilitating knowledge transfer across modalities. The authors evaluated their model using the Librispeech-test-other test set. For a 960-hour labeled dataset, their base model achieved a WER of 5.5, while the large model attained a WER of 3.7. The self-supervised training approach employed by data2vec enables the acquisition of representations without the dependency on labeled data, thus rendering it a scalable and cost-effective solution applicable to various domains.

A recent noteworthy contribution by Radford et al. (2023) a transformer-based model called Whisper, which has gained prominence for its applicability in speech recognition tasks conducted in noisy or low-resource environments. Whisper exhibits versatility by successfully performing multiple speech-related tasks, including speech translation, multilingual speech recognition, and language identification. The model's design encompasses weak supervision and employs a minimalist approach to data pre-processing. The model's architecture is trained on an large dataset comprising diverse audio samples and exhibits multitasking capabilities for applications such as transcription, education, entertainment, voice assistants, and accessibility. The minimalistic data pre-processing approach employed by Whisper enables the prediction of raw text transcripts without significant standardization eliminating the need for a separate inverse text normalization step, thereby simplifying the speech recognition pipeline. The authors evaluated the large variant of the Whisper on the LibriSpeech-clean dataset, achieving an impressive WER of 2.7. In a multilingual scenario, the Whisper model achieved a WER of 7.3 on the Multilingual LibriSpeech (MLS) dataset and 13.6 on the VoxPopuli dataset. Notably, the authors achieved these results without employing self-training or self-supervision techniques. They emphasized that training on a large and diverse supervised dataset while focusing on zero-shot transfer can significantly enhance the robustness of a speech recognition system.

Transformers have significantly advanced speech recognition through their ability to model long-range dependencies and parallel computations, particularly in NLP tasks. Leveraging self-attention mechanisms, they excel at capturing global dependencies within input sequences, thereby enhancing their ability to handle long-range dependencies efficiently. Pre-training on extensive text corpora using unsupervised methods like masked language modeling and sparse attention mechanisms has yielded substantial performance improvements across various NLP benchmarks. Pre-trained transformer models, such as BERT and GPT, have demonstrated exceptional performance on various speech recognition benchmarks with minimal fine-tuning. However, their implementation demands considerable computational resources, especially for large-scale models housing millions or billions of parameters. Transformers may encounter challenges in accurately capturing positional information within sequences, as their self-attention mechanisms treat all tokens uniformly, irrespective of their positions. Training transformers on speech recognition tasks with limited data can be intricate, given their reliance on pre-training on extensive corpora limiting their efficacy in low-resource settings, and their complex attention mechanisms present challenges in interpreting their inner workings. Despite these considerations, transformers excel in parallel processing, offer a global context understanding, exhibit scalability for long sequences, and demonstrate superior performance with pre-training, showcasing their potential impact on advancing speech recognition technologies.

8.6. Conformers

The Conformer model is a slight variation of the Transformer model because it combines Transformer with CNNs to provide a more efficient architecture and requires fewer parameters to capture both global and local speech dependencies. The Conformer module includes one CNN layer, two feed-forward layers, and a Multi-Head Attention (MHA) module as presented in Fig. 11. Its output is computed as

$$\begin{aligned} x_1 &= x + FFN(x) \\ x_2 &= x_1 + MHA(x_1) \\ x_3 &= x_2 + CNN(x_2) \\ y &= LN(x_3 + FFN(x_3)) \end{aligned} \tag{14}$$

The convolutional module here employs efficient pointwise and depthwise convolutions, as well as layer normalization. Transformer networks have also been employed with CTC and language models.

The Conformer model is the most recent model that is currently in use. This model is frequently employed by researchers in various configurations in their ASR systems because, in a few areas where attention-based Transformer models were trailing, those limitations

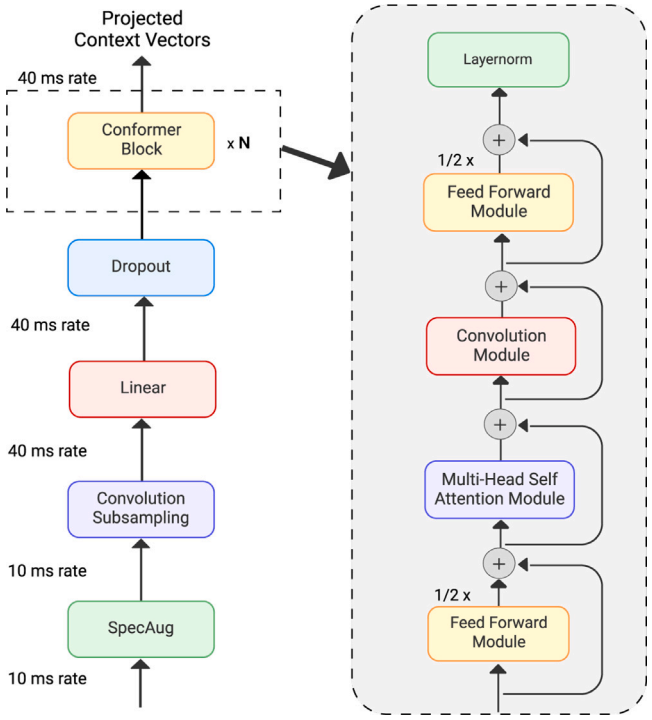

Fig. 11. Illustration of the generic Conformer model (Gulati et al., 2020).

were handled by these attention-based Conformer models. Gulati et al. (2020) addressed this topic, by hypothesizing that both local and global interactions are important in an audio signal in order for it to be parameter efficient. The authors pointed out a few flaws in the Transformer and convolution models. They claimed that the self-attention-based Transformer design was widely employed for modeling sequences because of its ability to identify lengthy interactions and great training efficiency. While transformers were effective in modeling long-range global context, they were less effective at extracting local feature patterns, which were better extracted by convolution modules, which gradually capture local context through a local receptive field layer by layer. However, one restriction of employing local connectivity was that convolution required a large number of parameters or layers to acquire global information. Hence, the authors presented a unique mix of convolution and self-attention to obtain the perfect combination: where convolutions efficiently capture local correlations based on relative offsets while self-attention learns global interactions. The Conformer Encoder model architecture mentioned in the paper is made up of four stacked modules: (1) a feed-forward module, (2) a multi-head self-attention module, (3) a convolution module, and (4) a second feed-forward module at the end. Their convolution module includes pointwise and depthwise convolution, as well as GLU and batch normalization. Authors tested their Conformer model on the LibriSpeech corpus; without the language model, their medium model (with 30.7M parameters) achieved a competitive WER of 5.0 on the test-other subset of LibriSpeech corpus when compared to state-of-art models, whereas their large model (with 118.8M parameters) outperformed state-of-art models with WER of 4.3 without the use of language model and 3.9 while utilizing language model.

In 2021, Chen, Xing, Xu, Pang, and Du (2022) introduced a self-supervised pre-trained model specifically designed for end-to-end speech recognition named Speechformer. The model utilizes masked acoustic modeling and contrastive predictive coding techniques. Unlike preceding models that employed either convolutional or recurrent neural networks, Speechformer adopts a transformer-based encoder–decoder architecture equipped with relative position encoding and layer normalization. It was trained on substantial amount of unlabeled speech data, amounting to 53,000 h and exhibits competitive performance when evaluated on various ASR benchmarks.

Conformers present a compelling advancement in the field of speech recognition, offering a hybrid architecture that combines the strengths of transformers and CNNs, facilitating the seamless integration of both sequential and spatial data. The incorporation of depth-wise convolutions and approximate attention mechanisms enhances computational efficiency, ensuring the capture of both local and global dependencies crucial for accurate recognition. Initially designed for bioinformatics, conformers demonstrate versatility, showing promise in domains necessitating the integration of sequential and spatial information, a trait valuable for speech recognition. They address scalability concerns inherent in traditional transformers by incorporating efficient attention mechanisms, making them particularly well-suited for processing lengthy sequences encountered in speech recognition tasks. However, challenges persist in their adoption, implementation and training conformer architectures may require large-scale speech datasets. Furthermore, fine-tuning pre-trained conformer models for specific speech recognition tasks may demand significant computational resources and labeled data, underscoring the complexities associated with their utilization in this domain.

8.7. Recent deep learning models in the context of low resource languages

End-to-end (E2E) multilingual models (Chen, Yang, Yeh, Jain, & Seltzer, 2020; Hussein, Chowdhury, & Ali, 2021; Kumar et al., 2021; Nowakowski, Ptaszynski, Murasaki, & Nieuważny, 2023) have shown considerable potential in expanding ASR coverage for the world's languages (Le et al., 2021; Li et al., 2020; Pham et al., 2021; Tang et al., 2021; Tjandra et al., 2022; Zhou, Li, Sun, & Liu, 2022). They have outperformed monolingual systems in terms of performance and have improved training and operation by removing language-specific pronunciation, acoustic and language models (Fan, Guo, Zhang, Yang, & Lin, 2023; Reitmaier et al., 2022).

There seems to be a little research and development in early years of the last decade because GMM/HMM models were adopted for speech recognition during that time, and GMM/HMM systems perform poorly when trained in a multilingual manner. However, with the arrival of DNNs and hybrid DNN/HMM models, as well as with the availability of large corpus/datasets, research in multilingual speech recognition began to take off in a more significant way. Later in 2015, Muller (Müller & Waibel, 2015) provided a way for directly adding linguistic information to the network, allowing it to become language adaptable by using DNN instead of GMM/HMM. Their proposed method was to provide the DNN with a language code to make better use of the multilingual data. As a result, the DNN was made aware of the different languages and could even implicitly learn linguistic features. Adding the language code early or at a later stage resulted in overall improvements as this made DNN more language adaptive and allowed it to learn the features of various languages, therefore the resulting DNN was language adaptive (LA-DNN).

In Asian and African Countries, due to their great cultural and linguistic diversity (Pratap et al., 2023), there is a huge potential of adopting speech technology in these type of multilingual environment (Dash, Kim, Teplansky, & Wang, 2018; Gupta, Mundra, Mahajan, & Modi, 2021; Khanuja et al., 2021; Shah, Guha, Khanuja, & Sitaram, 2020; Zhang, Shi, & Chen, 2021) and when some major work in the last five years was reviewed (Basu et al., 2019; Hou et al., 2020; Mridha, Ohi, Hamid, & Monowar, 2022; Pulugundla et al., 2018; Sailor & Hain, 2020; Sen, Agarwal, Ganesh, & Vuppala, 2021; Srivastava et al., 2018; Vuddagiri, Gurugubelli, Jain, Vydana, & Vuppala, 2018; Zhang et al., 2023), then in 2018 Toshniwal et al. (2018) presented a single seq-to-seq ASR model which was trained in nine different Asian languages having little overlap in between their character set and scripts. Their model was based on LAS attention-based model presented by Chan

et al. (2016). The authors down-sampled the original LAS model a bit to work with the above dataset. The encoder was comprised of 5 layers of stacked bidirectional RNN and the decoder was comprised of 2 layers of stacked unidirectional RNN. The database consisted of data from nine Asian languages making it a total of about 1500 h of training data and 90 h of testing data. The authors also included grapheme sets of all nine languages and trained them jointly on the seq-to-seq model along with data from all nine languages. Their Joint LAS model trained on all languages achieved a weighted average WER of 22.9% outperforming the language-specific model for all languages. Model accuracy was further improved by providing language identifiers to the encoder and decoder which achieved weighted average WER of 21.32%.

In an other work Kannan et al. (2019) in 2019 addressed two issues (1) the need for streaming ASR, (2) the challenge of imbalanced training data. For first, they presented a streaming E2E multilingual system using RNN-T (Graves, 2012). To address the data imbalance problem authors used up-sampled data from low-resource languages which was more even and distributed across different languages. Additionally, the authors incorporated language-specific adaptor modules, which enabled the model to specialize in every language in a similar manner as fine-tuning the entire model would, but with fewer parameters. Their training and test datasets were constituted of anonymous, human-transcribed sentences that are representative of Google's traffic and cover nine Asian languages which were equivalent to 37K hours of data. Since transcriptions contained a mixture of Latin and native scripts, the authors used transliteration-optimized WER as evaluation metrics. Using nine Asian languages, their best system, built with adaptor modules and an RNN-T model achieved an average WER of 22.6%, which outperforms the state-of-art monolingual traditional recognizers and the monolingual RNN-T models which achieved an average WER of 27.4% and 28.0% respectively. Along with a reduced WER, the multilingual RNN-T model had the advantage of integrating nine distinct recognizers (usually composed of pronunciation, acoustic, and language models) into a single, compact recognizer.

Later in 2020, Shetty and Mary (2020) explored the implementation of the transformer model on under-resourced Asian languages in a multilingual fashion. The authors experimented with a few approaches for incorporating language information into the multilingual transformer model. These method involves adding language information or utilizing language identity tokens to the acoustic vectors. By learning language embeddings, language information is passed to acoustic vectors. The authors did not employ a pronunciation dictionary or language models. Each speech in their sample belonged to a single language, and there was no language switching inside an utterance. Their transformer-based encoder–decoder models include an encoder with 12 layers and a single layer decoder, with four attention heads on each layer. The authors used a dataset made available at INTERSPEECH 2018 for Indian languages based on the Low resource speech recognition challenge. It contains data in three Asian languages (Telugu, Gujarati and Tamil). The authors discovered that feeding linguistic information into their transformer model during training enhanced performance over baseline models. After combining acoustic feature vectors with language embeddings and retraining, the authors achieved an average WER of 33.6% and an average CER of 9.06% on the evaluation set of their dataset.

In 2021, Diwan et al. (2021) proposed a MUCS multilingual ASR w.r.t to low-resource Multilingual and Code-Switching (MUCS) challenge. The authors used two different subtasks for developing MUCS ASR systems. Subtask 1 in which a multilingual ASR system in 6 Asian languages (Gujarati, Hindi, Telugu, Odiya, Marathi, Bengali, Tamil) was developed. Subtask 2 involves designing a code-switching ASR system for Bengali–English and Hindi–English pairs. For both subtasks, baseline systems were developed using hybrid DNN-HMM models and in addition, an E2E model is also employed for Subtask 2. For Multilingual ASR a Hybrid DNN-HMM model was built that was based on the Kaldi toolkit with a sequence-trained time-delay neural network (TDNN) architecture along with the lattice-free MMI objective function. This architecture consists of 6 TDNN blocks. For Code-switching ASR, a Hybrid DNN-HMM model same as in subtask 1 along with E2E ASR consisting of a hybrid CTC-attention model based on a Transformer was used. In this architecture, the encoder was of 12 layers with 6 layers decoder. The dataset used was approx 600 h and consisted of transcribed speech from different domains in 7 Asian languages (Hindi, Telugu, Odiya, Gujarati, Marathi, Tamil, and Bengali). The averaged WER across six languages on the blind test for Multilingual ASR and Monolingual ASR for subtask 1 was 32.73% and 28.38% respectively, whereas for subtask 2, the average WERs from GMM-HMM, DNN-HMM and E2E ASR systems for Hin-Eng and Ben-Eng test sets were 41.75, 35.63 and 32.45 respectively and blind test average WER for subtask 2 by end-to-end ASR was 29.17%.

In another research on multilingual speech recognition, Krishna (2021) proposed a dual-decoder conformer model in conjunction with multi-task learning. Their system consists of one conformer encoder and two decoders (Grapheme Decoder (GRP-DEC) and Phoneme Decoder (PHN-DEC)) both based on a transformer and a language classifier. The phoneme decoder and grapheme decoder network were made up of M cross-attention layers, which were in charge of the phoneme recognition task and predicts the grapheme sequence for a given input utterance respectively. Following that, the grapheme decoder estimates grapheme units from a multilingual vocabulary. This multilingual vocabulary comprises of linguistic labels and grapheme units from all languages. During training, the model performs auxiliary tasks such as phoneme sequence prediction and language identification. The author took two ways to give linguistic information at both the decoder and the encoder. The entire model was trained from the beginning to end using the joint CTC-Attention function. For language label prediction, the language classifier was made up of two linear layers proceeded by a softmax layer. The conformer block was made up of two feed-forward (linear) layers as well as a convolution module. It was in charge of deriving local feature representations from feature sequences. The output of the conformer encoder was fed into both decoders alongside the CTC and Language classifier. The CTC layer predicts grapheme units for each frame along with language identity information. Both decoders were built using the Transformer architecture. The authors used data from the MUCS 2021 challenge, which included testing and training data in 6 Asian languages: Gujarati, Hindi, Telugu, Odiya, Marathi, and Tamil. The dataset came from a variety of categories, including Stories, healthcare, agriculture, finance, and general themes, and provided them with approximately 135 h of training data and 10 h of test data. Their dual-decoder conformer model achieved a weighted average WER of 26.70%, which authors claimed was better than the other baselines examined on this dataset at that time.

In Javed et al. (2022), the authors used the wav2vec 2.0 architecture, which includes a feature encoder for converting raw audio into a sequence of latent representations, a transformer for learning contextualized representations for each unit in the sequence, and a quantizer for discretizing the learned representations to facilitate self-supervised learning targets. The authors gathered 17,000 h of raw speech data from 40 Indian languages, sourced from various fields such as education, news, technology, and finance. Several wav2vec-style models were pretrained on this data, demonstrating that the features across layers were specific to the language family, with attention heads focusing on local contexts. As a result, the authors fine-tuned the model for ASR tasks in nine languages, obtaining notable results, including an average WER of 10.5 in Hindi, 14.8 in Gujarati, 12.2 in Marathi, 17.2 in Oriya, 20.0 in Tamil, and 15.2 in Telugu. The authors also reported state-of-the-art performance across three public datasets, notably for low-resource languages like Sinhala and Nepali.

In Bhogale, Sundaresan, et al. (2023), the authors evaluated the performance of the OpenAI Whisper pretrained model, which they subsequently fine-tuned on the Vistaar-train set. They chose the Whisper model due to its notably lower WER compared to other models for the

Hindi language. Each language in the Vistaar-train set was represented by a separate model, collectively referred to as IndicWhisper. The authors employed the Whisper-Medium model with 769M parameters and 24 layers in both the encoder and decoder, fine-tuning it individually for each of the 12 languages in the Vistaar-train set. The resulting IndicWhisper model achieved an average WER of 13.8 for Hindi, 20.1 for Bengali, 22.8 for Gujarati, 18.3 for Kannada, 32.3 for Malayalam, 18.2 for Marathi, 27.4 for Odia, 20.5 for Punjabi, 48.0 for Sanskrit, 25.3 Tamil, 28.8 for Telugu, and 19.4 for Urdu, yielding an overall average WER of 24.6 across the entire Vistaar-train set. The authors noted that their model outperformed other transformer-based models on some benchmarks, thus setting a highly competitive standard for Indian language ASR. The WER varied significantly across languages, ranging from 13.6 in Hindi to 48 in Sanskrit. This suggests the potential for improvement through the collection of larger datasets, particularly for low resource languages such as Odia, Malayalam, and Sanskrit.

There is still a significant amount of research underway for multilingual ASR, particularly for low-resource languages (Al-Ghezi, Getman, Voskoboinik, Singh, & Kurimo, 2023; Kwon & Chung, 2023; Peterson, Tong, & Yu, 2022). The recent introduction of transformers and conformer-based models has greatly enhanced research in this area. Furthermore, this advancement has facilitated exploration into domain-specific ASR research, that was earlier less feasible. However, the development of ASR in multilingual contexts still has a long way to go before achieving accuracy comparable to that of monolingual ASR models. The performance of deep learning models for ASR varies significantly across languages and dialects due to factors like training data quality, linguistic diversity, and acoustic variability. High-resource languages like English, Mandarin, and Spanish benefit from abundant labeled data, resulting in a lower WER. In contrast, low-resource languages, particularly many African and indigenous languages, experience higher WER due to limited data. Languages with complex morphology, such as Finnish and Turkish, and those with diverse phonetic sounds, like Vietnamese, pose additional challenges. Accents and dialects within the same language also significantly impact ASR performance, as ASR models are trained primarily on standard accents and often struggle with regional variations.

Also, Transfer learning can improve the generalizability of ASR models across domains and tasks. Key optimization techniques include task-specific fine-tuning, which adapts pre-trained models to specific tasks or domains; multi-task learning, which improves generalization by training on multiple related tasks at the same time; and domain adaptation, which adjusts models for new domains while accounting for differences in accents or background noise and emotions (Kang, 2024; Pan, 2024). Furthermore, leveraging knowledge from similar domains via pre-training followed by fine-tuning can be useful, and employing regularization techniques such as dropout, weight decay, or layer normalization, can prevent overfitting and promote learning of generalizable features.

9. Latest ASR models based on deep learning

We discussed in Section 8 how DNNs, CNNs, RNNs, Transformers, and Conformers differ from each other, in terms of their working, advantages and limitations. DNNs have emerged as versatile models with applications spanning speech recognition, image classification, and natural language processing. Their ability of autonomous feature learning reduced the need for human feature extraction, while their scalability helped with training and deployment of large-scale networks. With advancements in deep learning techniques, DNNs consistently achieve state-of-the-art results across various tasks like classification and pattern recognition. However, challenges include the need for large labeled datasets, powerful hardware, and the risk of overfitting, especially with limited data. Additionally, interpreting DNN decisions is challenging due to their opaque nature. CNNs excel in spatial feature capture, benefiting speech recognition tasks by efficiently recognizing patterns regardless of their location within audio sequences. Despite their effectiveness, CNNs struggle with capturing long-range dependencies in sequential data, often relying on data augmentation techniques. RNNs specialize in sequential data processing, capturing temporal dependencies over variable-length sequences, but face challenges like vanishing gradients and slow training times. Transformers, leveraging self-attention mechanisms, advance speech recognition through parallel processing and global dependency capture. Pre-trained transformer models demonstrate exceptional performance but require significant computational resources. Conformers, a hybrid of transformers and CNNs, offer promise in integrating sequential and spatial data for speech recognition. Their efficient attention mechanisms address scalability concerns but may require large datasets and computational resources for training and fine-tuning. Despite challenges, transformers and conformers showcase advancements in speech recognition, offering potential impacts on future technologies in the field. Fig. 12 shows different speech recognition techniques observed in background study of this paper.

Through innovative methodologies and thorough evaluations across diverse datasets, ASR models like Wav2Vec, Wav2Vec2, HuBERT, Whisper, and WavLM have significantly improved the accuracy and robustness of ASR systems, paving the way for more efficient and reliable STT transcription technologies. Each model brings unique features and improvements, contributing to the evolving landscape of ASR technology. Fig. 13 showcases a timeline of noteworthy large Transformer models along with their parameter sizes developed for speech processing in recent years. Table 6 shows the WER of some of these models on the LibriSpeech Test sets. Given below are some key features associated with these models are discussed.

Wav2Vec (Schneider et al., 2019) an ASR model developed by Facebook AI Research (FAIR), introduced a transformative approach to ASR. Utilizing self-supervised learning, Wav2Vec pre-trains on large amounts of unlabeled speech data, eliminating the need for costly manual annotations. Wav2Vec was initially tested on the LibriSpeech dataset and on WSJ achieving state-of-the-art accuracy.

- Self-Supervised Learning: Wav2Vec's core innovation lies in its self-supervised learning approach. By extracting representations from unlabeled audio data, it overcomes the limitations of supervised methods, achieving impressive accuracy.
- Contrastive Predictive Coding: The model's pre-training methodology is based on contrastive predictive coding, enabling it to learn intricate patterns in speech signals. This leads to more robust and generalized representations.

DiscreteBERT (Nguyen, Sagot, & Dupoux, 2022) is an adaptation of BERT architecture that works with discrete token inputs rather than continuous token embeddings. This adaptation is useful for handling tokenized inputs from various sources, such as audio or text, where the tokens are discrete symbols representing different units of information.

- Bidirectional Contextual Understanding: Unlike traditional left-to-right or right-to-left models, DiscreteBERT understands context in both directions.
- Transformer Architecture: Utilizes self-attention mechanisms to weigh the importance of different tokens in the sequence.
- Masked Language Modeling (MLM): Trains by predicting randomly masked tokens in the input sequence, improving its understanding of language context.

Conformer (Gulati et al., 2020) integrates CNNs with transformers for speech recognition tasks. It combines the strengths of both the architectures to capture both local and global dependencies in audio sequences. Conformer architecture is already discussed in detail in Section 8 subsection 6.

- Hybrid Architecture: Combines convolutional layers for local feature extraction with transformers for global context modeling

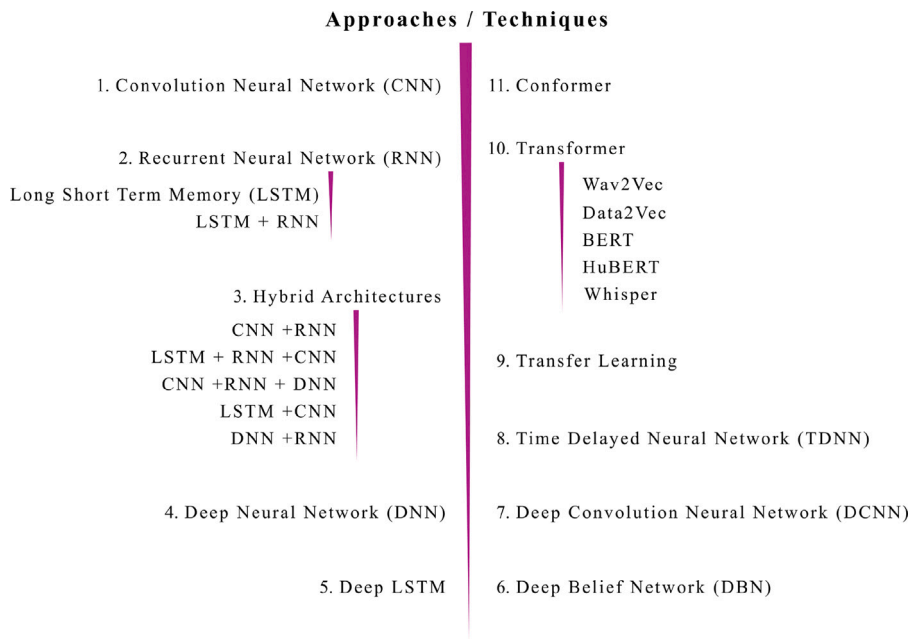

Fig. 12. Various speech recognition techniques observed in background study of the paper.

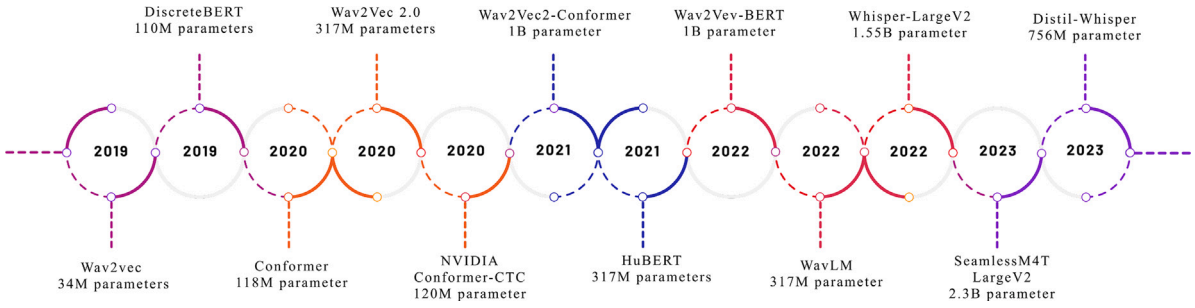

Fig. 13. A timeline showcasing some noteworthy large Transformer models along with their parameter sizes developed for speech processing in recent years.

Table 6
WER on the LibriSpeech Test sets after training on the clean 100-hour subset of LibriSpeech.

Ref	Model	Unlabled data	LM	test-clean	test-other
Schneider et al. (2019)	wav2vec 2.0	LS-960	None	6.1	13.3
Chen, Wang, et al. (2022)	WavLM Base	LS-960	None	5.7	12.0
Radford et al. (2023)	Whisper Large V2	LS-960	None	2.7	5.2
Chen, Wang, et al. (2022)	DiscreteBERT	LS-960	4-gram	4.5	12.1
Schneider et al. (2019)	wav2vec 2.0	LL-60k	Transformer	2.0	4.0
Hsu et al. (2021)	HuBERT Large	LL-60k	Transformer	2.1	3.9
Chen, Wang, et al. (2022)	WavLM Large	MIX-94K	Transformer	2.1	4.0

- Self-Attention Mechanism: Utilizes self-attention to model long-range dependencies in speech sequences. Supports end-to-end learning for speech recognition.
- Flexibility: Adaptable to various speech processing tasks, including recognition, synthesis, and translation.

Wav2Vec2 (Baevski, Zhou, Mohamed, & Auli, 2020) an evolution of Wav2Vec, introduced by Facebook AI Research. It builds upon the success of its predecessor by incorporating context prediction tasks and transformer architectures, leading to superior accuracy in transcribing speech to text. Wav2Vec2 has achieved remarkable results on datasets like LibriSpeech and Common Voice, surpassing its predecessor.

- Context Prediction Tasks: Wav2Vec2 enhances its representations by incorporating context prediction tasks during pre-training. This allows the model to capture more nuanced contextual information, improving its transcription capabilities.
- Transformer Architectures: The model leverages transformer architectures, further refining its ability to handle complex speech patterns.

Conformer-CTC (Burchi & Vielzeuf, 2021) is a combination of the Conformer model and the CTC loss function. The Conformer model processes the input audio features, extracting both local and global context. The output is then fed into the CTC layer. CTC layer allows the model to learn the mapping from input sequences to output sequences without requiring aligned data. It does this by summing over all possible alignments during training, making it suitable for sequence-to-sequence problems where the input and output lengths are not fixed.

- Conformer Architecture: Combines the strengths of CNNs and transformers to capture both local and global audio features.

- CTC Loss Function: Utilizes CTC loss for sequence alignment without the need for frame-level labeling.
- Efficient Alignment: Effective at handling varying speech lengths and aligning predictions with input sequences.

Wav2Vec2-Conformer (Lam et al., 2024) integrates Wav2vec2.0's audio representation capabilities with Conformer's hybrid architecture for enhanced speech recognition. Wav2Vec 2.0 is used to pre-train the model on large amounts of unlabeled audio data focusing on learning general-purpose speech representations. Then Conformer is used in the fine-tuning stage with labeled data to adapt the learned representations to specific ASR tasks. The fine-tuning process benefits from the Conformer's ability to handle both local and global dependencies in the speech data.

- Dual Architecture: The combined model demonstrates superior performance in ASR tasks by effectively leveraging the strengths of both Wav2Vec 2.0 and Conformer architectures.
- Robust Feature Extraction: Enhanced audio feature extraction and contextual understanding.
- Efficiency: The Conformer's hybrid approach to local and global feature extraction makes the model more efficient and accurate in handling diverse speech patterns.

HuBERT (Hsu et al., 2021) developed by Microsoft Research, takes a unique approach to ASR by focusing on masked prediction tasks for speech representation learning. This model has demonstrated competitive performance, particularly in scenarios with limited labeled data and showcased competitive accuracy on datasets such as LibriSpeech, VoxCeleb and has been tested across diverse datasets, demonstrating its versatility in handling different linguistic contexts.

- Masked Prediction Tasks: HuBERT focuses on predicting masked hidden units within the network, leading to robust representations of speech signals.
- Hierarchical Representations: The model employs hierarchical representations and self-attention mechanisms, enhancing its ability to capture contextual information.

Wav2vec-BERT (Kim, Kim, Shin, & Lee, 2021) Wav2vec-BERT integrates Wav2vec's speech representation capabilities with BERT's language modeling strengths, creating a hybrid model for comprehensive speech and language understanding. The audio data is first processed using Wav2Vec 2.0, which converts it into high-level speech representations. These representations are then fed into a BERT model fine-tuned for ASR or other specific tasks. BERT processes these representations to understand the context and improve transcription accuracy. Unlike traditional language models that read text sequentially (left-to-right or right-to-left), BERT reads in both directions, allowing it to understand context more fully.

- Dual Model Integration: Combines Wav2vec for audio processing and BERT for contextual language understanding.
- Self-Supervised Learning: Learns from unlabeled audio data, reducing dependency on large labeled datasets.
- Multi-Modal Learning: Combines audio and text modalities for improved performance in tasks requiring both speech and language understanding.

Whisper (Radford et al., 2023) is an ASR model introduced by OpenAI, which combines supervised and self-supervised learning techniques. Leveraging a diverse dataset consisting of 680,000 h of multilingual and multitask supervised data, Whisper learns robust representations from noisy audio or low-resource environments. This approach enhances its ability to transcribe speech correctly across various languages and tasks, making it suitable for real-world applications. Whisper effectively improves its ability to recognize speech in challenging situations and is capable of performing multiple speech-related tasks, uses weak supervision and a minimalist approach to data preprocessing. Whisper has shown great results on datasets like VoxCeleb and VoxForge, emphasizing its efficacy in capturing speaker-specific features for ASR tasks.

- Hierarchical Self-Attention: Whisper utilizes weighted hierarchical self-attention mechanisms, allowing it to capture detailed speaker-specific features during training.
- Speaker Representation Learning: The model focuses on learning representations that enhance speaker-dependent ASR tasks.

WavLM (Chen, Wang, et al., 2022) is an ASR model developed by researchers, integrating principles from both ASR and language modeling. The model extends the self-supervised learning paradigm to language modeling for speech, aiming to enhance the contextual understanding of spoken language. By pre-training on large-scale unlabeled speech corpora, WavLM learns contextual representations that capture the underlying semantics of spoken language. This approach eliminates the need for text transcriptions during pre-training, making it highly scalable and applicable to diverse languages and dialects. The model's performance remains competitive across various ASR tasks, showcasing its effectiveness in capturing the complexities of spoken language in real-world scenarios.

- Language Modeling for Speech: WavLM extends self-supervised learning to language modeling, capturing contextual information directly from raw audio signals.
- Scalability: The model's scalability is evident in its ability to handle large-scale datasets, contributing to its competitive performance.

SeamlessM4T (Barrault et al., 2023) architecture is designed for multilingual and multimodal tasks. It aims to integrate text and speech processing capabilities in multiple languages into a single model. It supports both text and speech as input formats and uses shared representations to process different types of inputs.It can also handle multiple languages without needing separate models for each language and uses a shared multilingual encoder–decoder architecture.

- Multilingual Capability: Supports multiple languages and tasks, making it highly versatile.
- Multitask Learning: Capable of handling various tasks such as speech recognition, translation, and language modeling.

Distil-Whisper (Gandhi, von Platen, & Rush, 2023) is a distilled version of OpenAI's Whisper model, providing efficient speech recognition and translation capabilities without significantly sacrificing performance. Distillation reduces the model size while maintaining performance through knowledge distillation. It is suitable for deployment in resource-constrained environments and at the same time maintaining high accuracy despite reduced model size.

- Model Distillation: Reduces the size and complexity of the original Whisper model through distillation.
- Efficient Speech Recognition: Lower computational requirements reduce infrastructure costs for large-scale deployment while retaining strong ASR capabilities and being more efficient.
- Efficiency: Faster inference times and reduced memory usage make it ideal for real-time applications.

The pros and cons of above deep learning models, along with their comparative advantages, are presented in Table 7. Moreover, the reliance on labeled data for training some of these models raises questions about the potential integration of unsupervised and semi-supervised learning methods to reduce data dependency. The following learning methods can be integrated into other models to reduce reliance on labeled data for training of those ASR models.

Table 7
Key findings of recent state-of-the-art deep learning models.

Model	Key Feature	Advantages	Shortcoming
Wav2vec	–Self-Supervised Learning: Learns speech representations from raw audio data without supervision. –Raw Audio Processing: Directly processes raw audio, transforming it into meaningful feature vectors. –Fine-Tuning for ASR: Can be fine-tuned with labeled data for improved ASR performance.	–Data Efficiency: Reduces reliance on large amounts of labeled data, making it scalable. –Robust Speech Features: Learns rich speech representations that generalize well to various speech tasks. –High Performance: Achieves competitive results on standard ASR benchmarks.	–Noise Sensitivity: Early versions may struggle with noisy or low-quality audio. –Resource Intensive: Requires significant computational power for training and fine-tuning.
DiscreteBERT	–Discrete Token Handling: Unlike the original BERT, which uses continuous embeddings, DiscreteBERT processes discrete tokens directly. –Transformer Architecture: Maintains the core transformer architecture, ensuring powerful bidirectional context understanding. –Pre-training on Text Corpora: Similar to BERT, DiscreteBERT is pre-trained on large text datasets, allowing it to learn a broad range of language patterns and structures.	–Efficiency with Discrete Data: More efficient for tasks involving discrete data inputs, such as code generation or structured text. –Strong Contextual Understanding: Retains BERT's ability to understand context deeply, leading to high performance in various NLP tasks. –Versatility: Can be fine-tuned for numerous downstream tasks, similar to BERT.	–Limited Application Scope: Primarily useful for discrete data, not suitable for continuous data inputs. –Complexity in Training: Requires substantial computational resources for pre-training and fine-tuning.
Conformer	–Hybrid Architecture: Integrates CNNs for local feature extraction and transformers for capturing global context. –End-to-End Learning: Supports end-to-end learning for speech recognition. –Self-Attention Mechanism: Utilizes self-attention to model long-range dependencies in speech sequences.	–Balanced Feature Extraction: Effectively captures both local and global features, enhancing speech recognition accuracy. –High Accuracy: Demonstrates state-of-the-art performance on ASR benchmarks. –Versatility: Can be adapted for various speech processing tasks beyond ASR.	–Complex Architecture: More complex than pure transformer or CNN models, leading to longer training times. –Resource Intensive: Requires significant computational resources for training and inference.
Wav2Vec 2.0	–Quantization Step: Introduces a quantization step in the self-supervised learning process to discretize the continuous speech representations. –Transformer-Based Architecture: Uses transformers for modeling the contextual dependencies in the speech data. –End-to-End Training: Enables end-to-end training for ASR, improving performance and efficiency.	–Robustness: Better handles variations in audio quality and background noise. –Efficient Fine-Tuning: Requires less labeled data for fine-tuning compared to traditional models. –High Performance: Achieves leading results on standard ASR benchmarks.	–Computational Resources: Training the model is resource-intensive. –Complexity: The model's architecture and training process are more complex than its predecessor.
Conformer-CTC	–Conformer Architecture: Inherits the hybrid architecture of Conformer, combining CNNs and transformers. –CTC Loss Function: Uses CTC loss for efficient sequence prediction and alignment. –End-to-End Training: Supports end-to-end training for improved ASR performance.	–Improved Sequence Alignment: CTC loss enhances the model's ability to align input and output sequences. –High Performance: Achieves strong results in ASR tasks due to the combination of Conformer and CTC. –Real-Time ASR: Suitable for real-time ASR applications due to efficient decoding.	–Training Complexity: The combination of Conformer architecture and CTC loss increases the complexity of training. –Computational Requirements: High resource requirements for training and inference.
Wav2vec2-Conformer	–Self-Supervised Learning: Leverages Wav2vec 2.0's approach for learning from raw audio data. –Hybrid Architecture: Integrates Conformer's CNNs and transformers for balanced feature extraction. –Enhanced ASR Performance: Designed to improve accuracy and robustness in speech recognition tasks.	–Robustness: Better handles noisy environments and variations in speech quality. –High Accuracy: Achieves superior performance on ASR benchmarks due to the combination of Wav2vec 2.0 and Conformer. –Efficient Feature Learning: Benefits from Wav2vec 2.0's efficient self-supervised learning.	–Model Complexity: More complex than individual Wav2vec 2.0 or Conformer models, leading to longer training times. –Resource Intensive: Requires significant computational resources for training and deployment.
HuBERT	–Self-Supervised Learning: Learns discrete speech units in a self-supervised manner from raw audio. –BERT-Like Architecture: Uses a transformer architecture similar to BERT for contextual modeling of speech units. –Multi-Stage Training: Involves a multi-stage training process, first learning hidden units and then fine-tuning on downstream tasks.	–Rich Speech Representations: Learns robust and meaningful speech representations, enhancing performance in various speech tasks. –Data Efficiency: Reduces the need for large amounts of labeled data through self-supervised learning. –Versatility: Can be applied to a wide range of speech processing tasks, including ASR and speaker recognition.	–Training Complexity: Involves a complex, multi-stage training process that requires significant computational resources. –Noise Sensitivity: May still be sensitive to noisy or low-quality audio inputs.

(*continued on next page*)

- Unsupervised pre-training: Techniques such as auto-encoders or self-supervised learning, may be used to learn useful representations of speech data in an unsupervised manner. These pre-trained representations can then be fine-tuned on a smaller labeled dataset for ASR tasks. By leveraging large amounts of unlabeled data, unsupervised pre-training may improve the generalization capability of ASR models and reduce the need for labeled data.
- Semi-supervised learning: By leveraging both labeled and unlabeled data during training to improve model performance, semi-supervised learning approaches such as co-training, self-training, or pseudo-labeling may be used to incorporate unlabeled data into the training process. This allows the model to learn from a larger and more diverse dataset, leading to improved performance, especially when labeled data is scarce or expensive to obtain.

Table 7 (*continued*).

Model	Key Feature	Advantages	Shortcoming
Wav2vec-BERT	–Wav2vec for Speech Features: Utilizes Wav2vec to extract meaningful speech features from raw audio. –BERT for Language Modeling: Applies BERT for understanding and generating text based on speech input. –Multi-Modal Learning: Combines audio and text modalities for improved performance in tasks requiring both speech and language understanding.	–Comprehensive Understanding: Integrates speech and language models for a deeper understanding of spoken language. –High Performance: Excels in ASR and other speech-related tasks by leveraging strengths of both Wav2vec and BERT. –Versatility: Can be applied to a wide range of tasks, including ASR, speech translation, and more.	–Model Complexity: Increased complexity due to the integration of two advanced models. –Resource Intensive: Requires substantial computational power for training and inference.
WavLM	–Self-Supervised Learning: Trains on raw audio without requiring labeled data. –Robust Speech Representations: Learns rich speech features that generalize well across tasks and datasets. –Low-Resource Performance: Optimized to perform well even with limited labeled data.	–High Performance: Achieves strong results across a range of speech tasks, including ASR, speaker recognition, and more. –Data Efficiency: Effective in low-resource settings, reducing the dependency on large labeled datasets. –Versatility: Can be applied to numerous speech processing tasks, making it a flexible tool for speech research and applications.	–Computational Resources: Requires significant computational resources for pre-training. –Model Complexity: Complex architecture and training process may pose challenges for deployment.
Whisper	–Transformer-Based Architecture: Effective for capturing long-range dependencies and contextual information in speech sequences. –Multilingual and Multitask Learning: Trained on diverse data, supporting multiple languages and tasks. –Large-Scale Training: Robust and generalizable due to extensive training on vast amounts of speech data.	–High Accuracy: State-of-the-art performance in ASR and speech translation. –Versatile: Suitable for transcription, translation, and speech-to-text services. –Robustness: Strong performance with noisy and accented speech.	–Resource Intensive: High computational requirements for training and deployment. –Complexity: Difficult to fine-tune and optimize for specific use cases. –Latency: Potentially higher latency due to model size.
SeamlessM4T	–Multilingual Capabilities: Supports multiple languages, making it suitable for global applications. –Multitask Learning: Capable of handling various tasks such as speech recognition, translation, and language modeling. –Large-Scale Training: Trained on extensive multilingual and multitask datasets.	–Versatility: Can be applied to a wide range of speech and language tasks in multiple languages. –High Performance: Demonstrates strong results across various benchmarks due to large-scale training. –Global Applicability: Suitable for applications requiring support for multiple languages and tasks.	–High Resource Requirements: Requires significant computational and memory resources for training and deployment. –Complexity in Fine-Tuning: Fine-tuning for specific languages or tasks can be challenging due to the model's size and complexity.
DistilWhisper	–Model Distillation: Reduces the size and complexity of the original Whisper model through distillation. –Efficient Speech Recognition: Retains strong ASR capabilities while being more efficient. –Multitask Support: Supports speech recognition and translation tasks.	–Efficiency: Lower computational and memory requirements compared to the original Whisper model. –Faster Inference: Reduced size leads to faster processing times, beneficial for real-time applications. –High Performance: Maintains competitive performance in speech recognition and translation tasks.	–Trade-Offs in Accuracy: May sacrifice some accuracy for efficiency, depending on the task and dataset. –Scope of Tasks: Primarily focused on speech recognition and translation, may not be as versatile for other tasks.

- Weakly supervised learning: In scenarios where obtaining precise transcriptions for large amounts of data is challenging or expensive, weakly supervised learning methods may be used to train ASR models using noisy or incomplete labels. Techniques such as teacher-student training, multi-task learning, or knowledge distillation can help leverage weak supervision to improve model performance.
- Transfer learning: By pre-training models on a large-scale ASR dataset and fine-tuning on a smaller target dataset.
- Data augmentation: Unlabeled data can be used to generate synthetic training examples through data augmentation techniques such as speed perturbation, background noise addition, or time warping. These augmented data techniques can then be combined with labeled data for training ASR models, effectively increasing the size and diversity of the training dataset without additional labeling efforts.

10. ASR: Computational, ethical, and environmental considerations

ASR systems have become integral to modern technology, powering anything from virtual assistants to transcription services. However, developing and deploying large-scale ASR models come with significant computational requirements, scalability, ethical and environmental considerations. Addressing these is important for ensuring the efficient training and operational viability of ASR technologies.

10.1. Computational and scalability consideration

Training and deploying large-scale deep learning models for ASR poses several computational requirements and scalability challenges due to complexity of the models and the size of the datasets involved. Some key considerations are:

- GPU resources: Training deep learning models for ASR typically requires significant computational resources, particularly GPUs, to accelerate the training process. The computational demand increases with the size of the model and the complexity of the dataset.
- Memory requirements: Deep learning models for ASR often have large numbers of parameters, leading to high memory requirements during training. Memory-efficient optimization techniques and strategies such as gradient check-pointing may be necessary to handle memory constraints.
- Training time: Training large-scale ASR models can be time-consuming, sometimes taking days, weeks and even months, especially with large datasets and complex architectures. Improving

training efficiency through techniques like distributed training, mixed-precision training, or efficient model architectures is essential for scalability. To further reduce training time, deep learning models often leverage parallelization techniques such as data parallelism or model parallelism across multiple GPUs or distributed computing resources.

- Inference speed: Deploying large-scale ASR models for real-time applications requires fast inference speeds to process audio input quickly and provide timely responses. Efficient model architectures, hardware acceleration (e.g., GPUs, TPUs), and optimization techniques (e.g., quantization, model pruning) are necessary to achieve low-latency inference.
- Memory footprint: Deploying large models on resource-constrained devices such as mobile phones or edge devices requires careful management of the model's memory footprint. Techniques such as model compression, quantization, and on-device optimization are essential for minimizing memory usage while maintaining performance.

10.2. Ethical considerations

Ethical considerations, particularly those pertaining to biases in multilingual and diverse language settings, are crucial for ensuring fairness and impartiality in ASR systems. It is imperative to address these concerns to uphold ethical standards and avoid perpetuating biases. Some key ethical considerations and potential biases in multilingual and diverse language settings include:

- Representation bias: ASR systems may exhibit representation bias, where certain languages, dialects, or accents are underrepresented or marginalized in the training data. This can lead to disparities in performance across different linguistic groups, with better performance for dominant languages and dialects and poorer performance for minority or low-resource languages.
- Data bias: ASR models rely heavily on training data, which may reflect existing biases present in society. Biases in the training data, such as gender, race, or socioeconomic biases, can be inadvertently learned by ASR systems and perpetuated in their output. For example, ASR systems may exhibit gender bias by misrecognizing female voices more frequently than male voices due to imbalances in the training data.
- Cultural bias: ASR systems may struggle to accurately transcribe speech that contains cultural references, idiomatic expressions, or linguistic features specific to certain cultural groups. This can result in inaccuracies or misinterpretations in the ASR output, particularly for languages and dialects with rich cultural and linguistic diversity.
- Language preservation: ASR systems may inadvertently contribute to language erosion or language extinction by prioritizing dominant languages over minority or endangered languages. This can have detrimental effects on linguistic diversity and cultural heritage, as marginalized languages may receive less attention and support in the development of ASR technologies.
- Accessibility: While ASR systems have the potential to improve accessibility for individuals with disabilities or language barriers, they may also inadvertently exclude certain groups if they are not designed with inclusivity in mind. For example, ASR systems that prioritize certain languages or dialects over others may limit accessibility for speakers of low source languages.

Strategies for mitigating biases and promoting fairness in ASR systems include:

- Diverse and representative datasets: Ensuring that training data for ASR systems is diverse, representative, and inclusive of different languages, dialects, accents, and cultural backgrounds.
- Bias detection and mitigation: Implementing techniques for detecting and mitigating biases in ASR systems, such as data preprocessing, algorithmic fairness, and bias-aware evaluation.
- Community engagement: Engaging with affected communities and stakeholders throughout the development process to understand their needs, preferences, and concerns regarding ASR technologies.
- Transparency and accountability: Providing transparency around the development, deployment, and use of ASR systems, including clear explanations of how data is collected, used, and shared, as well as mechanisms for accountability and recourse in case of errors or biases.

10.3. Environmental considerations

Environmental factors such as background noise, speaker accents, and varying recording conditions significantly impact the performance of ASR systems. Background noise, like traffic, machinery, or crowd noise, can impact speech signal quality and interfere with word recognition. Speaker accents, including regional, foreign, or non-native accents, introduce variations in pronunciation and intonation, making accurate transcription challenging for ASR systems. Recording conditions, such as microphone quality, distance from the microphone, reverberation, and room acoustics, affect speech signal clarity, leading to errors in ASR output. To mitigate these issues and improve recognition, several strategies can be employed like:

- Noise reduction: By applying noise reduction techniques, such as spectral subtraction, Wiener filtering, or deep learning-based denoising algorithms, background noise can be suppressed and the clarity of the speech signal may be enhanced.
- Multi-microphone arrays: Multiple microphones may be used to capture the speech signal and localize the source of the speech, enabling better noise suppression.
- Data augmentation: By augmenting training data with various types and levels of background noise, the robustness of ASR models to noisy environments can be improved.
- Accent adaptation: Adapting ASR models to speaker accents by fine-tuning or retraining the models on data from speakers with diverse accents, will allow the models to better capture accent-specific variations.
- Room Impulse Response (RIR) simulation: Simulation of room reverberation effects by convolving clean speech signals with room impulse responses, may allow ASR models to learn and to deal with reverberating environments.
- Accent-specific language models: Development of accent-specific language models or pronunciation dictionaries may allow better accommodation of accent-specific vocabulary and pronunciation patterns.

11. Reinforcement learning impact

Recently, Reinforcement Learning (RL) has been applied by researchers in various domains beyond ASR, yielding promising results (Javadpour, Ja'fari, Taleb, & Benzaïd, 2023; Oghim, Park, Bang, & Leeghim, 2024). In the past, RL methods were not as effective in the ASR domain. However, with the availability of larger datasets and advancements in training mechanisms, RL is now demonstrating significant improvements in ASR performance (Kheddar, Hemis, & Himeur, 2024). RL involves learning through trial and error enhancing overall generalizability of ASR systems across different environments and recording conditions (Chen & Zhang, 2023). RL agents allow ASR systems to learn continuously from user feedback and real-time interactions, ensuring robustness and adaptability. RL also facilitates transfer learning by helping ASR models leverage knowledge from high-resource languages to improve low-resource language recognition. Main characteristics of RL based ASR systems are:

11.1. Generalizability and robustness

RL enhances the generalizability and robustness of ASR models across diverse environments and recording conditions (Tjandra, Sakti, & Nakamura, 2018). RL enables dynamic adaptation to environments by training agents to detect and respond to various acoustic settings, such as adjusting noise suppression in a noisy street versus a quiet office. Reward-based learning can optimize ASR performance by balancing accuracy and noise resilience through multi-objective rewards and real-time feedback. Exploration of diverse conditions during training, including simulated environments and policy exploration, helps ASR models learn effective strategies for different scenarios (Dudziak, Abdelfattah, Vipperla, Laskaridis, & Lane, 2019). Combining RL with meta-learning and transfer learning allows models to quickly adapt to new environments and apply knowledge from high-resource conditions to low-resource ones. User feedback can be incorporated to personalize and continuously improve ASR performance. Robust policy learning techniques, like Robust Policy Optimization and adversarial training, further enhance model resilience. Implementation involves defining environments, designing reward functions, selecting suitable RL algorithms, training with diverse data, and ensuring continuous learning and evaluation. These RL strategies enable ASR models to adapt flexibly and accurately to varying conditions, improving their effectiveness in real-world applications.

11.2. Dynamic speech variation

In real-world scenarios RL can greatly improve ASR systems by improving their adaptability to dynamic speech variations such as accents, speech speed, and noise (Bai et al., 2024). Reward-based training guides the model by providing positive rewards for correct transcriptions, improving performance in diverse conditions. It also allows dynamic adjustment to context, enhancing accuracy under varying conditions. RL supports interactive learning with user feedback, enabling models to quickly adapt to individual speech patterns. Its suitability for sequential decision-making helps maintain coherence in transcriptions. Integrating RL makes ASR systems more resilient and adaptable to real-world speech dynamics.

11.3. Agent based

Agent-based methods enable dynamic noise handling by adjusting to changing conditions in real-time, leveraging multi-agent collaboration where different agents specialize in detecting, classifying, and mitigating noise (Yang et al., 2024). Agents use reinforcement learning to continuously improve noise adaptation strategies based on feedback, and they can tailor ASR performance to specific environments by learning typical noise patterns. Real-time feedback allows agents to make immediate adjustments, and the combination of various noise mitigation techniques, such as spectral subtraction and beamforming, improves overall resilience. Implementation involves defining agent roles, using a reinforcement learning framework, simulating noisy environments for training, developing collaboration mechanisms, ensuring real-time processing, and continually refining strategies based on performance evaluations. Integrating these methods with deep learning approaches leads to more accurate and reliable speech recognition in noisy conditions.

11.4. Data dependency issues

Agent-based methods can address data dependency issues in ASR for low-resource languages and speaker variability by employing data augmentation and synthesis, where agents generate synthetic speech and apply augmentation techniques to diversify training datasets (Singh, Malato, Hautamaki, Sahidullah, & Kinnunen, 2024). Transfer learning and multilingual models leverage knowledge from high-resource languages to enhance ASR for low-resource languages, with agents managing dynamic model adjustments. Active learning and data selection enable efficient training data usage by prioritizing informative samples for annotation. Speaker adaptation and personalization allow agents to tailor ASR models to individual speakers using limited data, improving accuracy over time. Crowd sourcing and collaborative learning facilitate data collection and annotation from native speakers, with agents coordinating these efforts and utilizing federated learning for privacy (Jin et al., 2024). Self-supervised and unsupervised learning reduce the need for labeled data, with agents overseeing the implementation of techniques that extract robust features from unlabeled speech.

11.5. Challenges associated

Key challenges associated integrating RL with ASR include the complexity of designing appropriate reward functions, which can be addressed through iterative refinement, multi-objective rewards, and human feedback. Balancing exploration and exploitation requires advanced techniques like epsilon-greedy and safe exploration strategies. Scalability and computational costs can be mitigated using parallel training, efficient RL algorithms, and transfer learning. The high data requirements of RL can be managed by creating simulation environments, using data augmentation, and implementing off-policy learning. Ensuring real-time processing demands model optimization and incremental learning. Robustness to noise and variability can be improved through robust training, noise adaptation agents, and adversarial training. Interpretability and debugging issues can be tackled with visualization tools, explainable RL methods, and systematic testing. Finally, integrating RL into existing ASR systems without disruption involves modular integration, ensuring backward compatibility, and adopting hybrid approaches. By addressing these challenges, RL can significantly improve ASR model accuracy and robustness.

12. Conclusion

Multiple past surveys in the field of ASR have often overlooked the significance of various deep learning-based methods, such as transformers, conformers, and language models, in the development of recent ASR models. These surveys primarily focused on classic ASR system solutions, such as the utilization of MFCC for feature extraction, HMM for classification, and language modeling using n-grams. However, it has been observed that HMMs and GMMs exhibited superior performance in basic translation tasks in the past, but these systems required extensive feature engineering and often struggled with speech variability such as accents, speaking styles, and noisy environments, making them inadequate to handle the complexities of current translation challenges. Even WER for early systems were quite high, especially in challenging conditions, often exceeding 20%–30% in real-world applications. Additionally, the computational time required by HMMs and GMMs to analyze large datasets was considerably higher. These critical issues are addressed in this survey paper, which aims to highlight the latest DNN-based techniques that can replace traditional approaches in ASR systems. This shift resulted in immediate performance gains, with WER dropping by around 10%–15% in various benchmarks. DNNs laid the foundation, and subsequent models were introduced to overcome their limitations and adapt to the unique characteristics of speech data. Each model bringing its own set of benefits and drawbacks, shaping the trajectory of ASR research and applications, often reaching around 10%–15% WER on challenging datasets.

While some recent surveys have emphasized the significance of deep learning methods, they have still overlooked certain key areas that are covered comprehensively in this survey. DNN solutions, previously unattractive due to limitations with smaller datasets, have now become highly effective when applied to larger data and model sizes. This survey paper emphasizes the need for adopting deep learning-based

techniques and provides in-depth discussions on their potential to overcome the limitations of conventional approaches. The transition from DNNs to CNNs to RNNs pushed WER down to around 5%–10% on many benchmarks, and current transformers and conformers-based architectures have pushed WERs to below 5%, with ongoing enhancement from multimodal integration and self-supervised learning approaches. This reflects the community's relentless efforts to enhance ASR performance, adapting it to the unique characteristics of speech data, driving continuous accuracy, robustness, and versatility improvements, propelling the field forward. However, it is important to acknowledge that

considerable research is still needed before speech recognition becomes widely accessible and consistently reliable for all users. Achieving this objective poses challenges but promises more natural, accessible, and seamless interactions with technology.

13. Challenges and future directions

With increased computation power and availability of additional data, deep learning models based on self-attention and transformer have emerged as the go-to solution for a wide range of complex problems, including speech recognition (Xu et al., 2021). Table 8 presents an analytical comparison of notable state-of-the-art work based on deep learning approaches. This paper aims to cover groundbreaking work in this field and suggests future challenges for the research community. The integration of multiple DNN layers has become feasible, enabling the replacement of various components of a traditional ASR system. Although end-to-end DNN models have made significant progress, they still exhibit linguistic errors, particularly with terms for which they lack sufficient training examples. Alternatively, technological advancements, particularly in hardware like GPUs and TPUs, along with large corpus of available speech may be used to train a Neural Network Language (NLM) Model based on transformer and conformer architectures.

ASR systems have made significant strides in recent years. Despite these notable advancements, several research gaps still persist across various domains such as dialect and accent adaptation, adaptive and multilingual speech recognition, real-time ASR, addressing challenges in noisy environments, and improving accuracy for low-resource languages. Some potential future directions in the field of ASR pertaining to low resource languages are outlined below:-

- In the English language, the best WER accuracy attained by ASR systems is 1.4 (Zhang et al., 2020), while in the case of low-resource languages like Hindi, Gujrati, Odiya, Tamil and Telugu it is 9.5, 14.3, 20.6, 19.5, 15.1 (Diwan et al., 2021) respectively, for Urdu WER is 13.5 (Farooq, Adeeba, Rauf, & Hussain, 2019) and for Nepali, CER is 10.3% (Regmi & Bal, 2021). therefore there is still a need to design a model to get better accuracy for low-resource languages (Alam et al., 2022).
- The prominent work in multilingual ASR obtained a weighted WER of 12.5 (Zhang et al., 2023) on FLEURS dataset of 62 languages, whereas the best work on Asian languages obtained a weighted WER of 24.6 (Bhogale, Sundaresan, et al., 2023) for a dataset of 12 low-resource languages. In the case of low-resource language pairs, code-switching accuracy right now is low (Mustafa et al., 2022). The WER recorded between English and Hindi is 21.77 (Diwan et al., 2021), while it is 28.27 (Diwan et al., 2021) between Bengali and English.
- Some popular datasets, such as LibriSpeech (WER 1.4), TIMIT (PE 8.3), Switchboard (PE 4.3), and TED-LIUM (WER 5.3), are related to the generic domain. However, further research attention is required for ASR systems in specific domains, particularly in improving speech recognition in healthcare, education, and corporate sectors, especially in the context of low resource languages.
- The WER accuracy observed in multi-speaker detection for the English language using the LibriMix dataset is 24.5 (Guo et al., 2021), while there is not much extensive work related to multi-speaker detection available in the case of low-resource language.
- The existing research on emotion recognition from spoken language primarily revolves around the detection of hate emotion (Yadav, Kumar, Kumar, Shivani, Kusum, & Yadav, 2023). However, it is crucial to address other human emotions in the context of low resource languages
- Biasness occurs in multilingual ASR due to the availability of more datasets from the high-resource language in contrast to the low-resource language (Zhao & Zhang, 2022).
- The existing ASR models are primarily trained on textual data, but when trained on speech data, they do not achieve comparable accuracy. Further research is required to improve the performance of ASR models trained on speech data.
- Although most models are trained on standard language datasets, a little change in language accent might impact model accuracy. Another aspect to consider in order to develop more accent-robust ASR systems.

Researchers have already shown a great deal of interest and attention in the field of automatic speech recognition (Liang & Yan, 2022, 2022; Reza, Ferreira, Machado, & Tavares, 2023; Thomas, Kessler, & Karout, 2022; Zhang et al., 2022) but it is regarded as one of the long-standing issues in the field of artificial intelligence. Nonetheless, this technology still faces numerous challenges and issues that researchers are trying to address (Reitmaier et al., 2022). Few of them are outlined below:-

- ASR systems encounter difficulties handling a broad spectrum of sound intensities, from faint whispers to loud shouts. This variation necessitates robust algorithms and architectures (Kim et al., 2023; Wei, Duan, Li, Yu, & Yang, 2023). Simultaneously, background noise in speech adds another challenge to the recognition process (Dua, Akanksha, & Dua, 2023).
- Data preparation for low-resource languages is a tedious task due to limited training data and vocabulary sizes. Data augmentation (Lam, Schamoni, & Riezler, 2023) emerges as a important technique to mitigate these constraints
- The identification and tracking of speakers in conversations (Moriya, Sato, Ochiai, Delcroix, & Shinozaki, 2023) present significant challenges. Lexical ambiguity adds another layer of complexity, especially when distinguishing homophones like "To", "Two" and "Right", "Write" that sound similar but carry distinct meanings. Additionally, human speech errors, such as mispronunciations and filler words, present another hurdle for accurate ASR recognition.
- Efficient algorithms and architectures are required for real-time speech recognition with minimal delay (Kwon & Chung, 2023). Furthermore, speech recognition systems should evolve over time by continuously adapting to new user, accents, languages, and environments by integrating mechanisms for user feedback and model updates

Some of the latest emerging trends in the field of ASR include few-shot learning and self-supervised learning, which offer significant benefits for training and deployment. Few-shot learning facilitates rapid adaptation to new speakers and environments, making ASR more accessible to diverse languages, dialects, and domains. By leveraging meta-learning techniques and small labeled datasets, few-shot learning enables ASR systems to generalize from limited examples, adapt quickly to new tasks, and scale efficiently in dynamic settings. Self-supervised learning, on the other hand, enhances the adaptability and generalization capabilities of ASR systems by learning from diverse data sources. Contrastive learning is a common approach in self-supervised

Table 8
Key findings of recent state-of-the-art sequence models.

Ref No.	Year	Methodology	Used Dataset	Metrics	Accuracy (in %)	Findings
Chan et al. (2016)	2016	Proposed an E2E LAS model with a BiLSTM-based encoder and an RNN and attention-based decoder.	Google voice search dataset	WER	14.1 (With LM) 10.3 (w/o LM)	LAS was successful when used without LM or dictionary, but not at par with models using LM.
Watanabe et al. (2017)	2017	Proposed a language-independent E2E multilingual ASR. The encoder uses Deep CNN with BLSTM and Decoder was a pretrained RNN-LM.	WSJ, CSJ, HKUST, Voxforge (7 langs)	CER (Avg.)	21.7 (10 langs), 16.6 (7 langs)	Their model outperformed several language-dependent E2E ASR systems by using pre-trained RNN-LM.
Vaswani et al. (2017)	2017	Proposed a seq2seq Transformer model based on multi-head self-attention which use encoder–decoder architecture.	WMT14 En-Gr En-Fr	BLEU	28.4 (En-Gr) 41.8 (En-Fr)	In comparison to RNN and CNN-based models, the Transformer can be trained quickly and performs better.
Dong et al. (2018)	2018	Proposed a Speech-Transformer, a no-recurrence Seq2Seq model based on multi-head self-attention.	WSJ	WER	10.9	On the WSJ, their model outperformed other RNNs or CNNs models while requiring roughly 80% less training time.
Watanabe et al. (2018)	2018	The toolkit employs a hybrid CTC/attention-based encoder–decoder. Additionally, RNN-LM and warp CTC libraries are used.	WSJ, CSJ, HKUST dataset	WER CER CER	8.9 (WSJ) 6.8 (CSJ), 28.3 (HKUST)	A new open-source E2E toolkit when all older were based on HMM and HMM/Neural network.
Toshniwal et al. (2018)	2018	A single seq2seq ASR model based on the LAS model with BiRNN RNN encoder and decoder with unidirectional RNN.	Own (9 Indian langs)	WER (Avg.)	21.32 (9 langs)	The authors downscaled the original LAS model to deal with the dataset and training model on all languages outperformed all language-specific models.
Kannan et al. (2019)	2019	A streaming E2E multilingual ASR model based on the RNN-transducer. RNN-transducer consists of an encoder, prediction network and a joint network	Own (9 Indian langs)	WER (Avg.)	22.6 (9 langs)	To address data imbalance issues, the dataset was upsampled. For fine-tuning, used language-specific adapters. Used transliterated WER.
Hou et al. (2020)	2020	A large-scale E2E multilingual ASR based on Transformer with hybrid CTC/attention architecture with LID	Combined Multiple datasets	WER CER LID.	49.6 (Avg) 27.2 (Avg) 94 (Avg)	Shared vocabulary and adopted language-independent architecture. On low resource languages, large scale pre-training improves model performance.
Shetty and Mary (2020)	2020	Implemented transformer in multilingual manner on Indian languages w/o using LM or pronunciation dictionary.	Interspeech 2018	WER CER	33.6(Avg) 9.06(Avg)	The transformer model was fed language identity tokens, which improved performance over baseline models.
Gulati et al. (2020)	2020	A Conformer model architecture was proposed, which merged Transformer and CNN into a single model.	LibriSpeech	WER	3.9 (With LM) 4.3 (w/o LM)	When both are combined, Transformer is good at capturing global context while CNN is good at capturing local context.
Baevski et al. (2020)	2020	Wav2vec 2.0 uses a self-supervised training approach and incorporated context prediction tasks with transformer architectures to enhance speech representation learning.	Librispeech	WER (Avg.)	1.8 (test-clean) 3.3 (test-other)	Wav2vev 2.0 uses contrastive loss for sampling negative frames and benefits from continuous latent speech representations for better context and robust training via quantized target representations.
Hsu et al. (2021)	2021	HuBert utilized masked prediction to derive strong representations from raw audio data,avoiding the need for annotated labels. This captures contextual and semantic information effectively.	Librispeech	WER (Avg.)	1.8 (test-clean) 2.9 (test-other)	HuBERT excels through iterative representation learning and refining cluster assignments but lags behind approaches combining pre-training with self-training.

(*continued on next page*)

Table 8 (*continued*).

Ref No.	Year	Methodology	Used Dataset	Metrics	Accuracy (in %)	Findings
Diwan et al. (2021)	2021	A hybrid DNN-HMM based system developed using the Kaldi toolkit with a sequence-trained TDNN architecture. An E2E ASR consists of a hybrid CTC-attention model based on Transformer.	MUCS 2021	WER (Avg.)	32.73 (multi) 28.39 (mono) 29.17 (E2E)	Provided MUCS dataset, a large corpus for 6 Asian languages. Also, provide data in two code-switched language pairs.
Krishna (2021)	2021	A dual decoder conformer model with multitask learning model. The model consists of one encoder, 2 transformer-based decoders and a language classifier.	MUCS 2021	WER (Avg.)	26.70	During training using dual decoder preserves LID and phoneme sequence which improved accuracy over other baselines.
Baevski et al. (2022)	2022	Data2vec uses a self-supervised training approach that does not require labels or annotations and learns representations by maximizing agreement between differently augmented views of the same data sample.	Librispeech	WER (Avg.)	5.5 (Base model) 3.7 (Large model)	Data2vec's self-supervised training approach allows it to learn representations without the need for labeled data, making it a scalable and cost-effective solution for many applications.
Radford et al. (2023)	2022	Whisper is a general-purpose model designed for speech recognition in noisy or low-resource settings, and is capable of performing multiple speech-related tasks and uses weak supervision and a minimalist approach to data pre-processing.	Librispeech Librispeech (multi) VoxPopuli	WER (Avg.)	2.7 (mono) 7.3 (multi) 13.6 (multi)	Whisper achieved state-of-the-art results without the need for the self-supervision and self-training techniques and by simply training model on a large and diverse supervised dataset and focusing on zero-shot transfer
Bhogale, Sundaresan, et al. (2023)	2023	The Whisper-Medium model, a transformer-based model with 769M parameters, was utilized and fine-tuned on Vistaar-train set. This collection of models was collectively termed IndicWhisper.	Vistaar	WER (Avg.)	24.6	IndicWhisper model achieved competitive WER results compared Google (23.9) and Azure (20) despite these model not supporting few languages from the 12 included in the Vistaar dataset, thus setting a highly competitive standard for Indian language ASR.

Note: LAS: Listen, attend and spell, E2E: End-to-End, LM: Language model, LID:Language Identification. w/o: without.

learning, training models to distinguish between different views of the same input data. Additionally, self-supervised ASR models can be trained on auxiliary tasks such as audio-visual alignment, audio-text alignment (Dida, Chakravarthy, & Rabbi, 2023), or audio reconstruction, integrating information from multiple modalities for richer speech representations. This adaptability is crucial for real-world applications where acoustic environments or user demographics vary. As these techniques evolve, they promise further improvements in ASR performance, leading to more versatile and contextually aware speech recognition systems, and extending to neuroscientific domains like Brain–Computer Interfaces (Luo, Rabbani, & Crone, 2023), Neurocognitive Models of Speech Processing (Elmer, Kurthen, Meyer, & Giroud, 2023), Audio-Visual Speech Recognition (Hwang, Park, Park, & Park, 2023), and Auditory Neural Networks (Li et al., 2023).

CRediT authorship contribution statement

Harsh Ahlawat: Writing – review & editing, Writing – original draft, Formal analysis, Data curation, Conceptualization. **Naveen Aggarwal:** Writing – review & editing, Validation, Supervision, Project administration, Investigation. **Deepti Gupta:** Writing – review & editing, Validation, Supervision, Project administration, Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Naveen Aggarwal reports financial support was provided by Design Innovation Centre (DIC), Panjab University.

Acknowledgment

This work is partially supported by the Design Innovation Centre (DIC), Department of Higher Education, Ministry of Human Resource Development, Government of India.

Data availability

The authors of this research study do not claim any of the datasets explored in this survey paper. The datasets analyzed in this study are accessible via the URL links provided in section 4 of the paper. Some datasets are open source and can be accessible openly, while others are licensed and can only be obtained from the respective authors and organizations upon reasonable request.

References

Al-Ghezi, R., Getman, Y., Voskoboinik, E., Singh, M., & Kurimo, M. (2023). Automatic rating of spontaneous speech for low-resource languages. In *2022 IEEE spoken language technology workshop* (pp. 339–345). IEEE.

Alam, S., Sushmit, A., Abdullah, Z., Nakkhatra, S., Ansary, M., Hossen, S. M., et al. (2022). Bengali common voice speech dataset for automatic speech recognition. arXiv preprint arXiv:2206.14053.

Aldarmaki, H., Ullah, A., Ram, S., & Zaki, N. (2022). Unsupervised automatic speech recognition: A review. *Speech Communication*.

Alharbi, S., Alrazgan, M., Alrashed, A., Alnomasi, T., Almojel, R., Alharbi, R., et al. (2021). Automatic speech recognition: Systematic literature review. *IEEE Access*, *9*, 131858–131876.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., et al. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. In *International conference on machine learning* (pp. 173–182). PMLR.

An, K., Xiang, H., & Ou, Z. (2020). CAT: A CTC-CRF based ASR toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency. arXiv preprint arXiv:2005.13326.

Anastasopoulos, A., Bojar, O., Bremerman, J., et al. (2021). Findings of the IWSLT 2021 evaluation campaign. In *IWSLT*.

Anoop, K., Pratik, M., Pushpak, B., et al. (2018). The IIT Bombay EnglishHindi parallel corpus. In *Language resources and evaluation conference*.

Ansari, E., Axelrod, A., Bach, N., Bojar, O., Cattoni, R., Dalvi, F., et al. (2020). Findings of the IWSLT 2020 evaluation campaign. In *Proceedings of the 17th international conference on spoken language translation* (pp. 1–34).

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., & Auli, M. (2022). Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning* (pp. 1298–1312). PMLR.

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, *33*, 12449–12460.

Bahar, P., Wilken, P., Alkhouli, T., Guta, A., Golik, P., Matusov, E., et al. (2020). Start-before-end and end-to-end: Neural speech translation by apptek and rwth aachen university. In *Proceedings of the 17th international conference on spoken language translation* (pp. 44–54).

Bai, Y., Chen, J., Chen, J., Chen, W., Chen, Z., Ding, C., et al. (2024). Seed-ASR: Understanding diverse speech and contexts with LLM-based speech recognition. arXiv preprint arXiv:2407.04675.

Bain, M., Huh, J., Han, T., & Zisserman, A. (2023). WhisperX: Time-accurate speech transcription of long-form audio. arXiv preprint arXiv:2303.00747.

Barker, J., Watanabe, S., Vincent, E., & Trmal, J. (2018). The fifth'CHiME' speech separation and recognition challenge: dataset, task and baselines. arXiv preprint arXiv:1803.10609.

Barrault, L., Chung, Y.-A., Meglioli, M. C., Dale, D., Dong, N., Duquenne, P.-A., et al. (2023). SeamlessM4T-massively multilingual & multimodal machine translation. arXiv preprint arXiv:2308.11596.

Basu, J., Khan, S., Roy, R., Saxena, B., Ganguly, D., Arora, S., et al. (2019). Indian languages corpus for speech recognition. In *2019 22nd conference of the oriental COCOSDA international committee for the co-ordination and standardisation of speech databases and assessment techniques* (pp. 1–6). IEEE.

Beilharz, B., Sun, X., Karimova, S., & Riezler, S. (2019). LibriVoxDeEn: A corpus for german-to-english speech translation and german speech recognition. arXiv preprint arXiv:1910.07924.

Bérard, A., Besacier, L., Kocabiyikoglu, A. C., & Pietquin, O. (2018). End-to-end automatic speech translation of audiobooks. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 6224–6228). IEEE.

Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, *56*, 85–100.

Bhable, S. G., Deshmukh, R. R., & Kayte, C. N. (2023). Comparative analysis of automatic speech recognition techniques. In *International conference on applications of machine intelligence and data analytics* ICAMIDA 2022, (pp. 897–904). Atlantis Press.

Bhogale, K., Raman, A., Javed, T., Doddapaneni, S., Kunchukuttan, A., Kumar, P., et al. (2023). Effectiveness of mining audio and text pairs from public data for improving ASR systems for low-resource languages. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.

Bhogale, K. S., Sundaresan, S., Raman, A., Javed, T., Khapra, M. M., & Kumar, P. (2023). Vistaar: Diverse benchmarks and training sets for Indian language ASR. arXiv preprint arXiv:2305.15386.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Bu, H., Du, J., Na, X., Wu, B., & Zheng, H. (2017). Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech i/o systems and assessment* (pp. 1–5). IEEE.

Burchi, M., & Vielzeuf, V. (2021). Efficient conformer: Progressive downsampling and grouped attention for automatic speech recognition. In *2021 IEEE automatic speech recognition and understanding workshop* (pp. 8–15). IEEE.

Cattoni, R., Di Gangi, M. A., Bentivogli, L., Negri, M., & Turchi, M. (2021). Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech and Language*, *66*, Article 101155.

Cettolo, M., Girardi, C., & Federico, M. (2012). Wit3: Web inventory of transcribed and translated talks. In *Conference of European association for machine translation* (pp. 261–268).

Chan, W., Jaitly, N., Le, Q., & Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE international conference on acoustics, speech and signal processing* (pp. 4960–4964). IEEE.

Chen, G., Chai, S., Wang, G., Du, J., Zhang, W.-Q., Weng, C., et al. (2021). Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. arXiv preprint arXiv:2106.06909.

Chen, D., & Mak, B. K.-W. (2015). Multitask learning of deep neural networks for low-resource speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(7), 1172–1183.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., et al. (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1505–1518.

Chen, W., Xing, X., Xu, X., Pang, J., & Du, L. (2022). Speechformer: A hierarchical efficient framework incorporating the characteristics of speech. arXiv preprint arXiv:2203.03812.

Chen, Y.-C., Yang, Z., Yeh, C.-F., Jain, M., & Seltzer, M. L. (2020). Aipnet: Generative adversarial pre-training of accent-invariant networks for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 6979–6983). IEEE.

Chen, Z., & Zhang, W. (2023). End-to-end speech recognition with reinforcement learning. In *Eighth international conference on electronic technology and information science, vol. 12715* (pp. 392–398). SPIE.

Cho, J., Baskar, M. K., Li, R., Wiesner, M., Mallidi, S. H., Yalta, N., et al. (2018). Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling. In *2018 IEEE spoken language technology workshop* (pp. 521–527). IEEE.

Chorowski, J., Bahdanau, D., Cho, K., & Bengio, Y. (2014). End-to-end continuous speech recognition using attention-based recurrent NN: First results. arXiv preprint arXiv:1412.1602.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. *Advances in Neural Information Processing Systems*, *28*.

Chung, Y.-A., Weng, W.-H., Tong, S., & Glass, J. (2019). Towards unsupervised speech-to-text translation. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing* (pp. 7170–7174). IEEE.

Conneau, A., Ma, M., Khanuja, S., Zhang, Y., Axelrod, V., Dalmia, S., et al. (2023). Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE spoken language technology workshop* (pp. 798–805). IEEE.

Cui, J., Kingsbury, B., Ramabhadran, B., Saon, G., Sercu, T., Audhkhasi, K., et al. (2017). Knowledge distillation across ensembles of multilingual models for low-resource languages. In *2017 IEEE international conference on acoustics, speech and signal processing* (pp. 4825–4829). IEEE.

Cui, J., Kingsbury, B., Ramabhadran, B., Sethy, A., Audhkhasi, K., Cui, X., et al. (2015). Multilingual representations for low resource speech recognition and keyword search. In *2015 IEEE workshop on automatic speech recognition and understanding* (pp. 259–266). IEEE.

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2011). Large vocabulary continuous speech recognition with context-dependent DBN-hmms. In *2011 IEEE international conference on acoustics, speech and signal processing* (pp. 4688–4691). IEEE.

Dash, D., Kim, M. J., Teplansky, K., & Wang, J. (2018). Automatic speech recognition with articulatory information and a unified dictionary for Hindi, Marathi, Bengali and Oriya. In *INTERSPEECH* (pp. 1046–1050).

Deng, L., Hinton, G., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8599–8603). IEEE.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dhanjal, A. S., & Singh, W. (2023). A comprehensive survey on automatic speech recognition using neural networks. *Multimedia Tools and Applications*, 1–46.

Dida, H. A., Chakravarthy, D., & Rabbi, F. (2023). ChatGPT and big data: Enhancing text-to-speech conversion. *Mesopotamian Journal of Big Data*, *2023*, 31–35. http://dx.doi.org/10.58496/MJBD/2023/005, URL https://mesopotamian.press/journals/index.php/bigdata/article/view/47.

Diwan, A., Vaideeswaran, R., Shah, S., Singh, A., Raghavan, S., Khare, S., et al. (2021). Multilingual and code-switching ASR challenges for low resource Indian languages. arXiv preprint arXiv:2104.00235.

Dong, L., Xu, S., & Xu, B. (2018). Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 5884–5888). IEEE.

Dua, M., Akanksha, & Dua, S. (2023). Noise robust automatic speech recognition: review and analysis. *International Journal of Speech Technology*, 1–45.

Dudziak, Ł., Abdelfattah, M. S., Vipperla, R., Laskaridis, S., & Lane, N. D. (2019). Shrinkml: End-to-end asr model compression using reinforcement learning. arXiv preprint arXiv:1907.03540.

Elmer, S., Kurthen, I., Meyer, M., & Giroud, N. (2023). A multidimensional characterization of the neurocognitive architecture underlying age-related temporal speech processing. *NeuroImage*, *278*, Article 120285.

Fan, P., Guo, D., Zhang, J., Yang, B., & Lin, Y. (2023). Enhancing multilingual speech recognition in air traffic control by sentence-level language identification. arXiv preprint arXiv:2305.00170.

Farooq, M. U., Adeeba, F., Rauf, S., & Hussain, S. (2019). Improving large vocabulary urdu speech recognition system using deep neural networks. In *Interspeech* (pp. 2978–2982).

Gaikwad, S. K., Gawali, B. W., & Yannawar, P. (2010). A review on speech recognition technique. *International Journal of Computer Applications*, *10*(3), 16–24.

Gandhi, S., von Platen, P., & Rush, A. M. (2023). Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling. arXiv:2311.00430.

Ghoshal, A., Swietojanski, P., & Renals, S. (2013). Multilingual training of deep neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 7319–7323). IEEE.

Graves, A. (2012). Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning* (pp. 1764–1772). PMLR.

Graves, A., Jaitly, N., & Mohamed, A.-r. (2013). Hybrid speech recognition with deep bidirectional LSTM. In *2013 IEEE workshop on automatic speech recognition and understanding* (pp. 273–278). IEEE.

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645–6649). Ieee.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., et al. (2020). Conformer: Convolution-augmented transformer for speech recognition. arXiv preprint arXiv:2005.08100.

Guo, P., Chang, X., Watanabe, S., & Xie, L. (2021). Multi-speaker ASR combining non-autoregressive conformer CTC and conditional speaker chain. arXiv preprint arXiv:2106.08595.

Guo, J., Tiwari, G., Droppo, J., Van Segbroeck, M., Huang, C.-W., Stolcke, A., et al. (2020). Efficient minimum word error rate training of rnn-transducer for end-to-end speech recognition. arXiv preprint arXiv:2007.13802.

Gupta, R., Mundra, J., Mahajan, D., & Modi, A. (2021). MCL@ IITK at SemEval-2021 task 2: Multilingual and cross-lingual word-in-context disambiguation using augmented data, signals, and transformers. arXiv preprint arXiv:2104.01567.

Hadian, H., Sameti, H., Povey, D., & Khudanpur, S. (2018). End-to-end speech recognition using lattice-free MMI. In *Interspeech* (pp. 12–16).

Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., et al. (2020). Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. arXiv preprint arXiv:2005.03191.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., et al. (2014). Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.

Harish, B., & Rangan, R. K. (2020). A comprehensive survey on Indian regional language processing. *SN Applied Sciences*, *2*(7), 1–16.

Hayashi, T., Yamamoto, R., Inoue, K., Yoshimura, T., Watanabe, S., Toda, T., et al. (2020). ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 7654–7658). IEEE.

Heigold, G., Vanhoucke, V., Senior, A., Nguyen, P., Ranzato, M., Devin, M., et al. (2013). Multilingual acoustic models using distributed deep neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8619–8623). IEEE.

Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., & Esteve, Y. (2018). TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In *International conference on speech and computer* (pp. 198–208). Springer.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Hou, W., Dong, Y., Zhuang, B., Yang, L., Shi, J., & Shinozaki, T. (2020). Large-scale end-to-end multilingual speech recognition and language identification with multi-task learning. *Babel*, *37*(4k), 10k.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3451–3460.

Huang, J.-T., Li, J., Yu, D., Deng, L., & Gong, Y. (2013). Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 7304–7308). IEEE.

Hussein, A., Chowdhury, S., & Ali, A. (2021). KARI: Kanari/QCRI's end-to-end systems for the INTERSPEECH 2021 Indian languages code-switching challenge. arXiv preprint arXiv:2106.05885.

Hwang, J.-W., Park, J., Park, R.-H., & Park, H.-M. (2023). Audio-visual speech recognition based on joint training with audio-visual speech enhancement for robust speech recognition. *Applied Acoustics*, *211*, Article 109478.

Inaguma, H., Kiyono, S., Duh, K., Karita, S., Soplin, N. E. Y., Hayashi, T., et al. (2020). Espnet-ST: All-in-one speech translation toolkit. arXiv preprint arXiv:2004.10234.

Iranzo-Sánchez, J., Silvestre-Cerda, J. A., Jorge, J., Roselló, N., Giménez, A., Sanchis, A., et al. (2020). Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 8229–8233). IEEE.

Javadpour, A., Ja'fari, F., Taleb, T., & Benzaïd, C. (2023). Reinforcement learning-based slice isolation against DDoS attacks in beyond 5G networks. *IEEE Transactions on Network and Service Management*, *20*(3), 3930–3946. http://dx.doi.org/10.1109/TNSM.2023.3254581.

Javed, T., Doddapaneni, S., Raman, A., Bhogale, K. S., Ramesh, G., Kunchukuttan, A., et al. (2022). Towards building asr systems for the next billion users. In *Proceedings of the AAAI conference on artificial intelligence, vol. 36, no. 10* (pp. 10813–10821).

Jin, Z., Xie, X., Wang, T., Geng, M., Deng, J., Li, G., et al. (2024). Towards automatic data augmentation for disordered speech recognition. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing* (pp. 10626–10630). IEEE.

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., et al. (2020). Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 7669–7673). IEEE.

Kakwani, D., Kunchukuttan, A., Golla, S., Gokul, N., Bhattacharyya, A., Khapra, M. M., et al. (2020). IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the association for computational linguistics* EMNLP 2020, (pp. 4948–4961).

Kalluri, S. B., Vijayasenan, D., Ganapathy, S., Krishnan, P., et al. (2021). NISP: A multi-lingual multi-accent dataset for speaker profiling. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing* (pp. 6953–6957). IEEE.

Kang, X. (2024). Speech emotion recognition algorithm of intelligent robot based on ACO-SVM. *International Journal of Cognitive Computing in Engineering*.

Kannan, A., Datta, A., Sainath, T. N., Weinstein, E., Ramabhadran, B., Wu, Y., et al. (2019). Large-scale multilingual speech recognition with a streaming end-to-end model. arXiv preprint arXiv:1909.05330.

Karafiát, M., Baskar, M. K., Watanabe, S., Hori, T., Wiesner, M., Černocký, J., et al. (2018). Analysis of multilingual sequence-to-sequence speech recognition systems. arXiv preprint arXiv:1811.03451.

Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., et al. (2019). A comparative study on transformer vs rnn in speech applications. In *2019 IEEE automatic speech recognition and understanding workshop* (pp. 449–456). IEEE.

Karmakar, P., Teng, S. W., & Lu, G. (2021). Thank you for attention: a survey on attention-based artificial neural networks for automatic speech recognition. arXiv preprint arXiv:2102.07259.

Kaur, A. P., Singh, A., Sachdeva, R., & Kukreja, V. (2023). Automatic speech recognition systems: A survey of discriminative techniques. *Multimedia Tools and Applications*, *82*(9), 13307–13339.

Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., et al. (2021). Muril: Multilingual representations for indian languages. arXiv preprint arXiv:2103.10730.

Kheddar, H., Hemis, M., & Himeur, Y. (2024). Automatic speech recognition using advanced deep learning approaches: A survey. *Information Fusion*, Article 102422.

Kim, S., Gholami, A., Shaw, A., Lee, N., Mangalam, K., Malik, J., et al. (2022). Squeezeformer: An efficient transformer for automatic speech recognition. arXiv preprint arXiv:2206.00888.

Kim, S., Kim, G., Shin, S., & Lee, S. (2021). Two-stage textual knowledge distillation for end-to-end spoken language understanding. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing* (pp. 7463–7467). IEEE.

Kim, K., Wu, F., Peng, Y., Pan, J., Sridhar, P., Han, K. J., et al. (2023). E-branchformer: Branchformer with enhanced merging for speech recognition. In *2022 IEEE spoken language technology workshop* (pp. 84–91). IEEE.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and Software Technology*, *55*(12), 2049–2075.

Ko, T., Peddinti, V., Povey, D., & Khudanpur, S. (2015). Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association*.

Kolobov, R., Okhapkina, O., Omelchishina, O., Platunov, A., Bedyakin, R., Moshkin, V., et al. (2021). Mediaspeech: Multilanguage asr benchmark and dataset. arXiv preprint arXiv:2103.16193.

Krishna, D. (2021). A dual-decoder conformer for multilingual speech recognition. CoRR.

Kuchaiev, O., Li, J., Nguyen, H., Hrinchuk, O., Leary, R., Ginsburg, B., et al. (2019). Nemo: a toolkit for building ai applications using neural modules. arXiv preprint arXiv:1909.09577.

Kumar, M. G., Kuriakose, J., Thyagachandran, A., Seth, A., Prasad, L. D., Jaiswal, S., et al. (2021). Dual script E2E framework for multilingual and code-switching ASR. arXiv preprint arXiv:2106.01400.

Kunchukuttan, A., Kakwani, D., Golla, S., Bhattacharyya, A., Khapra, M. M., Kumar, P., et al. (2020). Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. arXiv preprint arXiv:2005.00085.

Kwon, Y., & Chung, S.-W. (2023). MoLE: Mixture of language experts for multi-lingual automatic speech recognition. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.

Lam, T. K., Schamoni, S., & Riezler, S. (2023). Make more of your data: Minimal effort data augmentation for automatic speech recognition and translation. In *ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing* (pp. 1–5). IEEE.

Lam, M. W., Tian, Q., Li, T., Yin, Z., Feng, S., Tu, M., et al. (2024). Efficient neural music generation. *Advances in Neural Information Processing Systems*, *36*.

Lamere, P., Kwok, P., Walker, W., Gouvêa, E. B., Singh, R., Raj, B., et al. (2003). Design of the CMU sphinx-4 decoder. In *Interspeech*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Latif, S., Zaidi, A., Cuayahuitl, H., Shamshad, F., Shoukat, M., & Qadir, J. (2023). Transformers in speech processing: A survey. arXiv preprint arXiv:2303.11607.

Le, H., Barbier, F., Nguyen, H., Tomashenko, N., Mdhaffar, S., Gahbiche, S., et al. (2021). ON-TRAC'systems for the IWSLT 2021 low-resource speech translation and multilingual speech translation shared tasks. In *International conference on spoken language translation*.

Le, H., Pino, J., Wang, C., Gu, J., Schwab, D., & Besacier, L. (2020). Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. arXiv preprint arXiv:2011.00747.

Lee, A., Kawahara, T., & Shikano, K. (2001). Julius—an open source real-time large vocabulary recognition engine.

Lekshmi, K., & Sherly, E. (2016). Automatic speech recognition using different neural network architectures–a survey. *International Journal of Computer Science and Information Technologies*, *7*(6), 242–248.

Lewis, P. M. A. (2009). Ethnologue : languages of the world.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., et al. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Li, Y., Anumanchipalli, G. K., Mohamed, A., Chen, P., Carney, L. H., Lu, J., et al. (2023). Dissecting neural computations in the human auditory pathway using deep neural networks for speech. *Nature Neuroscience*, *26*(12), 2213–2225.

Li, M., & Doddipatla, R. (2023). Non-autoregressive end-to-end approaches for joint automatic speech recognition and spoken language understanding. In *2022 IEEE spoken language technology workshop* (pp. 390–397). IEEE.

Li, X., Wang, C., Tang, Y., Tran, C., Tang, Y., Pino, J., et al. (2020). Multilingual speech translation with efficient finetuning of pretrained models. arXiv preprint arXiv:2010.12829.

Li, J., et al. (2022). Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, *11*(1).

Liang, S., & Yan, W. Q. (2022). A hybrid CTC+ Attention model based on end-to-end framework for multilingual speech recognition. *Multimedia Tools and Applications*, *81*(28), 41295–41308.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81).

Linguistic Data Consortium (2022). LDC catalog. https://catalog.ldc.upenn.edu/.

Liu, Y., Li, T., Zhang, P., & Yan, Y. (2021). Improved conformer-based end-to-end speech recognition using neural architecture search. arXiv preprint arXiv:2104.05390.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Lu, Z., Cao, L., Zhang, Y., Chiu, C.-C., & Fan, J. (2020). Speech sentiment analysis via pre-trained features from end-to-end asr models. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 7149–7153). IEEE.

Luo, S., Rabbani, Q., & Crone, N. E. (2023). Brain-computer interface: applications to speech decoding and synthesis to augment communication. *Neurotherapeutics*, *19*(1), 263–273.

Malik, M., Malik, M. K., Mehmood, K., & Makhdoom, I. (2021). Automatic speech recognition: a survey. *Multimedia Tools and Applications*, *80*(6), 9411–9457.

Mamyrbayev, O. Z., Oralbekova, D. O., Alimhan, K., & Nuranbayeva, B. M. (2023). Hybrid end-to-end model for Kazakh speech recognition. *International Journal of Speech Technology*, *26*(2), 261–270.

Mehrish, A., Majumder, N., Bharadwaj, R., Mihalcea, R., & Poria, S. (2023). A review of deep learning techniques for speech processing. *Information Fusion*, Article 101869.

Miao, Y., Gowayyed, M., & Metze, F. (2015). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *2015 IEEE workshop on automatic speech recognition and understanding* (pp. 167–174). IEEE.

Moriya, T., Sato, H., Ochiai, T., Delcroix, M., & Shinozaki, T. (2023). Streaming end-to-end target-speaker automatic speech recognition and activity detection. *IEEE Access*, *11*, 13906–13917.

Morris, A. C., Maier, V., & Green, P. (2004). From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *Eighth international conference on spoken language processing*.

Mridha, M. F., Ohi, A. Q., Hamid, M. A., & Monowar, M. M. (2022). A study on the challenges and opportunities of speech recognition for bengali language. *Artificial Intelligence Review*, *55*(4), 3431–3455.

Müller, M., & Waibel, A. (2015). Using language adaptive deep neural networks for improved multilingual speech recognition. In *Proceedings of the 12th international workshop on spoken language translation: papers*.

Mustafa, M. B., Yusoof, M. A., Khalaf, H. K., Rahman Mahmoud Abushariah, A. A., Kiah, M. L. M., Ting, H. N., et al. (2022). Code-switching in automatic speech recognition: The issues and future directions. *Applied Sciences*, *12*(19), 9541.

Napoles, C., Sakaguchi, K., Post, M., & Tetreault, J. (2016). GLEU without tuning. arXiv preprint arXiv:1605.02592.

Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE Access*, *7*, 19143–19165.

Nguyen, T. A., Sagot, B., & Dupoux, E. (2022). Are discrete units necessary for spoken language modeling? *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1415–1423.

Nguyen, T.-S., Stüker, S., & Waibel, A. (2020). Toward cross-domain speech recognition with end-to-end models. arXiv preprint arXiv:2003.04194.

Nowakowski, K., Ptaszynski, M., Murasaki, K., & Nieuważny, J. (2023). Adapting multilingual speech representation model for a new, underresourced language through multilingual fine-tuning and continued pretraining. *Information Processing & Management*, *60*(2), Article 103148.

Oghim, S., Park, J., Bang, H., & Leeghim, H. (2024). Deep reinforcement learning-based attitude control for spacecraft using control moment gyros. *Advances in Space Research*.

O'Neill, P. K., Lavrukhin, V., Majumdar, S., Noroozi, V., Zhang, Y., Kuchaiev, O., et al. (2021). Spgispeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition. arXiv preprint arXiv:2104.02014.

OpenAI (2023). GPT-4 technical report. arXiv.

Oruh, J., Viriri, S., & Adegun, A. (2022). Long short-term memory recurrent neural network for automatic speech recognition. *IEEE Access*, *10*, 30069–30079.

Padmanabhan, J., & Johnson Premkumar, M. J. (2015). Machine learning in automatic speech recognition: A survey. *IETE Technical Review*, *32*(4), 240–251.

Pan, S. (2024). Emotional analysis of broadcasting and hosting speech by integrating grid PSO-SVR and PAD models. *International Journal of Cognitive Computing in Engineering*.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing* (pp. 5206–5210). IEEE.

Papastratis, I. (2021). Speech recognition: a review of the different deep learning approaches. https://Theaisummer.Com/.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318).

Peterson, K., Tong, A., & Yu, Y. (2022). OpenASR21: The second open challenge for automatic speech recognition of low-resource languages. In *Proc. Interspeech 2022* (pp. 4895–4899).

Pham, N.-Q., Nguyen, T. N., Ha, T.-L., Stüker, S., Waibel, A., & He, D. (2021). Multilingual speech translation KIT@ IWSLT2021. In *Proceedings of the 18th international conference on spoken language translation* (pp. 154–159).

Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the third conference on machine translation: research papers* (pp. 186–191). Belgium, Brussels: Association for Computational Linguistics, URL https://www.aclweb.org/anthology/W18-6319.

Potapczyk, T., & Przybysz, P. (2020). Srpol's system for the IWSLT 2020 end-to-end speech translation task. In *Proceedings of the 17th international conference on spoken language translation* (pp. 89–94).

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., et al. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.

Prabhavalkar, R., Hori, T., Sainath, T. N., Schlüter, R., & Watanabe, S. (2023). End-to-end speech recognition: A survey. arXiv preprint arXiv:2303.03329.

Pratap, V., Hannun, A., Xu, Q., Cai, J., Kahn, J., Synnaeve, G., et al. (2019). Wav2letter++: A fast open-source speech recognition system. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing* (pp. 6460–6464). IEEE.

Pratap, V., Tjandra, A., Shi, B., Tomasello, P., Babu, A., Kundu, S., et al. (2023). Scaling speech technology to 1,000+ languages. arXiv preprint arXiv:2305.13516.

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., & Collobert, R. (2020). Mls: A large-scale multilingual dataset for speech research. arXiv preprint arXiv:2012.03411.

Pulugundla, B., Baskar, M. K., Kesiraju, S., Egorova, E., Karafiát, M., Burget, L., et al. (2018). BUT system for low resource Indian language ASR. In *Interspeech* (pp. 3182–3186).

Qamar, R., & Zardari, B. A. (2023). Artificial neural networks: An overview, vol. 2023. (pp. 124–133). http://dx.doi.org/10.58496/MJCSC/2023/015, URL https://mesopotamian.press/journals/index.php/cs/article/view/118.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *International conference on machine learning* (pp. 28492–28518). PMLR.

Rao, K., Sak, H., & Prabhavalkar, R. (2017). Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In *2017 IEEE automatic speech recognition and understanding workshop* (pp. 193–199). IEEE.

Ravanelli, M., Parcollet, T., & Bengio, Y. (2019). The pytorch-kaldi speech recognition toolkit. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing* (pp. 6465–6469). IEEE.

Regmi, S., & Bal, B. K. (2021). An end-to-end speech recognition for the nepali language. In *Proceedings of the 18th international conference on natural language processing* (pp. 180–185).

Reitmaier, T., Wallington, E., Kalarikalayil Raju, D., Klejch, O., Pearson, J., Jones, M., et al. (2022). Opportunities and challenges of automatic speech recognition systems for low-resource language speakers. In *CHI conference on human factors in computing systems* (pp. 1–17).

Reza, S., Ferreira, M. C., Machado, J., & Tavares, J. M. R. (2023). A customized residual neural network and bi-directional gated recurrent unit-based automatic speech recognition model. *Expert Systems with Applications*, *215*, Article 119293.

Rybach, D., Gollan, C., Heigold, G., Hoffmeister, B., Lööf, J., Schlüter, R., et al. (2009). The RWTH aachen university open source speech recognition system. In *Tenth annual conference of the international speech communication association*.

Sailor, H. B., & Hain, T. (2020). Multilingual speech recognition using language-specific phoneme recognition as auxiliary task for Indian languages. In *INTERSPEECH* (pp. 4756–4760).

Sainath, T. N., Vinyals, O., Senior, A., & Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing* (pp. 4580–4584). IEEE.

Sanabria, R., Caglayan, O., Palaskar, S., Elliott, D., Barrault, L., Specia, L., et al. (2018). How2: a large-scale dataset for multimodal language understanding. arXiv preprint arXiv:1811.00347.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Saon, G., Tüske, Z., Bolanos, D., & Kingsbury, B. (2021). Advancing RNN transducer technology for speech recognition. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing* (pp. 5654–5658). IEEE.

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). Wav2vec: Unsupervised pre-training for speech recognition. arXiv preprint arXiv:1904.05862.

Sen, B., Agarwal, A., Ganesh, M. S., & Vuppala, A. K. (2021). Reed: An approach towards quickly bootstrapping multilingual acoustic models. In *2021 IEEE spoken language technology workshop* (pp. 272–279). IEEE.

Sercu, T., Saon, G., Cui, J., Cui, X., Ramabhadran, B., Kingsbury, B., et al. (2017). Network architectures for multilingual speech representation learning. In *2017 IEEE international conference on acoustics, speech and signal processing* (pp. 5295–5299). IEEE.

Shah, S., Guha, S., Khanuja, S., & Sitaram, S. (2020). Cross-lingual and multilingual spoken term detection for low-resource Indian languages. arXiv preprint arXiv:2011.06226.

Shetty, V. M., & Mary, N. J. M. S. (2020). Improving the performance of transformer based low resource speech recognition for Indian languages. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 8279–8283). IEEE.

Shor, J., Bi, R. A., Venugopalan, S., Ibara, S., Goldenberg, R., & Rivlen, E. (2023). Clinical BERTScore: An improved measure of automatic speech recognition performance in clinical settings. arXiv preprint arXiv:2303.05737.

Singh, M. T., Fayjie, A., & Kachari, B. (2015). A survey report on speech recognition system. *International Journal of Computer Applications*, *121*, 1–3.

Singh, A., Kadyan, V., Kumar, M., & Bassan, N. (2020). ASRoIL: a comprehensive survey for automatic speech recognition of Indian languages. *Artificial Intelligence Review*, *53*(5), 3673–3704.

Singh, V. P., Malato, F., Hautamaki, V., Sahidullah, M., & Kinnunen, T. (2024). ROAR: Reinforcing original to augmented data ratio dynamics for Wav2Vec2. 0 based ASR. arXiv preprint arXiv:2406.09999.

Singh, A. P., Nath, R., & Kumar, S. (2018). A survey: Speech recognition approaches and techniques. In *2018 5th IEEE Uttar Pradesh section international conference on electrical, electronics and computer engineering* (pp. 1–4). IEEE.

Singh, R., Puri, H., Aggarwal, N., & Gupta, V. (2020). An efficient language-independent acoustic emotion classification system. *Arabian Journal for Science and Engineering*, *45*, 3111–3121.

Singh, L., Singh, S., & Aggarwal, N. (2018a). Improved TOPSIS method for peak frame selection in audio-video human emotion recognition. *Multimedia Tools and Applications*, *78*, 6277–6308.

Singh, L., Singh, S., & Aggarwal, N. (2018b). Two-stage text feature selection method for human emotion recognition. In *Proceedings of 2nd International Conference on Communication, Computing and Networking*.

Singh, L., Singh, S., Aggarwal, N., Singh, R., & Singla, G. (2021). An efficient temporal feature aggregation of audio-video signals for human emotion recognition. In *2021 6th International Conference on Signal Processing, Computing and Control* (pp. 660–668).

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th conference of the association for machine translation in the Americas: technical papers* (pp. 223–231).

Srivastava, B. M. L., Sitaram, S., Mehta, R. K., Mohan, K. D., Matani, P., Satpal, S., et al. (2018). Interspeech 2018 low resource automatic speech recognition challenge for Indian languages. In *SLTU* (pp. 11–14).

Tang, Y., Gong, H., Li, X., Wang, C., Pino, J., Schwenk, H., et al. (2021). FST: the FAIR speech translation system for the IWSLT21 multilingual shared task. arXiv preprint arXiv:2107.06959.

Thomas, S., Ganapathy, S., & Hermansky, H. (2012). Multilingual MLP features for low-resource LVCSR systems. In *2012 IEEE international conference on acoustics, speech and signal processing* (pp. 4269–4272). IEEE.

Thomas, B., Kessler, S., & Karout, S. (2022). Efficient adapter transfer of self-supervised speech models for automatic speech recognition. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 7102–7106). IEEE.

Tjandra, A., Choudhury, D. G., Zhang, F., Singh, K., Conneau, A., Baevski, A., et al. (2022). Improved language identification through cross-lingual self-supervised learning. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 6877–6881). IEEE.

Tjandra, A., Sakti, S., & Nakamura, S. (2018). Sequence-to-sequence ASR optimization via reinforcement learning. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 5829–5833). IEEE.

Toshniwal, S., Sainath, T. N., Weiss, R. J., Li, B., Moreno, P., Weinstein, E., et al. (2018). Multilingual speech recognition with a single end-to-end model. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 4904–4908). IEEE.

Trentin, E., & Gori, M. (2001). A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing*, *37*(1–4), 91–126.

Tüske, Z., Pinto, J., Willett, D., & Schlüter, R. (2013). Investigation on cross-and multilingual MLP features under matched and mismatched acoustical conditions. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 7349–7353). IEEE.

Vadwala, A. Y., Suthar, K. A., Karmakar, Y. A., Pandya, N., & Patel, B. (2017). Survey paper on different speech recognition algorithm: challenges and techniques. *International Journal of Computational Application*, *175*(1), 31–36.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Veaux, C., Yamagishi, J., & MacDonald, K. (2017). CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit.

Veselỳ, K., Karafiát, M., Grézl, F., Janda, M., & Egorova, E. (2012). The language-independent bottleneck features. In *2012 IEEE spoken language technology workshop* (pp. 336–341). IEEE.

Vuddagiri, R. K., Gurugubelli, K., Jain, P., Vydana, H. K., & Vuppala, A. K. (2018). IIITH-ILSC speech database for Indain language identification. In *SLTU* (pp. 56–60).

Wali, A., Alamgir, Z., Karim, S., Fawaz, A., Ali, M. B., Adan, M., et al. (2022). Generative adversarial networks for speech processing: A review. *Computer Speech and Language*, *72*, Article 101308.

Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., et al. (2020). Transformer-based acoustic modeling for hybrid speech recognition. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 6874–6878). IEEE.

Wang, C., Pino, J., Wu, A., & Gu, J. (2020). Covost: A diverse multilingual speech-to-text translation corpus. arXiv preprint arXiv:2002.01320.

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., et al. (2021). Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. arXiv preprint arXiv:2101.00390.

Wang, C., Wu, A., & Pino, J. (2020). Covost 2 and massively multilingual speech-to-text translation. arXiv preprint arXiv:2007.10310.

Watanabe, S., Boyer, F., Chang, X., Guo, P., Hayashi, T., Higuchi, Y., et al. (2021). The 2020 espnet update: new features, broadened applications, performance improvements, and future plans. In *2021 IEEE data science and learning workshop* (pp. 1–6). IEEE.

Watanabe, S., Hori, T., & Hershey, J. R. (2017). Language independent end-to-end architecture for joint language identification and speech recognition. In *2017 IEEE automatic speech recognition and understanding workshop* (pp. 265–271). IEEE.

Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., et al. (2018). Espnet: End-to-end speech processing toolkit. arXiv preprint arXiv:1804.00015.

Wei, G., Duan, Z., Li, S., Yu, X., & Yang, G. (2023). LFEformer: Local feature enhancement using sliding window with deformability for automatic speech recognition. *IEEE Signal Processing Letters*, *30*, 180–184.

Weiss, R. J., Chorowski, J., Jaitly, N., Wu, Y., & Chen, Z. (2017). Sequence-to-sequence models can directly translate foreign speech. arXiv preprint arXiv:1703.08581.

Xu, Q., Baevski, A., Likhomanenko, T., Tomasello, P., Conneau, A., Collobert, R., et al. (2021). Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing* (pp. 3030–3034). IEEE.

Yadav, A. K., Kumar, M., Kumar, A., Shivani, Kusum, & Yadav, D. (2023). Hate speech recognition in multilingual text: Hinglish documents. *International Journal of Information Technology*, *15*(3), 1319–1331.

Yadav, H., & Sitaram, S. (2022). A survey of multilingual models for automatic speech recognition. arXiv preprint arXiv:2202.12576.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, *32*.

Yang, F., Yang, M., Li, X., Wu, Y., Zhao, Z., Raj, B., et al. (2024). A closer look at reinforcement learning-based automatic speech recognition. *Computer Speech and Language*, *87*, Article 101641.

Yao, Z., Wu, D., Wang, X., Zhang, B., Yu, F., Yang, C., et al. (2021). Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. arXiv preprint arXiv:2102.01547.

Yu, S.-I., Jiang, L., & Hauptmann, A. (2014). Instructional videos for unsupervised harvesting and learning of action examples. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 825–828).

Zeng, X., Li, L., & Liu, Q. (2021). Multilingual speech translation with unified transformer: Huawei Noah's ark lab at IWSLT 2021. arXiv preprint arXiv:2106.00197.

Zhang, Y., Han, W., Qin, J., Wang, Y., Bapna, A., Chen, Z., et al. (2023). Google usm: Scaling automatic speech recognition beyond 100 languages. arXiv preprint arXiv:2303.01037.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675.

Zhang, C., Li, B., Sainath, T., Strohman, T., Mavandadi, S., Chang, S.-y., et al. (2022). Streaming end-to-end multilingual speech recognition with joint language identification. arXiv preprint arXiv:2209.06058.

Zhang, J.-X., Ling, Z.-H., Jiang, Y., Liu, L.-J., Liang, C., & Dai, L.-R. (2019). Improving sequence-to-sequence voice conversion by adding text-supervision. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing* (pp. 6785–6789). IEEE.

Zhang, J.-X., Ling, Z.-H., Liu, L.-J., Jiang, Y., & Dai, L.-R. (2019). Sequence-to-sequence acoustic modeling for voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27*(3), 631–644.

Zhang, J., Peng, Y., Van Tung, P., Xu, H., Huang, H., & Chng, E. S. (2021). E2e-based multi-task learning approach to joint speech and accent recognition. arXiv preprint arXiv:2106.08211.

Zhang, Y., Qin, J., Park, D. S., Han, W., Chiu, C.-C., Pang, R., et al. (2020). Pushing the limits of semi-supervised learning for automatic speech recognition. arXiv preprint arXiv:2010.10504.

Zhang, B., & Sennrich, R. (2021). *Edinburgh's end-to-end multilingual speech translation system for IWSLT 2021*. ACL Anthology.

Zhang, H., Shi, K., & Chen, N. F. (2021). Multilingual speech evaluation: Case studies on English, Malay and Tamil. arXiv preprint arXiv:2107.03675.

Zhao, Y., Wakita, H., & Zhuang, X. (1991). An HMM based speaker-independent continuous speech recognition system with experiments on the TIMIT DATABASE. In *ICASSP 91: 1991 international conference on acoustics, speech, and signal processing* (pp. 333–336). IEEE.

Zhao, J., & Zhang, W.-Q. (2022). Improving automatic speech recognition performance for low-resource languages with self-supervised models. *IEEE Journal of Selected Topics in Signal Processing, 16*(6), 1227–1241.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., et al. (2023). A survey of large language models. arXiv preprint arXiv:2303.18223.

Zhou, L., Li, J., Sun, E., & Liu, S. (2022). A configurable multilingual model is all you need to recognize all languages. In *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing* (pp. 6422–6426). IEEE.