Speech Communication 167 (2025) 103151

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

An overview of high-resource automatic speech recognition methods and their empirical evaluation in low-resource environments

Kavan Fatehi [a,*], Mercedes Torres Torres [b], Ayse Kucukyilmaz [a]

[a] School of Computer Science, University of Nottingham, United Kingdom
[b] B-hive Innovations, United Kingdom

ARTICLE INFO

Keywords:
Automatic speech recognition
End-to-end model
Deep learning models
Low-resource environment

ABSTRACT

Deep learning methods for Automatic Speech Recognition (ASR) often rely on large-scale training datasets, which are typically unavailable in low-resource environments (LREs). This lack of sufficient and representative training data poses a significant challenge for applying ASR systems in specific domains categorized as LREs. In this paper, we provide a comprehensive overview and empirical analysis of state-of-the-art deep learning techniques for ASR, which are primarily designed for high-resource environments (HREs). Our aim is to explore their potential effectiveness in LRE settings. We focus on identifying key factors that influence the adaptation of HRE models to LRE tasks. To this end, we survey advanced deep learning models and conduct a comparative evaluation of their performance in LRE contexts. Additionally, we propose that pre-training ASR models on HRE datasets, followed by domain-specific fine-tuning on LRE data, can significantly enhance performance in data-scarce settings. Using LibriSpeech and WSJ as our HRE datasets, we evaluate these models on two LRE datasets: UASpeech for dysarthria speech and iCUBE, our novel human–robot interaction dataset. Our systematic experiments, involving varying dataset sizes for pre-training, demonstrate the efficacy of combining pre-training and fine-tuning strategies to improve recognition accuracy in LREs.

Contents

* Corresponding author.
E-mail addresses: kavan.fatehi@nottingham.ac.uk (K. Fatehi), mtorrestorres@b-hiveinnovations.co.uk (M. Torres Torres), ayse.kucukyilmaz@nottingham.ac.uk (A. Kucukyilmaz).

https://doi.org/10.1016/j.specom.2024.103151
Received 4 October 2023; Received in revised form 16 September 2024; Accepted 21 November 2024
Available online 10 December 2024

1. Introduction

Speech is the most natural and widespread form of human communication, as highlighted by Amodei et al. (2016a). This fundamental aspect of human interaction has motivated computer scientists and linguists to develop Automatic Speech Recognition (ASR) systems, which enable machines to effectively interpret and respond to spoken language (Besacier et al., 2014). Since their inception in the 1970s, ASR systems have been a cornerstone of machine learning research (Wang et al., 2019a). Today, they are embedded in various everyday technologies, such as Apple's Siri and Amazon's Echo, serving as integral components of personal assistants and voice-activated services (Roger et al., 2022).

ASR systems span a broad spectrum of tasks, including speech-to-text transcription, text-to-speech generation, speaker identification, and emotion and affect recognition (López-Cózar et al., 2014; Szymański et al., 2020; Besacier et al., 2014). Despite their diverse applications, the primary goal remains the same: to facilitate seamless and natural communication between humans and machines via speech (Amodei et al., 2016a). This paper focuses on ASR systems, which convert spoken language into corresponding textual transcriptions (Bahdanau et al., 2016). Over the past decade, the integration of deep learning techniques has dramatically advanced ASR performance, resulting in more robust and accurate models and enabling a wider array of applications (Nassif et al., 2019; Chan et al., 2016; Alam et al., 2020). Notable progress has been made using Recurrent Neural Networks (RNNs) equipped with Long Short-Term Memory (LSTM) architectures (Hochreiter and Schmidhuber, 1997). Extensive research on LSTM-RNNs (Graves et al., 2013; Sak et al., 2014a,?; Miao et al., 2016) and Convolutional Neural Networks (CNNs) (Sainath et al., 2013; Amodei et al., 2016a) has demonstrated that these models outperform traditional Deep Neural Networks (DNNs) (Yu et al., 2010; Sainath et al., 2011; Jaitly et al., 2012) in a variety of ASR tasks (Yu and Li, 2017).

State-of-the-art results in speech-to-text ASR systems achieve a Word Error Rate (WER) as low as 9.34% (Li et al., 2020) with two-headed cltLSTM, which is trained with transcribed data from a variety of Microsoft products, and 3.6% (Huang et al., 2020) with Conv-Transformer Transducer on Librispeech clean data, while some other models in the area obtain a WER of 9.92% (Li et al., 2019b), and 10.92% (Dong et al., 2018). How such low WERs can be achieved is directly linked to the increase in the amount of annotated data used during the training process: two-headed cltLSTM (Li et al., 2020) use a training set of 65k hours of data, while cltLSTM and Conv-Transformer Transducer (Huang et al., 2020) respectively use training sets of more than 30k hours and more than 1k hours of data. The Neural Speech Recognizer model (Soltau et al., 2017) uses 125k hours of YouTube videos to obtain a WER of 13.5%, and Deep Speech 2 (Amodei et al., 2016a) obtains a WER of 13.59% by using almost 12k hours of English speech and 9.4k hours of Mandarin Chinese speech to train. Therefore, in addition to the amount of training data, in-domain data is also very important in model learning. Using in-domain data can help to ensure that the model learns the most relevant and important patterns from the data, which makes the training process more efficient.

The need for large volumes of data to train deep learning based ASR systems is a commonality in all state-of-the-art techniques. Indeed, having more data typically means improved performance in modern deep learning ASR systems (Roger et al., 2022). For example, the amount of data in the TED-LIUM 3 dataset (Hernandez et al., 2018), which was released in 2018, more than doubled in comparison to the amount of data from the previous release, TED-LIUM 2 (Rousseau et al., 2014), released in 2014. Consequently, the proposed model trained with TED-LIUM 3 was able to achieve better results (Hernandez et al., 2018): By using a classical 3-gram language model used in a beam search on top of the end-to-end architecture, WER decreases to 13.7% with the TED-LIUM 3 training data, while with the TED-LIUM 2 training data, the same model reached a WER of only 20.3% (Hernandez et al., 2018). Furthermore, in Panayotov et al. (2015), the authors achieved an improvement in ASR performance by training the model over LibriSpeech dataset, which contains over 1k hours of speech. The positive effect of larger training datasets in ASR model's performance is also observable with the VoxCeleb2 dataset (Chung et al., 2018). By increasing the number of sentences and participants compared to the previous version of VoxCeleb (Nagrani et al., 2017), the models trained with VoxCeleb2 outperformed the same models previously trained with VoxCeleb.

These high performances being coupled with such high numbers of training samples raise an interesting question from a machine learning perspective: what happens when the training data is limited? For example, what happens if the environment in which these ASR systems will be deployed is highly specialized, such that the usual available corpora (such as Librispeech and WSJ) are not relevant, or when collecting large amounts of data is difficult?

Low-resource environments (LREs), are those that are constrained by limited amount of data to learn a model for different reasons (Meyer, 2019). Examples of these environments include noisy environments, and environments in which speakers have a limited (Juan and Flora, 2015) or a very specialized vocabulary (Wu et al., 2020). Under-resourced languages and domain-specific tasks create limitations for ASR systems that current state-of-the-art models are unable to solve in order to produce high-accuracy output sequences due to the lack of sufficiently large training data. Limited acoustic and text corpora is the main challenge in ASR systems in under-resourced or low-resource areas (Roger et al., 2022).

Other instances of LREs relate to the domain of the ASR system. Domains can be defined as very specific ASR tasks, such as ASR systems for children (Yan, 2018) or the accented speech recognition task (Sun et al., 2018). Researchers have not yet treated domain-specific or domain adaptation in ASR systems in much detail. State-of-the-art ASR models

use popular benchmark datasets, such as Librispeech (Panayotov et al., 2015), WSJ (Paul and Baker, 1992), Fisher (Cieri et al., 2004), and Switchboard (Godfrey et al., 1992) which contain data from general environments (Szymański et al., 2020). The nature of the data has a significant impact on the vocabulary and the form of the conversations. A domain adaptation in the ASR system can be seen in Moore et al. (2018), which has shown that the amount of in-domain data has a direct effect on the accuracy of ASR models when working with conversational topics spoken by people with dysarthria, and corroborated the importance of more inclusive systems. Previous state-of-the-art techniques of ASR systems lack focus on non-native language speakers, so acoustics and linguistics are not considered in the evaluation of the systems (Koenecke et al., 2020). Our experiments show that general benchmark datasets are insufficient for specialized LREs.

There are various techniques to enhance ASR in situations where resources are scarce. These include employing self-supervised methods for learning representations (Ravanelli et al., 2020), using semi-supervised approaches to utilize unlabeled speech data (Kahn et al., 2020), and applying data augmentation strategies (Park et al., 2019). Additionally, pre-training using Autoencoders (Ling et al., 2020) and transfer learning, where knowledge from a well-resourced language is adapted for a language with fewer resources (Kunze et al., 2017), are also effective methods.

Transfer Learning is a method used in model training where a model is initially developed for a specific task in a certain domain. After this initial training, a subset of the parameters, which might sometimes include all the parameters, are then employed to establish a new model for a different domain or task. This approach can be fully applied when the input features and output labels of the new task share the same dimensionality as those of the original task. Transfer learning is one of the most effective solution for addressing challenges in low-resource environments, as demonstrated in several past studies (Kunze et al., 2017).

To begin to tease apart this issue, in this paper, we focus on two fundamental questions surrounding low-resource ASR systems:

1. Can pre-training ASR systems on high-resource data improve the WER in LREs?
2. Can fine-tuning a model trained on high-resource data, with low-resource data in a related domain improve WER rates?

To address these research questions, we investigate the pre-training and fine-tuning of state-of-the-art deep learning models in the context of LREs. Initially, we train the models on two widely-used high-resource environment HRE ASR benchmark datasets – LibriSpeech and WSJ – and evaluate their performance on LRE datasets, namely UASpeech (Kim et al., 2008) and our own low-resource human–robot interaction dataset, iCUBE. Next, we pre-train the same models on the benchmark datasets, followed by fine-tuning with UASpeech and iCUBE to demonstrate the impact of incorporating target-domain data on improving WER in ASR systems. Additionally, in both experimental phases, we train the models using incremental portions of the training data (ranging from 10% to 100%) to emphasize the critical role of both data quantity and quality in optimizing LRE ASR performance.

The main contributions of this paper are summarized as follows:

- We present a comprehensive set of experiments to evaluate the performance of current state-of-the-art HRE ASR models tested over low-resource environments. According to our results, increasing the amount of the training data from another domain is unable to improve the accuracy of an ASR system for low-resource environments.
- Empirical results demonstrate that deeper structures are not efficient as models for LREs. Deeper structures are powerful when we have a large amount of training data.
- We show that pre-training with high resource language and fine-tuning with related in-domain data is an effective way to deal with low-resource ASR tasks.

This paper is organized as follows: Section 2 discusses a brief history and classification of ASR models. Section 3 presents the state of the art in end-to-end deep learning approaches in ASR system. Section 4 gives a definition of low-resource environments and Section 5 presents the datasets that we have used in our survey, including the introduction of iCUBE, a low-resource dataset in the area of Human-Robot Interaction. Section 6 explains our evaluation protocol to assess the performance of state-of-the-art ASR models in low-resource settings and presents their results in terms of WER and Character Error Rate (CER). Section 8 discusses our findings and their implications to the area of LRE ASR systems. Finally, Section 9 gives the conclusion of the paper and findings, along with future directions of research in the area.

2. Background in speech-to-text ASR models

In this paper, we focus on the impact of E2E deep learning techniques that are advancing state-of-the-art ASR models. ASR systems process input voice signals and generate corresponding transcriptions in a computer-readable format Scholz et al. (2006). The performance of an ASR system significantly influences the complexity of machine language understanding, which, in turn, affects the effectiveness of spoken dialogue systems (Celikyilmaz et al., 2018). As seamless communication between humans and machines is critical, this paper focuses on this aspect. Section 2.1 provides a brief history of ASR systems and their components, followed by a discussion of state-of-the-art, deep learning-based ASR techniques in Section 3. Finally, Section 4 presents an overview of ASR applications in LREs, highlighting the challenges that motivated this study.

2.1. History and components of ASR systems

An ASR system transforms an acoustic input sequence $X = \{x_1, \ldots, x_T\}$ of length T into a corresponding label sequence $L = \{l_1, \ldots, l_N\}$ of length N. The objective is to determine the most likely label sequence $\hat{L}$ for the given speech input X, defined as:

$$\hat{L} = \arg\max_{L \in \mathcal{V}^*} P(L|X) \tag{1}$$

where $\mathcal{V}^*$ represents the set of all possible label sequences (Wang et al., 2019a). In accordance with Eq. (1), the goal of an ASR system is to construct a model that accurately estimates the posterior distribution $P(L|X)$.

The earliest ASR system, developed by Bell Labs in the 1950s, utilized a filter bank to map speech input to hand-crafted templates, enabling the recognition of ten digits (Davis et al., 1952). Other early examples of filter-based approaches include voice-activated typewriters (Olson and Belar, 1956) and speaker-independent systems designed for recognizing ten vowels (Forgie and Forgie, 1959). These primary systems, however, were limited to detecting single words.

Subsequent research focused on expanding speech recognition to Large Vocabulary Continuous Speech Recognition (LVCSR) systems (Vintsyuk, 1968; Atal and Hanauer, 1971). Unlike simpler systems, handling context in speech data and managing a large vocabulary corpus posed significant challenges for LVCSR models (Wang et al., 2019a). As a result, *hybrid* ASR systems were developed, with Hidden Markov Model (HMM)-based approaches showing notable success (Bahl et al., 1983). These systems decomposed the ASR task into sub-problems – such as language and acoustic modeling – allowing separate models to handle each aspect. However, more recently, deep learning-based approaches have begun to replace hybrid systems, offering *end-to-end* ASR solutions where a single neural network maps input audio directly to text.

2.1.1. Classification of ASR systems

A large and growing body of literature has investigated ASR systems. Based on foundational principles and key innovations, previous research can be broadly classified into two categories: hybrid ASR systems and end-to-end ASR systems.

Hybrid ASR systems. A hybrid ASR approach converts input audio to its corresponding text representation by utilizing multiple models to address various sub-tasks within the overall system (Roger et al., 2022). This approach was widely used in ASR for several decades (Baker, 1975b; Bahl et al., 1983; Rabiner, 1989). A typical configuration of hybrid systems includes a combination of three independent modules: the acoustic model, the lexical model, and the language model, each playing a distinct role in the process.

A general hybrid ASR system consists of three main components: acoustic (acoustic–phonetic) modeling, lexical (pronunciation, lexicon/vocabulary) modeling, and language modeling. In this setup, the acoustic model learns to classify phonemes or equivalent speech units, the language model generates the desired sequence of words, and the lexical model captures the probabilistic relationships between latent variables and lexical units (such as the sequence of phones within a word).

An acoustic model (AM) is built by analyzing audio recordings of speech to generate a statistical representation of the sounds that constitute each word. The primary goal of the acoustic model is to map a sequence of acoustic features to a sequence of phonetic units. Hidden Markov Models (HMMs) have long been popular for acoustic modeling (Levinson et al., 1983). Both probabilistic and deterministic models can be utilized in this structure. A probabilistic model, such as a Gaussian Mixture Model (GMM), combines trained parameters with randomness. HMMs can be paired with GMMs, forming GMM-HMM systems (Stuttle, 2003), which enhance the accuracy of hybrid ASR systems.

On the other hand, a deterministic model, such as DNNs, generates an output sequence that directly corresponds to the input sequence. GMMs and DNNs can be integrated to compute the hidden states of HMMs, producing a final output sequence (Miao et al., 2015). Recent advances in deep learning have reignited interest in ASR systems, offering more accurate transcription (Li et al., 2015). As a result, HMMs have also been combined with DNNs for acoustic modeling in modern hybrid ASR systems.

The language model (LM) is employed to impose constraints on the recognition process, capturing the structure and semantics of the target language. Language modeling helps convert a sequence of phonetic units into meaningful words and sentences by considering the language's syntax, structure, and semantics during recognition. In hybrid ASR systems, LMs play a crucial role in improving the accuracy of output by guiding the system towards linguistically valid results. One of the most well-known models for language modeling in ASR systems is the Recurrent Neural Network Language Model (RNN-LM) (Kombrink et al., 2011; Goodman, 2001).

The lexical model in hybrid ASR systems is responsible for mapping acoustic features of speech to corresponding words in the vocabulary. It is typically based on a pronunciation dictionary that maps phonemes or sub-word units to their corresponding word forms. This dictionary may also include information about stress and intonation patterns, which are important for accurately transcribing spoken language. In some hybrid ASR systems, the lexical model may also include mechanisms for handling out-of-vocabulary (OOV) words—words not present in the vocabulary. This can be achieved by mapping OOV words to a set of similar vocabulary words or by using a separate module to generate new pronunciations based on the word's spelling or phonetic structure.

A speech decoder is a key component of a hybrid ASR system, responsible for converting audio input into a sequence of words. The acoustic signal is first transformed into a vector of speech features, which reduces the dimensionality of the data (Besacier et al., 2014). The ASR decoder generates a set of recognition hypotheses based on the input data. By applying language models, the decoder can then present the best recognition hypothesis. Language models play a crucial role in helping the decoder distinguish between different interpretations of the same acoustic input, thereby improving the overall accuracy of the hybrid ASR system (Amodei et al., 2016a).

A hybrid ASR system can be simply formalized as follows:

$$\hat{y} = f(g(X)) \tag{2}$$

in which g, f and X are the acoustic model, the language model and speech data, respectively.

Hybrid ASR systems have certain limitations (Zhang et al., 2016b). Since different models within the system require distinct training methods and in-domain data, the training process becomes increasingly complex when aiming for global optimization. Additionally, these systems often assume conditional independence between the tasks handled by each model to simplify training. However, this assumption does not reflect the reality of speech, where all components are highly interconnected.

End-to-end ASR systems. End-to-end models are supervised learning methods where the input audio features are directly mapped to an output sequence (Roger et al., 2022). In this context, end-to-end models can be seen as a streamlined alternative to hybrid ASR architectures, using a single deep neural network to convert audio to text without relying on multiple separate models. As a result, these models eliminate the need to design numerous modules with distinct optimization functions (Hori et al., 2017).

An end-to-end model includes encoder and decoder where the encoder transforms the input sequence into a feature sequence, while the decoder generates the final text representation (Hori et al., 2017). This architecture allows the model to learn both the acoustic model and language model jointly within a single network, unlike hybrid ASR systems, which separate these components.

A considerable amount of literature has been published on end-to-end ASR systems (Graves and Jaitly, 2014; Hannun et al., 2014a; Maas et al., 2014; Chorowski et al., 2014a). End-to-End models can be trained from scratch and can operate on words, sub-words or characters. They can be formally defined as:

$$\hat{y} = f(X), \tag{3}$$

where X is the speech data.

End-to-end speech recognition models utilize a single loss function for parameter estimation, allowing the model to directly optimize for the final result, which significantly improves the accuracy of the ASR system. Unlike the hybrid ASR approach, there is no need for additional processing to achieve accurate transcription in end-to-end models.

As end-to-end models replace the engineering process required to construct hybrid ASR systems with a learning-based approach, there is less reliance on domain-specific knowledge and experience for building the ASR model (Wang et al., 2019a). However, because end-to-end ASR systems rely on deep neural networks to map input sequences directly to output sequences, they require large amounts of training data to achieve high accuracy, which may not be feasible in all real-world applications.

3. A survey of the state-of-the-art in end-to-end ASR systems

The vast majority of state-of-the-art research in the speech recognition area has focused on large training datasets that use up to hundreds or thousands of hours of audio data in their models. In this section, we will provide an overview and will discuss the extensive literature in the area according to the base architecture used in the model. This section meticulously explores various methodologies, starting with GMM-HMM models in Section 3.1, followed by neural networks which integrate with HMM in Section 3.2. The discourse extends to end-to-end architectures in Section 3.3, transfer learning techniques in Section 3.4, self-supervised learning approaches in Section 3.5, and delves into the specifics of CNN, RNN, and Transducer strategies in Sections 3.6 and 3.7, respectively. Additionally, it highlights the latest advancements in Attention-Based and Transformer-based models in Section 3.8 and Section 3.9, showcasing the evolution and diversification of approaches in the field.

In the 1950s and 1960s, ASR research mainly concentrated on speaker dependent recognition of isolated words. This was driven by R&D labs such as Bell Labs and NEC Corporation, as well as academic labs like MIT Lincoln Labs (Dudley and Balashek, 1958). Phoneme-level transcription was also attempted, but with less success. The common scenario was a single adult reading single digits through a microphone. One of the earliest experiments achieved up to 99% accuracy on isolated digit recognition (Davis et al., 1952). This experiment used formant frequency estimates to identify whole words. These systems did not have any model of sub-word units such as syllables, consonants, or vowels. The word was the only unit, and all words were compared to each other during classification to find the best match. The template-matching technique was not feasible for a few reasons. For example, it relied on vowels that needed to match entire words, which required a storage of each word on disk. This technique would encounter major challenges in time and space complexity if expanded to larger vocabularies.

In 1971, speech research received a major boost when the US Department of Defense's Advanced Research Program Agency (ARPA) initiated the 5-year Spoken Understanding Research (SUR) program. The aim of the program was to achieve a breakthrough in speech understanding capability that could then be applied to the development of practical human–machine communication systems (Klatt, 1977). ARPA envisioned a system that Airforce pilots could operate with their voice while their hands were occupied with steering. The challenge was to create a system that could recognize simple sentences from a vocabulary of 1, 000 words with a 10% WER in reasonable time. To build a recognizer that could handle sentences instead of isolated words, where the system did not know the length of the string, major revisions of the Isolated Word system were required. The phoneme is the smallest speech sound that has meaning. Every language has a limited number of phonemes, and all words are formed with this limited number of phonemes. Usually languages have no more than 50 phonemes, and this number does not increase with vocabulary or grammar complexity. Simple systems had hard limits of 100 or 1000 words, but with only 50 distinct phonemes there is no upper limit to the number of words a system based on phonemes can recognize. All the teams in the ARPA project used the phoneme as the unit for speech modeling. At the end of the project, the team at Carnegie Mellon had the best performance with Harpy (Lowerre, 1976). Like in Isolated Word Recognition, all teams used some kind of template matching, but with templates of phonemes instead of templates of words. Harpy is a system that is specific to each speaker, and the 98 phoneme templates have to be adjusted to each speaker. For a new speaker to be added to the system, she has to record example sentences for about 30 min, which are then aligned to the graph by force. This forced-alignment, however, requires that at least one speaker has been enrolled before, and their 98 phoneme templates are used to align the next speaker's audio. Since the intended application was command and control, a restricted vocabulary and grammar was acceptable. Harpy, which decoded in about 80x real time, was not practical to use and difficult to speed up. Besides the issue of speed, grammar flexibility was also a crucial factor for developing systems that could recognize natural speech. Harpy could only recognize sentences that followed a Backus–Naur form (BNF) grammar. This was composed of a set of manually designed rules, and was not very adaptable. A major change was imminent in the speech recognition field, shifting from template matching and rigid grammars to statistical Acoustic Models and statistical grammars. Rather than fixed assignments, a better system would assign a probability to the sentence, word, or sound in question.

3.1. The GMM-HMM methods

In the 1970s, Hidden Markov Models (HMMs) (Baker, 1975a) were proposed, and in the 1980s, Gaussian Mixture Models (GMMs) (Juang, 1985) and statistical grammars (Katz, 1987) were developed, but it was only in the 1990s that all three components were integrated into one Open Source toolkit: the Hidden Markov Model Toolkit (Young et al., 2002). The Hidden Markov Model Toolkit (HTK) holds the distinction of being the first toolkit to integrate all essential elements of the modern Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) approach to speech recognition. While Carnegie Mellon University's (CMU) Sphinx toolkit was a frontrunner in the ASR toolkit landscape (Lee et al., 1990), its initial version differed from HTK in its methodology. CMU's Sphinx originally employed Vector Quantized codebooks for estimating emission probabilities in HMM states, whereas HTK adopted GMMs as early as 1994. Although HTK and Sphinx had slight variations in performance, HTK gained widespread acclaim and preference among speech recognition researchers, largely due to its comprehensive documentation, known as 'The HTK Book'.

This book became an essential resource and a go-to reference in the field. HTK's capabilities were first demonstrated as a part of a benchmark test on DARPA's Resource Management task in 1992 (Woodland and Young, 1993), showcasing its effectiveness and setting a precedent for future developments in speech recognition research. This benchmark not only illustrated HTK's proficiency but also marked a significant step forward in the evolution of speech recognition technology, laying the groundwork for subsequent advancements in the field. The process of transforming a raw audio signal into a format more suitable for speech modeling is crucial, as the raw signal itself is not immediately amenable to effective speech analysis. Feature extraction does more than just refine the audio by enhancing the signal-to-noise ratio; it fundamentally transforms the nature of the data. Raw audio represents variations in air pressure over time, encapsulating all elements essential to human speech, as speech is essentially a modulation of air pressure. However, distinguishing speech sounds involves analyzing these variations in both frequency and time domains. Feature extraction serves to explicitly present this frequency information, which is crucial for differentiating speech sounds. It circumvents the need for a statistical model to independently deduce this information. In the domain of ASR, and particularly within the framework of the Hidden Markov Model Toolkit (HTK), the two predominant techniques for audio feature extraction are Mel-frequency cepstral coefficients (MFCCs) and perceptual linear prediction (PLPs).

A monophone GMM-HMM Acoustic Model, as implied by its name, comprises a singular model representing each phoneme. In a language that encompasses 50 phonemes, this system would feature 50 distinct GMM-HMMs, each corresponding to a different phoneme. The specific number of states assigned to the HMM of each phoneme is not predetermined and can vary based on the discretion of the researcher. Considering the sequential nature of human speech, which progresses in a left-to-right manner, these monophone HMMs are configured to follow a strictly left-to-right pattern, incorporating self-loops as well.

The acoustic properties at the boundaries of phonemes are markedly different from those at the center. Typically, the central state of a monophone is expected to exhibit more consistency compared to its edges, given the variable nature of speech sounds at these boundary points. Moreover, the use of a three-state model in monophones not only aids in capturing the nuances of co-articulation (where phonemes influence each other) but also enables the effective representation of complex sounds. Such sounds include affricates or stop consonants, which are characterized by multiple acoustic events. This three-state structure is therefore crucial in providing a more detailed and accurate modeling of the diverse range of sounds present in human speech, enhancing the overall effectiveness and precision of the acoustic model in capturing the intricacies of spoken language.

Understanding the acoustic properties of human speech need to collection of extensive speech data. Yet, merely possessing a large corpus of spoken language is not sufficient. It is crucial to accurately identify how specific segments of the audio align with the states in our GMM-HMMs. This alignment is essential for appropriately updating the parameters of the models with relevant data. While manual alignment

of audio segments to HMM states is feasible, it is an exceedingly labor-intensive process. This approach has been employed in creating corpora such as TIMIT (Garofolo et al., 1993) and the Ohio Buckeye corpus (Pitt et al., 2005), where meticulous effort was put into ensuring precise alignment. Such detailed and careful alignment is fundamental for constructing a robust and reliable acoustic model, as it ensures that the training data accurately reflects the complex variability inherent in human speech.

For the successful training of monophone GMM-HMMs using speech datasets, it is crucial to segment the entire corpus to align with the HMM states, setting the stage for the Baum–Welch algorithm. This necessitates an initial guess of the model parameters, a step known as initial alignment, where flat-start training offers a simple solution. In the Baum–Welch re-estimation, these initial parameters are crucial for generating new alignments within the model. Instead of opting for a random parameter initialization, which could lead to long time training due to its significant deviation from actual data, a more practical and efficient approach is to derive this initial guess directly from the audio data. This method ensures a closer approximation to the real data right from the beginning, facilitating a faster and more accurate convergence of the model to the optimal parameters, thus streamlining the training process of the GMM-HMMs in speech recognition applications. After obtaining an initial approximation of all the parameters in our monophone GMM-HMMs, we can use both the data and the model to improve the alignments. The main concept of Baum–Welch is that we can use the existing model parameters of any utterance HMM to produce the most probable alignment of that HMM to its matching audio clip. Then we use that alignment as the true alignment, and update the model parameters based on it.

3.2. DNN-HMM methods

The contemporary methodology in HMM based speech recognition has evolved to incorporate DNNs, leading to the development of DNN-HMMs. Often described as the 'hybrid' approach, this technique represents a fusion of traditional HMMs, as used in GMM-HMM speech recognition systems, with neural network-based acoustic modeling. The hybrid approach began to gain significant momentum in the 2000s, a period marked by a resurgence of interest in neural networks. This integration of DNNs into HMM-based frameworks has not only enhanced the accuracy and robustness of speech recognition models but has also led to a paradigm shift in the field, driving forward the boundaries of what is achievable in acoustic modeling and speech processing. This hybrid approach exemplifies the convergence of traditional statistical models with modern neural network architectures, showcasing a significant milestone in the evolution of speech recognition technologies.

One of the most widely used ASR toolkits for DNN-HMM research is Kaldi, which has several advantages over other toolkits. First, Kaldi was among the first toolkits to support DNN acoustic modeling, which has become the dominant paradigm for ASR systems. Second, Kaldi adopts a novel approach to the graph decoder, based on Weighted Finite State Transducer technology (Mohri, 1997), which allows for efficient and flexible integration of various linguistic and acoustic components. Third, Kaldi is written in C++, a programming language that offers high performance and portability which make Kaldi attractive to both academic and commercial developers of ASR systems.

DNN-HMM and GMM-HMM methods are almost identical in most aspects, and the DNN can be easily substituted for the GMM acoustic model. However, some modifications are necessary for the Viterbi decoding math to be correct. The standard decoding calculations need the Acoustic Model to produce a likelihood of the phoneme given the audio, and while GMMs provide likelihoods of the phoneme given the audio, neural nets provide a posterior probability for each audio given the phoneme. Apart from some minor math modifications, GMMs and DNNs have the same function at decoding time. Both Acoustic Models provide information about audio-phoneme relations. The rest of the decoding graph (phonetic dictionary, n-gram language model) is unchanged. The training of GMM-HMM models uses the Baum–Welch algorithm to estimate the parameters via flat-start monophone training, which iterates the steps of alignment and estimation to produce increasingly precise alignments (and more accurate GMMs). This procedure is mathematically valid, ensuring that the parameter estimates improve with each iteration for the given data.

In speech recognition research area, both the Deep Neural Network-Hidden Markov Model (DNN-HMM) and the GMM-HMM methodologies are very similar to each other, with the DNN essentially functioning as an interchangeable replacement for the GMM in acoustic modeling. However, to effectively integrate the DNN with the Viterbi decoding algorithm, certain modifications are required. Traditional decoding processes need that the Acoustic Model computes the likelihood of a given audio segment corresponding to a specific phoneme. While GMMs directly provide the likelihood of the audio for a phoneme, neural networks output the posterior probability of each phoneme given the audio. Both types of Acoustic Models propose crucial information on the relationship between phonemes and audio. Other components of the decoding process, such as the phonetic dictionary and the n-gram language model, remain unaffected and continue to function as before.

Incorporating temporal information into the training of DNN Acoustic Models involves a technique beyond simply inputting features frame by frame. This method, known as frame-splicing, entails adding context to a central frame from adjacent frames on both the left and right sides. Despite this addition of contextual data, the network continues to make predictions on a frame-by-frame basis. However, the key difference lies in its ability to utilize the surrounding context to make a more informed prediction about the phoneme associated with the central frame. This approach enhances the model's understanding of the temporal dynamics in speech, allowing for a more nuanced and accurate identification of phonemes.

3.3. End-to-end methods

In 2014, the field of ASR systems experienced a paradigm shift with the advent of the first End-to-End (E2E) speech recognition systems, capable of directly translating audio into text (Graves and Jaitly, 2014; Hannun et al., 2014a; Maas et al., 2014; Graves et al., 2013). This marked a significant departure from conventional ASR methodologies, which traditionally involved the separate training of distinct models for acoustic processing, lexical sequencing, and phonetic transcription. This was a time-consuming and challenging task for an engineer, and a hindrance to deploying and debugging speech recognition in production. Moreover, the traditional, Hybrid approach was not preferred a priori because each model needed its own objective function for parameter estimation, and thus joint estimation of all models was impossible. Furthermore, this traditional methodology was inherently limited in its lexical scope; words not encompassed within the language model's vocabulary were deemed Out-Of-Vocabulary (OOV) and thus, undetectable in speech recognition. The emergence of E2E systems signified a critical advancement, proposing a more integrated and potentially more effective framework for ASR, addressing some of the inherent limitations of the previous models.

End-to-end speech recognition models are preferred due to their unified approach towards parameter estimation, characterized by a singular loss function. This unified loss function directly targets the core objective of ASR: deducing the most accurate transcription for a given audio segment. However, by discarding the HMMs from the Hybrid approach, End-to-End models lack the alignment information that is usually used in training conventional models, thereby losing access to the alignment information that is integral to the training of traditional models. Consequently, end-to-end models face the complex challenge of independently establishing an alignment between the textual and audio

data during the training phase. This task of alignment, integral to effective model training, presents a significant hurdle in the development and optimization of end-to-end speech recognition systems.

The Connectionist Temporal Classification (CTC) objective function was used by the first methods for End-to-End speech recognition (Amodei et al., 2016a; Graves and Jaitly, 2014; Graves et al., 2013). This was closely followed by the development of the Sequence-to-Sequence model incorporating an Attention mechanism (Chorowski et al., 2014b; Bahdanau et al., 2016). An important advancement within this framework was the introduction of the Listen, Attend, Spell (LAS) model, an extension that further refined the Sequence-to-Sequence model with Attention (Chan et al., 2016). Deviating from the recurrent model paradigm, the Wav2Letter model emerged, notable for its exclusive reliance on convolutional layers and a modified version of the CTC Loss function (Collobert et al., 2016). Additionally, the RNN Transducer model was developed, offering capabilities for online, streaming decoding similar to CTC models, but with the added advantage of circumventing the conditional independence assumption inherent in CTC models (Battenberg et al., 2017; Rao et al., 2017).

3.4. Transfer learning based models

Adapting models initially trained for a specific domain or language to another can be effectively achieved through the method of transfer learning (Wang and Zheng, 2015). This process involves transferring the parameters, which are essentially the weights of a neural network, from a pre-trained deep neural network-based model to new domains or languages (Ghahremani et al., 2017). These weights, estimated and computed during the initial training of the model, carry significant learned information. In the field of Natural Language Processing (NLP), transfer learning is particularly useful for transferring knowledge across models that have been trained on data from different but related languages. This technique leverages the underlying similarities and shared characteristics.

One of the first works in applying transfer learning to DNN-based ASR was presented in Huang et al. (2013), where a multilingual DNN was constructed using four European languages. This DNN was uniquely structured such that all hidden layers were common across the languages, with the exception of the final Softmax layer, which was language-specific. After training on these four languages, the study explored the application of transfer learning for two additional languages that were not included in the initial training phase. These languages, American English and Mandarin Chinese, were selected for their contrasting phonetic characteristics relative to the European languages used in the network's training. American English shares phonetic similarities with the European languages, whereas Mandarin Chinese exhibits significant phonetic divergence. To facilitate this, a dedicated Softmax layer for each of the new target languages was integrated into the network, enabling an assessment of transfer learning's efficacy across linguistically and phonetically diverse languages. The authors demonstrated that cross-lingual hidden layer transfer can enhance the ASR performance for two novel languages. They further observed that transfer learning has a substantial impact when the target language has limited training data. In this scenario, they recommended only fine-tuning the Softmax layer rather than updating more layers.

The authors of Kunze et al. (2017) implemented a fully convolutional, end-to-end ASR model, initially training it with English data before applying transfer learning to adapt it for German. The cross-lingual transfer outperformed the ASR model that was trained from scratch on German data only. Another study, (Cho et al., 2018), developed a multilingual ASR model utilizing 10 languages from the BABEL speech corpus. This model was then adapted through transfer learning for an additional four BABEL languages. Their findings suggested that using transfer learning from a multilingual ASR model is more advantageous than using monolingual ASR models. A more recent study (Yi et al., 2018) suggested a language-adversarial transfer learning method. This method ensured that the shared layers across multiple languages had less redundant language-dependent information. By using adversarial learning, the shared layers could acquire language invariant features. They showed promising results on IARPA Babel dataset.

The authors in Durrett et al. (2012) utilized transfer learning in the context of dependency parsing by employing bilingual lexicons from two distinct languages, designated as the source and target. The methodology involved conducting syntactic analyses of parallel sentences from languages with varying levels of resources — one being resource-rich and the other resource-poor. The objective was to facilitate the transfer of acquired syntactic knowledge between words that convey similar meanings or concepts in both languages. This approach leverages the syntactic structures from the resource-rich language to enhance the parsing capabilities in the resource-poor language, demonstrating the potential of transfer learning in bridging linguistic resource gaps.

Speech recognition applications have also benefited from transfer learning, which enables acoustic models trained for high-resourced domains (or languages) to be adapted to low-resourced domains (or languages). The main benefits of the adaptation are to overcome resource limitations and to save the effort of collecting a large amount of data, which is always a hurdle in speech recognition research. In addition, some studies (Ghahremani et al., 2017; Yan et al., 2018; Feng and Lee, 2018) have shown that the transferred acoustic models achieve higher performance when the source models have good quality and are relevant to the target languages/domains. Transfer learning differs from other adaptation methods such as multilingual training, which needs a common phone set across languages. Multilingual training conducts joint training, which involves combining data from source and target languages, and creating a shared acoustic model where each language has its own final (softmax) layer. Transfer learning, however, does not have to match phone sets. This practical advantage makes it more suitable for low-resourced languages, especially those whose phone set is very distinctive and difficult to share with others.

3.5. Self-supervised based models

To address the need for labeled data, researchers have investigated methods that utilize unpaired audio-only data, thereby enabling new industrial speech applications and supporting low-resource languages (Ma et al., 2006). Drawing inspiration from how children acquire their first language through listening and interaction with their environment, scientists aim to use raw waveforms and spectral signals to develop speech representations. These representations encompass low-level acoustic events, lexical knowledge, and extend to syntactic and semantic information. Subsequently, these learned representations are applied to downstream tasks, which require a minimal amount of labeled data (Bengio et al., 2013). Representation learning, in this context, involves algorithms that extract latent features to uncover the underlying explanatory factors of the observed input (Bengio et al., 2013).

Representation learning approaches are typically regarded as forms of unsupervised learning. This category of machine learning methods identifies naturally occurring patterns in training samples that lack preassigned labels or scores (Jordan and Mitchell, 2015). The term unsupervised differentiates these methods from supervised approaches, where each training sample is labeled, and semi-supervised approaches, which use a small set of labeled samples to guide the learning process for a larger set of unlabeled samples. A rapidly expanding subset of unsupervised learning is self-supervised learning (SSL), which uses information extracted from the input data itself as labels to learn representations beneficial for downstream tasks. For instance, traditional unsupervised methods like k-means clustering do not fall under self-supervision as they minimize within-cluster variance during learning. This review focuses on self-supervised learning approaches. The SSL approaches categories into generative approaches, contrastive approaches and predictive approaches.

3.5.1. Generative SSL approaches

In generative approaches, the pretext task aims to create or reconstruct input data using a restricted perspective. This involves predicting future inputs from past ones, distinguishing masked from unmasked inputs, or recovering the original input from a corrupted view. In this paper, the term generative term specifically refers to models that focus on the original input during their pretext task.

Autoencoders (AEs) (Hinton and Zemel, 1993) represent a generative approach that has played a crucial role in acquiring distributed latent representations from sensory data. AEs consist of an encoder and decoder, with the pretext task being input reconstruction. The most prevalent type of AE introduces an information bottleneck in the latent representation by using fewer hidden units than input features. This design choice compels the model to omit low-level details and discourages the learning of trivial solutions. The Variational Autoencoder (VAE) (Rezende et al., 2014) represents a probabilistic variant of the AE, defining the latent representation through a posterior distribution over stochastic latent variables. VAEs have found applications in speech-related research –. Another model within this category is the vector-quantized variational autoencoder (VQ-VAE) (Van Den Oord et al., 2017), which enhances the original VAE by introducing a novel parameterization for the posterior distribution of discrete latent representations. Autoregressive predictive coding (APC) draws inspiration from the classic Linear Predictive Coding (LPC) method used for speech feature extraction (O'Shaughnessy, 1988), as well as from autoregressive language models applied to text. In Chung and Glass (2020), the APC objective is expanded to multi-target training, generating both past and future frames based on prior context. VQ-APC (Chung et al., 2020) combines quantization with the APC objective, introducing an information bottleneck as a regularization mechanism.

Masked reconstruction draws inspiration from BERT's masked language model (MLM) task (Devlin et al., 2018). During BERT pretraining, certain tokens in input sentences are randomly masked by replacing them with a learned masking token or another input token. The model then learns to reconstruct the masked tokens using information from the non-masked tokens. In the context of non-autoregressive predictive coding (NPC) (Liu et al., 2021a), time masking is introduced through masked convolution blocks. Additionally, taking cues from XLNet (Yang et al., 2019), some researchers have proposed reconstructing the input from a shuffled version (Song et al., 2020) to address discrepancies between pre-training and fine-tuning in masking-based approaches.

3.5.2. Contrastive SSL approaches

Contrastive models acquire representations by discerning a target sample (positive) from distractor samples (negatives) based on an anchor representation. The pretext task involves maximizing the latent space similarity between the anchor and positive samples while minimizing the similarity between the anchor and negative samples. The Contrastive Predictive Coding (CPC) model (Oord et al., 2018) learns representations by maximizing mutual information between the current context and future embeddings while minimizing the noise-contrastive estimation-based (NCE) loss (Gutmann and Hyvärinen, 2010). Wav2Vec2.0 (Baevski et al., 2020), a contrastive learning-based approach, comprises a CNN-based encoder network, a VQ module, and a Transformer-based context representation network. The model also incorporates the masking concept, randomly masking input tokens before feeding them to the Transformer. During pretraining, Wav2Vec2.0 is optimized using a contrastive loss function. Additionally, a regularization term is introduced to enhance the diversity of the codebook within the VQ module. While representations learned through contrastive approaches have demonstrated effectiveness in various downstream applications, they encounter challenges when applied to speech data. One such challenge arises from the strategy used to define positive and negative samples, which can inadvertently impose invariances on the learned representations. Additionally, due to the absence of explicit segmentation of acoustic units in speech input, the negative and positive samples do not necessarily correspond to complete language units; instead, they may cover partial or multiple units, depending on the span of each sample. Wav2vec-C (Sadhu et al., 2021) presents a novel self-supervised learning approach combining wav2vec 2.0 and VQ-VAE techniques. The model uses a contrastive loss to predict masked speech encodings and introduces a consistency network to reconstruct input features, improving codebook utilization. Trained on 10k hours of unlabeled data, Wav2vec-C achieves significant error reduction in ASR tasks compared to wav2vec 2.0, particularly in noisy environments. This demonstrates its effectiveness for realistic, low-latency speech recognition applications. Speech SimCLR (Jiang et al., 2021) introduces a new self-supervised learning framework for speech representation. This method, termed Speech SimCLR, combines contrastive loss, which maximizes agreement between differently augmented samples, with a reconstruction loss to enhance input representation. The proposed approach applies augmentations to raw speech and its spectrogram during training. Unspeech (Milde and Biemann, 2018) focuses on improving ASR systems through the use of audio augmentation techniques. The researchers explore various data augmentation strategies to enhance the performance of ASR systems, particularly for low-resource languages and noisy environments. The study includes techniques such as speed perturbation, volume control, and adding background noise to create more robust training datasets. These methods aim to simulate real-world variability in audio data, thus improving the ASR systems' ability to generalize and perform accurately in diverse conditions.

3.5.3. Predictive SSL approaches

Predictive approaches, similar to the contrastive methods discussed earlier, rely on a learned target for the pretext task. However, unlike contrastive approaches, they do not utilize a contrastive loss; instead, they employ loss functions such as squared error or cross-entropy. While a contrastive loss prevents the model from learning trivial solutions by using negative samples, predictive methods take a different approach. They compute targets outside the model's computational graph, often with a separate model.

Discrete BERT (Baevski and Mohamed, 2020) evaluates various self-supervised representation learning algorithms for ASR systems. It compares methods that explicitly quantize audio data, like vq-wav2vec, with those that do not, finding that quantization builds a more effective vocabulary for subsequent BERT training. Unlike previous work, this study fine-tunes pre-trained BERT models directly on transcribed speech using a Connectionist Temporal Classification (CTC) loss. The results show that using vq-wav2vec followed by BERT training significantly reduces the WER on ASR tasks, especially in low-resource settings. WavLM (Chen et al., 2022) is a pre-trained model designed to handle a wide range of speech processing tasks beyond just speech recognition.

WavLM improves upon previous models by combining masked speech prediction and denoising, enhancing both ASR and non-ASR tasks like speaker diarization, speaker verification, and speech separation. The model employs a gated relative position bias within the Transformer structure to better capture the sequence ordering in speech. It is trained on a large and diverse dataset of 94,000 h, including Libri-Light, GigaSpeech, and VoxPopuli, to improve robustness across different acoustic environments. WavLM is built upon a Transformer architecture with a convolutional feature encoder followed by a Transformer encoder, incorporating a gated relative position bias to better capture sequence order in speech. The convolutional encoder includes seven blocks of temporal convolution layers, while the Transformer consists of multiple encoder layers with relative position embeddings. WavLM employs a masked speech prediction and denoising framework, allowing it to learn robust representations from 94,000 h of diverse audio data, including Libri-Light, GigaSpeech, and VoxPopuli. There are three model variants: WavLM Base with

12 layers, WavLM Base+ with the same architecture but trained on more data, and WavLM Large with 24 layers. The model demonstrates state-of-the-art performance on the SUPERB benchmark and excels in tasks such as speaker verification, speech separation, and speaker diarization, showcasing its versatility and effectiveness for full-stack speech processing.

data2vec (Baevski et al., 2022) a self-supervised learning framework applicable across speech, vision, and language domains. Data2Vec leverages masked prediction tasks to learn representations that predict latent representations of input data rather than the data itself, thereby improving performance and robustness across various modalities. The model demonstrates superior performance on benchmarks for speech recognition, image classification, and natural language understanding, showcasing its versatility and effectiveness in learning generalized representations. BEST-RQ (Chiu et al., 2022) is a self-supervised learning algorithm for speech recognition that simplifies the training process by using a random-projection quantizer. This quantizer projects speech inputs using a randomly initialized matrix and performs nearest-neighbor lookup in a randomly-initialized codebook, which remains fixed during training. This separation from the speech recognition model allows flexibility and compatibility with various architectures. The model masks speech signals and predicts masked regions based on unmasked parts. BEST-RQ achieves competitive WER on the LibriSpeech benchmark, outperforming previous models like wav2vec 2.0 and w2v-BERT, particularly in streaming scenarios and multilingual tasks. It employs a random-projection quantizer that projects speech signals through a randomly initialized matrix and maps them to discrete labels using a nearest-neighbor search in a randomly-initialized codebook. This quantizer is independent of the model and remains fixed during training, allowing the algorithm to be flexible and compatible with various speech recognition architectures. The model architecture features a Conformer encoder, which is used to predict the masked regions of the speech signal based on the unmasked parts. For pre-training, a softmax layer is added on top of the Conformer encoder to predict the quantized speech labels. The pre-training uses a masking strategy where parts of the input speech are masked and replaced with noise, and the encoder learns to predict the labels of these masked segments. This design supports both non-streaming and streaming models, making it versatile for low-latency applications.

Adaptation in SSL (Chen et al., 2024) plays a crucial role in scenarios where there is a distribution shift between the training data (source domain) and the data on which the model will be deployed (target domain). In many practical applications, especially in fields like medical imaging, collecting labeled data for every possible condition or variation is impractical due to resource constraints. SSL aims to leverage abundant unlabeled data alongside limited labeled data to improve model performance. However, when the unlabeled data comes from a different distribution than the labeled data, adaptation mechanisms are necessary to bridge the gap between the domains.

One effective approach to address this challenge is domain adaptation within SSL frameworks. This involves designing models that can learn domain-invariant features, allowing them to generalize well across different data distributions. Techniques such as adversarial training can be employed, where a discriminator is trained to distinguish between source and target domain features while the feature extractor tries to fool the discriminator (Chen et al., 2024). This results in the learned features being indistinguishable across domains, thus reducing the domain shift and improving the model's performance on the target domain.

Another strategy is to use self-supervised learning tasks that are applicable to both domains. By training the model on auxiliary tasks that do not require labels – such as predicting rotations, solving jigsaw puzzles, or reconstructing input data – the model can learn rich representations from the unlabeled target domain data. These representations capture essential structures and patterns inherent in the target domain, which can then be fine-tuned with the limited labeled data available. This approach not only enhances the feature extraction process but also aids in aligning the source and target domains at a representational level (Chen et al., 2024).

3.6. Methods based on Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN)

Due to the impressive results of DNN models, almost all state-of-the-art ASR systems use a modification of DNNs in their structure (Dahl et al., 2011; Hinton et al., 2012; Jaitly et al., 2012). As RNN architectures, especially LSTMs, are a good option for sequence processing, they have been used in state-of-the-art STT systems (Sak et al., 2014a). CNN-based models are considered as another solution for sequence learning (Liptchinsky et al., 2017). The combination of CNN layers before RNN layers are used to help the model to provide more accurate feature extraction (Sainath et al., 2015b). In this section we summarize the CNN and RNN based a Table 1 illustrates an overview of these models.

3.6.1. RNN-LSTM-based models

RNN-LSTM-based architectures were used as language models in end-to-end models in the literature in Sundermeyer et al. (2012), Frinken et al. (2012). Research has shown the restricted advantages of minor architectural improvements in the original LSTM as a language model (Jozefowicz et al., 2015). The design of Vanilla-LSTM is based on the use of intuitive multiplicative gates. Highway connections (Zhang et al., 2016a) and residual connections (Zhao et al., 2016; Kim et al., 2017a) are the most prominent changes in the LSTM-based architectures, along with dropout (Srivastava et al., 2014).

A typical LSTM layer converts the input vector x_t to the output vector h_t through a gate-cell structure as follows (Li et al., 2018):

$$i_t = \sigma(W_{ix}X_t + W_{ih}h_{t-1} + P_i \odot c_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_{fx}X_t + W_{fh}h_{t-1} + P_f \odot c_{t-1} + b_f) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}X_t + W_{ch}h_{t-1} + b_c) \tag{6}$$

$$o_t = \sigma(W_{ox}X_t + W_{oh}h_{t-1} + P_o \odot c_t + b_o) \tag{7}$$

$$h_t = o_t \odot \phi(c_t) \tag{8}$$

where X_t is the speech spectrum input at the time step t. The activation of the input, output, forget and memory cells are i_t, o_t, f_t and c_t, respectively. The output of the LSTM cell is h_t. $W_{.x}$ and $W_{.h}$ which are the weight matrices for the input and recurrent inputs, respectively. P_i, P_o, P_f are vectors that are associated with peephole connections. Finally, b_i, b_f, b_c, and b_io are bias vectors.

A large and growing body of literature has investigated the LSTM-RNN model (Graves et al., 2013; Sak et al., 2014a; Miao and Metze, 2015; Sainath and Li, 2016) and has shown state-of-the-art results on ASR tasks. LSTM variations, such as multiple LSTM layers stacked, yield better results (Sak et al., 2014a). However, gradient vanishing errors and the need for a large amount of training data are major problems of such models (Hsu et al., 2016).

The vanishing gradient problem in LSTMs can be mitigated through the incorporation of skip connections or gating mechanisms between layers. Residual LSTMs (Zhao et al., 2016; Kim et al., 2017a) address this issue by introducing connections between layers, thereby reducing the impact of vanishing gradients. Another approach, Highway-LSTMs (Zhang et al., 2016a), connects memory cells across adjacent layers, establishing parallel pathways for data flow. Grid-LSTMs (Kalchbrenner et al., 2015) further extend this concept by organizing memory cells into a multidimensional grid within the LSTM structure, resulting in improved performance over Highway-LSTMs on several ASR tasks. Layer-Trajectory LSTMs (ltLSTM) (Li et al., 2018) introduce a stacked LSTM architecture that combines outputs from time-LSTMs with a summarized layer to incorporate trajectory information for final classification. This model decouples the time-recurrence mechanism from the classification process, allowing forward propagation in both the time-LSTM and layer-LSTM threads independently. As a result, the computational complexity remains comparable to that of standard time-LSTMs (Li et al., 2018).

Future context frames contain valuable information that enhances the accuracy of target label prediction. In Li et al. (2019b), the authors enhance the performance of Layer-Trajectory LSTM (ltLSTM) models by incorporating future context frames through a technique called the look-ahead embedding, which represents future variables as a fixed-size vector (Li et al., 2018). Additionally, bi-directional LSTM (LC-BLSTM) reduces the latency of traditional BLSTM models by employing chunk-wise forward LSTMs (Zhang et al., 2016a). Time-delay neural networks (TDNN) (Peddinti et al., 2015) and feed-forward sequential memory networks (FSMN) (Zhang et al., 2017) also leverage future acoustic frames by utilizing 1-D Convolutional Neural Networks with a sliding window of acoustic frames. These models effectively incorporate future context to improve prediction accuracy. Furthermore, in Li et al. (2020), the authors refine the contextual layer trajectory LSTM (cltLSTM) (Li et al., 2019b) by introducing a two-headed structure, where one head operates with zero latency and the other with minimal latency, further enhancing model performance.

In end-to-end models, deep BLSTM neural networks have achieved state-of-the-art results (Graves and Schmidhuber, 2005; Graves et al., 2005). Since BLSTM-based ASR systems require the entire speech utterance to compute output frames, they must account for both past and future speech context (Moritz et al., 2019b). Consequently, these models are not suitable for streaming ASR applications due to their dependency on extensive future context. One potential solution to this limitation involves using overlapping chunks of frames to compute the backward LSTM output. This approach is implemented in latency-controlled BLSTM (LC-BLSTM) models (Chen and Huo, 2016; Zeyer et al., 2016), though the overlapping of frames increases computational costs. Deep contextualized acoustic representations (DeCoAR) (Ling et al., 2020), on the other hand, leverage large amounts of unlabeled data through representation learning, reconstructing temporal slices of filterbank features from both past and future context frames. By incorporating contextual information – such as preceding and following words – DeCoAR improves the model's ability to understand the meaning and context of spoken language. DeCoAR has demonstrated promising results across a range of speech recognition tasks, including speech-to-text transcription, speaker recognition, and keyword spotting.

Feedforward Neural Networks (FNN) have also been combined with RNN to improve the accuracy of STT systems. In Chien and Misbullah (2016), the authors presented different networks in which FNNs were combined with LSTM to show temporal patterns and to summarize the long history of previous inputs. LSTM Recurrent Projection was proposed in Sak et al. (2014a), in which the authors added a feedforward layer by considering recurrent information based on the output of LSTM. Simultaneously, a LSTM structure was updated by adding FNN before and after LSTM in Li and Wu (2015). However, while clear improvements have been introduced to manage the gradient vanishing problem, no clear improvements or modifications have been done to LSTM-based models to tackle the limited training datasets.

3.6.2. CNNs for ASR

Convolutional Neural Network (CNN) is a specialized form of neural network that uses a supervised deep feature learning model to process data (Goodfelow et al., 2016). A specialized kind of linear operation called convolution is utilized by this type of deep neural network. One of the first research on CNN has been done in LeCun et al. (1989) which this network has been used to identify handwritten characters. Image and video recognition, recommendation systems, image classification, medical image analysis, and natural language processing are different applications of this network. Dahl et al. (2011). CNN architecture needs a large amount of data for training to be able to utilize it for applications with high-dimensional input data, such as image processing and speech recognition. Furthermore, by increasing the number of parameters to train in deeper structures, this network requires high performance computing power (Deng et al., 2009).

CNNs are efficient deep networks that exploit local properties, called *formants* in speech recognition (Cai and Liu, 2016). The frequency variation in speech signals is another application of this network in speech recognition systems (Fantaye et al., 2020). Recent developments in CNNs have led to renewed interest in their use in both high-resource (Sainath et al., 2015a) and low-resource (Chan and Lane, 2015) environments. CNN is used for acoustic modeling in Lee et al. (2009), Hau and Chen (2011). In these approaches, to achieve more stable acoustic features from the input audio, convolution layers are applied over the windows of the acoustic frames. Jasper (Li et al., 2019a) is a deep convolution model in which the convolutional layer $1D$ is stacked with skip connections. In, (Hannun et al., 2019) the authors used depth-wise separable convolution layers (Chollet, 2017) to improve the speed and accuracy of CNN networks.

QuartzNet (Kriman et al., 2020) is a CNN-based architecture that achieves state-of-the-art results in ASR systems. The authors proposed a very deep network utilizing 1D time-channel separable (TCS) convolution layers. Similarly, ContextNet (Han et al., 2020), another CNN-based model, incorporates a squeeze-and-excitation (SE) layer (Hu et al., 2018) to improve ASR accuracy in terms of word error rate (WER). ContextNet employs the SE layer after the convolution layer to capture global information from the audio input, enhancing the system's output. However, despite the advancements in CNN-based ASR, there is growing concern regarding the modeling of long-term context dependencies in the speech signal spectrum. While CNNs can access context at higher layers, they struggle to modulate information from the lower layers effectively. The problem-agnostic speech encoder (PASE) (Pascual et al., 2019) addresses this limitation by combining CNNs and LSTMs in an end-to-end architecture. PASE processes raw speech waveforms as input, generating high-level features that capture essential characteristics of the speech signal. These features can then be used as inputs for other speech processing models or downstream tasks such as speech recognition or speaker identification.

Recurrent Neural Networks (RNNs) are unable to directly map speech input sequences to corresponding textual output sequences (Graves et al., 2006). Output units such as phonemes or other small speech units require additional processing to produce the final transcription (Wang et al., 2019a). Consequently, pre-segmentation of the training data is necessary, and post-processing is required to generate the final label sequence (Graves et al., 2006). Additionally, end-to-end models encounter data alignment challenges when using RNNs and Convolutional Neural Networks (CNNs) to model time-domain features. Since the loss functions in RNNs and CNNs are defined for each point in the input sequence, these models must know the alignment relationship between the input and target sequences for training purposes. Connectionist Temporal Classification (CTC) is a loss function that can be integrated with RNNs and CNNs to address this issue. CTC helps resolve alignment problems while calculating the loss (Wang et al., 2019a). Thus, the introduction of CTC has effectively solved both the data alignment and transcription generation challenges, enabling the use of RNNs and CNNs in end-to-end ASR models. Table 1 summarizes CTC-based ASR models.

BLSTM-CTC, proposed by Eyben et al. (2009), combines a feed-forward layer with two LSTM layers. The authors demonstrated that increasing the number of hidden units in the network structure improves the accuracy of the ASR system. Another approach, presented in Graves and Jaitly (2014), integrates deep bidirectional LSTM layers with CTC objective functions, where audio spectrograms are processed through a deep bidirectional LSTM layer, followed by a CTC loss function as the output layer. In Song and Cai (2015), an end-to-end model was implemented, consisting of two distinct neural networks for phoneme recognition: convolutional layers for frame-level classification, and RNN with CTC for decoding the output sequence. A different combination, presented in Zhang et al. (2016b), uses hierarchical CNNs with a CTC layer without recurrent connections, demonstrating the CNN's ability to capture temporal dependencies. In this model, a

Table 1
CNN and RNN based ASR models.

Architecture	Paper	Data set (duration)	Error rate
RNN/CNN	DLSTM (Graves et al., 2013)	TIMIT	37.6 – 17.7(PER)
	Sak et al. (2014a)	Google Voice Search Task (1900 h)	11.8 – 10.7 (WER)
	two-head cltLSTM (Li et al., 2020)	English Spoken Utterance (200 h–2000 h)	12.24 – 9.34 (WER)
	Highway LSTM (Zhang et al., 2016a)	AMI (100 h)	57.5 – 37.7
	Soltau et al. (2017)	data from YouTube, Google Videos, and Broadcast News	24.0 (WER)
	ltLSTM (Li et al., 2018)	Microsoft Cortana and Conversation Data (30k h)	19.41 – 9.28 (WER)
	Liptchinsky et al. (2017)	WSJ (81 h), Librispeech (1000 h)	19.7 – 17.3 (WER)
	Residual LSTM (Kim et al., 2017a)	AMI (100 h)	57.5 – 37.7 (WER)
	residual LSTM (Zhao et al., 2016)	TIMIT, HKUST (150 h)	50.8 – 39.3 (PER/CER)
	improved ltLSTM (Li et al., 2019b)	Microsoft Anonymized Production (65k h)	19.41 – 9.28 (WER)
	Sainath and Li (2016)	artificially created data (2k h)	15.7 (WER)
	prioritized Grid LSTM (pGLSTM) (Hsu et al., 2016)	AMI (100 h), HKUST (150 h), GALE Mandarin, Arabic MGB	22.54 (WER)
	feedforward sequential memory networks (FSMN)(Zhang et al., 2017)	Switchboard (300 h)	13.2 (WER)
	Graves and Schmidhuber (2005)	TIMIT	29.0 – 18.3 (WER)
	Graves et al. (2005)	TIMIT	–
	time-delay LSTM (TDLSTM) (Moritz et al., 2019b)	(81 h), HKUST (150 h), LibriSpeech (1000 h)	35.5 – 4.6 (WER)
	DBLSTM-HMM (Chen and Huo, 2016)	Switchboard (300 h)	14.7 (WER)
	Chien and Misbullah (2016)	CHiME	11.91 (WER)
	CRNN (Li and Wu, 2015)	HKUST	31.43 (WER)
	Chan and Lane (2015)	Bable (10 h)	83.8 – 67.7 (WER)
	Cai and Liu (2016)	Bable (10 h)	–
	Lee et al. (2009)	TIMIT	–
	Jasper (Li et al., 2019a)	WSJ, Hub5	16.1 – 2.95 (LER)
	TDS convolution (Hannun et al., 2019)	LibriSpeech (1000 h)	7.25 – 3.01 (WER)
	DeCoAR (Ling et al., 2020)	WSJ (81 h), LibriSpeech (1000 h)	10.38 – 4.64 (WER)
	PASE (Pascual et al., 2019)	DIRHA	33.5 – 29.8 (WER)
CTC-Based	Quartznet (Kriman et al., 2020)	WSJ (81 h), Librispeech (1k h)	10.98 – 2.96 (LER)
	Contextnet (Han et al., 2020)	Librispeech (1k h)	1.9 (LER)
	Graves et al. (2006)	TIMIT	30.51 (LER)
	Maas et al. (2014)	WSJ (81 h)	14.1 (WER)
	Deep speech 2 (Amodei et al., 2016a)	WSJ, LibriSpeech, Vox Forge, CHiME (11940 h)	50.7 – 3.1 (WER)
	Song and Cai (2015)	TIMIT	–
	Audhkhasi et al. (2017)	Switchboard-1(300 h), Fisher (1698 h)	20.8 (WER)

stacked convolutional layer creates a large context window for each output, followed by multiple fully connected layers and CTC layers.

A deep RNN layer combined with CTC, applied to large labeled training datasets in two languages – English and Mandarin – is presented in Amodei et al. (2016b). The direct acoustics-to-word CTC model (Audhkhasi et al., 2017) demonstrated results on two well-known benchmark datasets, Switchboard and CallHome. In this model, two techniques were proposed to enhance the training of the ASR model on these datasets. To accelerate the ASR training process, the authors of Hannun et al. (2014a) introduced a partition scheme that improved parallelization and successfully mapped their RNN model to GPUs. Additionally, this model utilized a novel combination of collected and synthesized data, enabling a robust process to handle realistic variations in noise and speaker characteristics. This approach proved to be an efficient method for large-scale data training in ASR tasks.

An important limitation of the CTC model is its assumption that all labels in the output sequence are independent of each other (Wang et al., 2019a), preventing it from effectively modeling languages. As a result, models using CTC must be combined with external language models to improve their final accuracy. In Hannun et al. (2014b), the authors integrated a recurrent neural network with a language model that incorporated a large vocabulary and supported continuous speech recognition. The results highlighted the crucial role of language models in achieving high accuracy in ASR systems. Additionally, other models, such as DeepSpeech2 (Amodei et al., 2016a), further demonstrated the importance of language models within more complex structures. Recently, the CTC loss has been successfully combined with attention-based and transformer-based models, yielding improved results in ASR tasks, as discussed in the following sections. Lee and Watanabe (Lee and Watanabe, 2021) proposed an efficient auxiliary loss function called intermediate CTC loss to improve CTC-based ASR performance. The key innovation is attaching an additional CTC loss to an intermediate layer (typically at the middle) of the CTC encoder network, effectively creating a sub-model that shares lower layers with the full model. This intermediate loss acts as a regularizer for the lower layers while requiring minimal code modification and computational overhead during training, with no overhead during inference. When combined with stochastic depth training on a Conformer network, their approach achieved competitive results with word error rate (WER) of 9.9% on WSJ and character error rate (CER) of 5.2% on AISHELL-1 using only CTC greedy search without any language model — performance comparable to state-of-the-art autoregressive systems. In Do et al. (2021) the authors proposed a novel method for adapting end-to-end speech recognition systems using multiple ASR hypotheses to improve performance in semi-supervised scenarios. By integrating multiple 1-best hypotheses into the CTC loss function during adaptation, the method reduces the impact of transcription errors in pseudo-labels generated from unlabeled data. Hypotheses are derived from systems trained with different acoustic features, such as FBANK and subband temporal envelope (STE). This approach was evaluated on WSJ and CHiME-4 training sets and tested on Aurora-4, achieving significant WER reductions of 6.6% and 5.8% in clean and multi-condition scenarios, respectively, compared to baseline systems adapted with only partially labeled data. The method demonstrates a promising direction for semi-supervised adaptation in end-to-end ASR, achieving improvements without full reliance on manual transcriptions. Additionally, the authors in Nozaki and Komatsu (2021) introduced a method to address the conditional independence limitation of CTC-based ASR. By incorporating intermediate predictions from earlier encoder layers during both training and inference, the approach conditions final predictions on these intermediate outputs, improving recognition accuracy while retaining the simplicity and speed of CTC-based ASR. The method involves adding intermediate CTC losses to specific encoder layers and using their predictions as inputs for subsequent layers, effectively enabling refinement of predictions within the encoder itself. Experiments on datasets such as WSJ, TEDLIUM2, and AISHELL-1 show significant improvements in WER (e.g., over 20% relative reduction on WSJ) with minimal computational overhead, achieving performance comparable to strong autoregressive models with beam search but maintaining much faster decoding speeds. The method demonstrates the potential for balancing speed and accuracy in non-autoregressive ASR systems.

3.7. Methods based on RNN-transducer models

The field of speech recognition has seen remarkable advancements, particularly in the development of objective functions that enhance the accuracy and efficiency of models. Notably, the introduction of the CTC objective function marked a significant milestone, offering a method for models to learn the alignment between audio input sequences and their corresponding text transcriptions without predefined alignments. Concurrently, Google introduced a low-frame-rate objective function (Pundak and Sainath, 2016) aimed at reducing computational requirements while maintaining high recognition accuracy. This innovation allowed for more efficient processing of speech data, contributing to the advancement of real-time speech recognition systems. Additionally, Kaldi, a widely used open-source speech recognition toolkit (Povey et al., 2011), adopted a lattice-free Maximum Mutual Information (MMI) objective function, which further optimized model training by directly maximizing the mutual information between the input speech and its correct transcription. Google's RNN transducer emerged as another pivotal development, offering frame-level speech recognition capabilities akin to CTC but with improved modeling flexibility and accuracy.

A probabilistic Transduction RNN-based system is presented in Graves (2012) which can convert any input sequence into any finite discrete output sequence. This model can improve the acoustic model, language model, and decoding process by applying encoders, a prediction network, and a joint network, respectively. The authors of Graves et al. (2013) improved the previous work in Graves (2012) by replacing the joint network with a fully connection layer. This change increased the depth of the proposed network and increased the accuracy of the system. An investigation of the training of end-to-end speech recognition architectures based on an RNN transducer is shown in Rao et al. (2017). In this work, the authors improved the accuracy of TTS systems in various architectures by using additional text or pronunciation data. Contextual signals were incorporated into RNN-Transducer models to improve the accuracy of end-to-end models (Wu et al., 2020). A novel technique for transferring fine-grained textual knowledge from BERT into RNN Transducer (RNN-T) models for both ASR and spoken language understanding (SLU) is proposed in Sunder et al. (2023). By integrating BERT's advanced language understanding capabilities into RNN-T models through a tokenwise knowledge transfer process, the study significantly enhances ASR and SLU performances. This approach achieves state-of-the-art results on the SLURP dataset for SLU tasks, outperforming existing methods with a more compact model that requires substantially less speech pretraining data. These advancements demonstrate the potential of leveraging textual knowledge to improve speech recognition and understanding systems, offering insights into efficient model training and deployment strategies for real-world applications.

The model in Li et al. (2019b) improved RNN-transducer in two ways: firstly, by optimizing the training algorithm of RNN-transducer, consequently reducing the memory consumption, so it can use mini-batches for faster training. Secondly, the proposed architecture is improved and can obtain better accuracy. In Li et al. (2019b), the authors considered a look-ahead mechanisms to utilize future information in the network to improve the accuracy of ASR systems. The efficient minimum word error rate (MWER) (Guo et al., 2020) is a novel training method for RNN-Transducer models. In this model, after generating alignment scores and an N-best list, the scores of all possible alignments

Table 2
RNN-Transducer based ASR models.

Architecture	Paper	Data set (duration)	Error rate
RNN-Transducer	Graves (2012)	TIMIT	23.2 (PER)
	Graves et al. (2013)	TIMIT	17.7 (PER)
	Rao et al. (2017)	voice-search	8.3 (WER)
	Wu et al. (2020)	voice-Search	5.9 (WER)
	Li et al. (2019b)	Cortana and Conversation	8.75 (WER)
	Guo et al. (2020)	23,000 h training set	0.997 (Normalized WER)
	Sunder et al. (2023)	Switchboard, CallHome	7.2, 14.8 (WER)

for each hypothesis are recalculated in a given N-best list. Forward-backward algorithm is used to calculate the hypothesis probability scores and back-propagation gradients. Due to the success of RNN-transducer-based models in the ASR system, it has been used in Google devices instead of Attention-based Encoder-Decoder (AED) (He et al., 2019) networks. In Jeon and Kim (2020), the authors explore multitask learning, joint optimization, and joint decoding methods for RNN-transducer systems.

However, compared to CTC-based models, RNN-Transducer methods are more complex to train due to their architecture and synchronous decoding constraints. One of the main complexities of this model is that the encoder and prediction network are built from a grid of alignments. Furthermore, applying the forward–backward training, posteriors need to be calculated at each point in the grid (Li et al., 2019b). RNN-Transducer models are shown in Table 2.

3.8. Methods based on attention models

The Neural Transducer (Jaitly et al., 2016) introduced the use of attention mechanisms on input chunks, employing an end-of-chunk symbol for training. One of the primary challenges affecting the models discussed in the previous sections is incremental prediction. This issue arises when new input data arrives or when the model processes long input and output sequences. Neural Transducer models address this by computing the next-step distribution conditioned on the partially observed input. In Raffel et al. (2017), an end-to-end model based on hard monotonic attention was presented for online decoding, achieving linear time complexity. Similarly, Monotonic Chunk-wise Attention (MoChA) was proposed in Chiu* and Raffel* (2018), where a soft attention mechanism is applied to small chunks of data from the input sequence. An improved MoChA-based ASR system was introduced in Kim et al. (2019), where CTC and cross-entropy (CE) losses are used jointly to train the MoChA models, and the Minimum Word Error Rate (MWER) objective is applied to optimize performance. In Watanabe et al. (2017), Kim et al. (2017b), a hybrid CTC-attention architecture was proposed, utilizing CTC loss as a regularization process within an attention-based network. Furthermore, (Miao et al., 2020b) presented an end-to-end hybrid CTC-attention architecture that incorporates stable monotonic chunk-wise attention (sMoChA) for stream-based global attention and a truncated CTC (T-CTC) to compute prefix scores.

Another common output sequence for attention-based end-to-end ASR systems is a character (i.e., *grapheme*) sequence (Bahdanau et al., 2016; Lu et al., 2016). In Kannan et al. (2018), words and sub-word units (WSUs) are used as the language model to be learned in the decoder. The length bias and the corresponding beam problem are the main problems of the attention-based encoder–decoder model, which has been mentioned in Zhou et al. (2020), and a heuristic-based model is not suitable for it; therefore, a beam search structure based on reinterpreting the posterior sequence was proposed. Attention based ASR models are set out in Table 3. In Moritz et al. (2019a) the authors present a novel approach to real-time, streaming speech recognition by integrating CTC with attention-based mechanisms within a unified end-to-end framework. This method capitalizes on the complementary strengths of both CTC and attention models: while CTC excels in aligning input speech frames with their corresponding textual output, making it suitable for streaming applications, the attention mechanism offers superior accuracy by focusing on relevant parts of the audio input when predicting each symbol. By combining these approaches, the proposed system achieves efficient online speech recognition, offering a balance between the real-time processing requirements of streaming applications and the high accuracy demands typical of end-to-end models. A Transformer-based online CTC/attention E2E ASR architecture is presented in Miao et al. (2020a) that synergizes CTC and attention mechanisms within an end-to-end framework. This architecture leverages the Transformer model's ability to handle sequential data with high parallelization and low latency, making it well-suited for real-time speech recognition tasks. By integrating CTC, which excels in temporal alignment of speech frames to textual output, with the attention mechanism's capacity to dynamically focus on relevant segments of the input sequence, the proposed system significantly enhances the accuracy and efficiency of speech recognition.

3.9. Methods based on transformer networks

Transformer Networks have become one of the most popular and powerful models in natural language processing (Vaswani et al., 2017). The architecture of the Transformer model has made it possible to train a stack of self-attention layers (Lin et al., 2017) by applying residual connections between layers, (He et al., 2016) followed by a normalization layer (Irie, 2020). In Liu* et al. (2018), Al-Rfou et al. (2019), a transformer decoder was used as a language model and showed impressive results on different benchmarks. Transformer based ASR models are presented in Table 4.

The Speech-Transformer (Dong et al., 2018) is an end-to-end sequence-to-sequence model that eliminates recurrence, relying entirely on attention mechanisms. It follows the basic structure of a transformer network, but its encoder combines self-attention layers with convolutional layers to approximate hidden representations with character-level granularity. In Sperber et al. (2018), the authors argued that an encoder based solely on self-attention was insufficient for effective acoustic modeling, leading them to propose a combination of self-attention and LSTM layers. A transformer-based acoustic model for hybrid ASR systems was presented in Wang et al. (2020), which evaluated various architectures to encode input sequences using either absolute or relative positional information. Moreover, the application of iterated loss allowed for the training of deeper models based on transformer networks. Wav2vec 2.0 (Baevski et al., 2020) is a Transformer-based framework for self-supervised learning of representations from raw audio. It employs a multi-layer convolutional neural network to encode input data, then masks spans of the resulting latent speech representations, which are subsequently fed into a Transformer network to generate contextualized representations. Another self-supervised learning model, HuBERT (Hsu et al., 2021), also utilizes a Transformer encoder, incorporating an offline clustering step to generate target labels for a BERT-like prediction loss (Devlin et al., 2019).

The Conformer (Gulati et al., 2020) is a hybrid model that combines Transformer layers with convolutional layers to capture both global

Table 3
Attention based ASR models.

Architecture	Paper	Data set (duration)	Error rate
Attention-Based	Neural Transducer (Jaitly et al., 2016)	TIMIT	33.4 – 18.2 (PER)
	Raffel et al. (2017)	TIMIT	16.0 (PER)
	MoChA (Chiu* and Raffel*, 2018)	WSJ	17.4 – (WER)
	Improved MoChA (Kim et al., 2019)	LibriSpeech (1000 h)	8.82 (WER)
	hybrid CTC/attention (Watanabe et al., 2017)	WSJ (81 h), CHiME	43.45 – 11.27 (CER)
	joint CTC-attention (Kim et al., 2017b)	WSJ1 (81 h), WSJ0 (15 h), CHiME	44.99 – 7.36 (WER)
	CTC/attention (Miao et al., 2020b)	LibriSpeech (1000 h), HKUST(200 h)	22.5 – 5.3 (WER)
	Lu et al. (2016)	WSJ (81 h), TIMIT	25.8 (WER)
	Zhou et al. (2020)	WSJ (81 h), TIMIT	15.7 (WER)
	Moritz et al. (2019a)	WSJ (81 h), TED-LIUM, TIMIT	5.7 (WER)
	Miao et al. (2020a)	HKUST (200 h)	23.65 (CER)

content-based interactions and local correlations based on relative offsets. In Wu* et al. (2020), the Lite-Transformer architecture was introduced as an efficient model for mobile natural language processing (NLP). This architecture integrates self-attention with convolutional layers positioned between pairs of feedforward modules. The Conv-Transformer Transducer (Huang et al., 2020), designed for streaming ASR systems, combines a unidirectional transformer with interleaved convolutional layers to capture future context during the audio encoding process. In Hrinchuk et al. (2020), a Transformer-based architecture was proposed for post-processing ASR outputs, producing grammatically and semantically correct final output sequences. The w2v-BERT model (Chung et al., 2021) combines contrastive learning and masked language modeling (MLM). In this approach, contrastive learning trains the model to convert continuous speech signals into a set of distinct speech tokens, while MLM teaches the model to understand speech context by predicting masked tokens in the discretized speech data.

In Tripathi et al. (2022), a Transformer-Transducer architecture was proposed, introducing a training technique that defines both streaming and non-streaming models within a single algorithm. A stack of Transformer layers is used to encode the audio. In Chen et al. (2020), a speech Transformer is combined with a bidirectional decoder (STBD) to jointly learn the encoder and decoder. The encoder in STBD is similar to a standard Transformer encoder but features two unidirectional decoders, each generating targets in opposite directions. Additionally, Mohamed et al. (2019) proposed using convolutional layers in place of positional embeddings in Transformer networks to capture relative positional information. Transformer Encoder Representations from Alteration (TERA) (Liu et al., 2021b) introduces a self-supervised learning method to train Transformer encoders for generating high-quality speech representations. This approach addresses the challenge of training speech recognition systems in low-resource settings where large amounts of labeled speech data are unavailable. TERA learns by reconstructing acoustic frames from their altered counterparts, which vary in time, frequency, and magnitude.

Speech SimCLR (Jiang et al., 2021) is a self-supervised objective for speech representation learning that applies augmentations to both raw speech and its spectrogram. The Speech SimCLR objective combines contrastive loss, which maximizes agreement between differently augmented samples in the latent space, with a reconstruction loss of the input representation. WavLM (Chen et al., 2022) learns universal speech representations from large amounts of unlabeled speech data and adapts effectively across various speech processing tasks. BERT-based Speech pre-Training with Random-projection Quantizer (BEST-RQ) (Chiu et al., 2022) is a self-supervised learning algorithm for speech recognition that masks portions of the speech input and feeds them into the encoder. The encoder learns to predict the masked regions based on unmasked speech signals, with learning targets provided by a random-projection quantizer. ScoutWav (Fatehi et al., 2022) is a model that integrates context-based word boundaries with self-supervised learning, specifically wav2vec 2.0, to develop a low-resource ASR model. ScoutWav pre-trains a model on high-resource environment (HRE) datasets and then fine-tunes it using low-resource environment (LRE) datasets to learn context-based word boundaries. These boundaries are used to fine-tune a pre-trained, iteratively refined wav2vec 2.0 model, which learns appropriate representations for the downstream ASR task. LABERT (Fatehi and Kucukyilmaz, 2023) combines an active learning approach with a Local Aggregation (LA) function to detect and represent hidden speech units effectively. Active learning is used to select informative and diverse speech samples, while the LA function groups similar units and separates dissimilar ones in the latent space, addressing the limitations of noisy clustering processes. LABERT employs a BERT-based masked prediction model for pre-training and fine-tuning and introduces layer selection using Canonical Correlation Analysis (CCA) to identify the most suitable features for downstream LRE ASR tasks. Evaluated on LRE datasets like UASpeech and iCUBE, LABERT achieves significant improvements in WER compared to state-of-the-art models, demonstrating its scalability and effectiveness in addressing data bottlenecks in LREs.

4. Low-resource environments

Recent developments in the field of deep learning have led to renewed interest in combining the main components of speech recognition systems into a single end-to-end model, which can directly map the input audio sequence into the output text sequences (Hsu et al., 2020). However, current state-of-the-art techniques for training end-to-end models require a considerable amount of labeled data (Hsu et al., 2020). Therefore, evidence suggests that a large volume of recorded and transcribed speech (i.e., a spoken corpus) is needed to create an accurate and robust automatic speech recognition system in a new domain or application (Meyer, 2019).

This is a challenging prospect for those interested in carrying out research in domains where large amounts of data for training are not available. We define low-resource environments as environments where the lack of sufficient amount of training data diminishes the performance of the ASR system. Examples of this include domains such as new or less wide-spread languages (e.g., the Kyrgyz language) (Meyer, 2019), domains in which a highly technical or specific language is required (e.g., a chemical plant, or a surgery theatre), child speech recognition (Wu et al., 2020), speakers with speech disorders (e.g., dysarthria) (Meyer, 2019), or speakers with accents are all examples of low-resource environments. In these environments, to the best of our knowledge, there are very limited suitable public corpora for training purposes.

Acoustic models for ASR systems are especially crucial for endangered and unwritten languages. Traditional models, like the HMM (Gales, 1998) and GMM (Povey et al., 2011), operate based on state transitions. In contrast, sequence processing models such as LSTM (Wang et al.,

Table 4
Transformer based ASR models.

Architecture	Paper	Data set (duration)	Error rate
Transformer	Al-Rfou et al. (2019)	TEDLIUM (200 h)	38.6 - 15.82 (WER)
	Sperber et al. (2018)	LibriSpeech (1000 h)	2.26 (WER)
	Wang et al. (2020)	LibriSpeech (1000 h)	11.28 – 4.3 (WER)
	HuBERT (Hsu et al., 2021)	LibriSpeech (1000 h)	6.8 – 1.7 (WER)
	wav2vec 2.0 (Baevski et al., 2020)	LibriSpeech (1000 h), TIMIT	15.6 – 1.9 (WER)
	Conformer (Gulati et al., 2020)	AISHELI-1 (178 h)	10.56 – 6.64 (CER)
	Lite-Transformer (Wu* et al., 2020)	LibriSpeech (1000 h)	14.7 – 5.2 (WER)
	Hrinchuk et al. (2020)	LibriSpeech (1000 h)	8.3 – 3.4 (WER)
	Speech-transformer (Dong et al., 2018)	WSJ (81 h)	12.2 – 10.92 (WER)
	Tera (Liu et al., 2021b)	LibriSpeech (1000 h), TIMIT	8.31 – 6.01 (WER)
	w2v-BERT (Chung et al., 2021)	LibriSpeech (1000 h), Libri-Light(60k h)	5.0 – 1.3 (WER)
	Speech SimCLR (Jiang et al., 2021)	LibriSpeech (1000 h), TIMIT, IEMOCAP	15.1 - 5.89 (WER)
	WavLM (Chen et al., 2022)	VoxCeleb1, VoxCeleb2	29.2 – 4.0 (WER)
	BEST-RQ (Chiu et al., 2022)	LibriSpeech (1000 h)	2.7 – 1.4 (WER)
	ScoutWav (Fatehi et al., 2022)	LibriSpeech (1000 h), WSJ, TEDLIUM, Common Voice	10.14 – 24.55 (WER)

2017), CTC (Chan et al., 2016), and Transformer (Miao et al., 2020a) leverage an end-to-end approach for processing sequences. Tachbelie et al. (2014) applied HMM for acoustic modeling of Amharic, an endangered language, using syllables as the modeling unit. Amharic speech was automatically segmented into syllables and recorded. However, HMM's frame-by-frame training approach, which outputs single-frame frequencies, was inadequate in capturing the contextual semantic relationships within speech sequences. To address this limitation, researchers transitioned to sequence-based, end-to-end deep learning models for more effective modeling. Inaguma et al. (2019) utilized BLSTM-CTC to recognize languages with limited data, integrating transfer learning by leveraging knowledge from larger language corpora to enhance recognition accuracy. This method outperformed the BLSTM-HMM hybrid model. However, transfer learning is ineffective for recognizing languages with significantly different acoustic spaces. Transfer learning is designed to apply previously acquired knowledge to new tasks when data is scarce, but significant differences between the source and target languages can impede the learning process of the target language. In addition, speech enhancement, the primary method for reducing noise interference, is predominantly applied to the speech recognition of major languages (Pandey and Wang, 2018). Yu et al. (2019) extended this application to endangered languages by introducing an end-to-end speech enhancement model based on an improved deep convolutional generative adversarial network (DCGAN) to enhance the speech quality of the Tujia language. For the enhanced speech, Yu et al. conducted a perceptual evaluation of speech quality and calculated the mean opinion score for listening quality objectives.

Generalization of ASR models to new speakers, or speaker adaptation, is crucial for low-resource scenarios. Speaker-adapted recognition allows model parameters to reflect features of multiple speakers. Ochiai et al. (2018) improved model adaptability by retaining the acoustic model's parameters while varying speaker features during training. Parameters are re-evaluated when adapting to target speakers. However, insufficient data for adaptation training can lead to overfitting, resulting in unsatisfactory adaptation outcomes. Current recognition methods primarily rely on acoustic models, but their accuracy is nearing a plateau (Ivanko et al., 2018). Consequently, multimodal approaches are gaining traction. Audio-visual fusion speech recognition (AVSR) leverages the bimodal speech perception mechanism (Yu et al., 2023), combining audio and visual modalities to enhance transcription accuracy. The visual modality provides complementary information to the audio modality, reducing overall uncertainty and improving accuracy when one modality encounters errors. AVSR offers a viable solution to the limitations of single-modality recognition, addressing the challenges of insufficient training data and low accuracy.

This indicates the need to understand the various perceptions of low-resource environments that exist in ASR systems. As shown in Section 3, most state-of-the-art models need more than 2k hours of transcribed audio as training data (Meyer, 2019). Such requirements are simply unattainable in low-resource environments. And, as our experiments in this paper show, benchmark corpora and models architecture prove to be insufficient to achieve robust ASR systems using models designed for high resource environments. Consequently, special attention should be considered when developing low-resource ASR systems to account for such limited training data.

Large language models (LLMs) (Chowdhery et al., 2023; Touvron et al., 2023) have demonstrated remarkable flexibility, capable of tackling a wide array of tasks. Trained on extensive unsupervised text data to predict the next token, these systems encode world knowledge within their network parameters, making them effective for various open-domain generative tasks such as abstractive summarization, question answering, knowledge retrieval, text generation, and machine translation. However, text-based interaction with LLMs can be limiting, as many structured modalities encode information difficult to capture through text alone. For instance, audio can convey a range of emotions in speech, and images can depict the geometry and location of objects, which might be challenging to describe textually. Recent advancements have extended LLMs to process other modalities. The multi-modal PaLM-E (Driess et al., 2023) combines a large pre-trained visual transformer (Dehghani et al., 2023) with the PaLM LLM (Chowdhery et al., 2023), achieving state-of-the-art performance in robotics tasks. Similarly, Zhu et al. (2023) integrates a pre-trained visual model with the large language model Vicuna, derived from LLaMA (Chiang et al., 2023), creating a model capable of reasoning with both visual and textual inputs. Furthermore, Gong et al. (2023) introduces LTU, an extension of LLaMA with an aligned audio encoder trained on an audio question-answering corpus, enabling it to understand and reason with sounds. However, LTU's speech understanding and recognition abilities remain limited.

Given the vast number of parameters in large language model-based systems, fully adapting these systems to new tasks is often computationally impractical and costly. In Zhu et al. (2023), the authors proposed a highly parameter-efficient approach by training a single projection layer to align the outputs of the visual encoder with the language model. However, this approach significantly restricts the system's adaptability and performance on new tasks. On the other hand, the multi-modal PaLM-E (Driess et al., 2023) explored jointly training both the visual encoder and language model, but this method is prohibitively expensive and impractical. Alternative strategies, such as adding adapter layers (Houlsby et al., 2019) or prefix embeddings (Li

and Liang, 2021), are more parameter-efficient and can be trained for new tasks. However, these methods increase inference costs. Low-Rank Adaptation (LoRA) (Hu et al., 2022) addresses this issue by using low-rank matrices to adjust specific system parameters, proving to be both memory-efficient during training and unaffected by inference runtime. Despite its promise, the integration of speech and large language models (LLMs) remains a largely unexplored area. One challenge is aligning speech and text using pretrained LLMs, as speech signals are typically much longer than text sequences. Additionally, given the high cost of training LLMs, minimizing integration costs while maintaining performance remains a significant challenge.

Numerous data augmentation techniques have been proposed for ASR systems, primarily focusing on enhancing speech data. Speed perturbation (Ko et al., 2015), pitch adjustment (Shahnawazuddin et al., 2016), noise addition (Tóth et al., 2018), and vocal tract length perturbation modify audio by adjusting speed, pitch, or adding noise to the original clean signal. Another recent approach, SpecAugment (Park et al., 2019), masks the mel-spectrogram in both time and frequency dimensions, resulting in improved recognition accuracy. Additionally, a study (Wang et al., 2019b) explores semantical relationships by masking speech sequences in the time domain based on text alignment.

MixSpeech (Meng et al., 2021) introduces a novel data augmentation method specifically designed to enhance the performance of ASR systems in low-resource settings. MixSpeech creates a weighted combination of two different speech features (e.g., mel-spectrograms or MFCCs) and uses this combined input to train ASR models. The method presents a new approach to mixup by blending both the input features and the recognition losses of the respective text sequences, using the same combination weight. The authors evaluated MixSpeech on two popular end-to-end ASR models – Listen, Attend and Spell (LAS) and Transformer – and conducted experiments on several low-resource datasets, including TIMIT, WSJ, and HKUST. Their results demonstrate that MixSpeech consistently outperforms baseline models and even surpasses the widely used SpecAugment technique in terms of accuracy. The effectiveness of MixSpeech is attributed to its simplicity, requiring only a single hyperparameter, and its ability to provide contrastive signals that improve the model's ability to differentiate between mixed speech inputs.

MixRep (Xie and Hansen, 2023), a related data augmentation method, further improves low-resource ASR by interpolating hidden representations within neural networks. This approach generalizes the MixSpeech method by applying mixup to both acoustic input features and hidden layer outputs. MixRep also combines mixup with regularization along the time axis, enhancing model robustness and generalization. Experimental results on the WSJ and Switchboard datasets show that MixRep outperforms other regularization methods, achieving significant reductions in WER compared to strong baselines such as SpecAugment.

5. Datasets

In the preceding section, different ASR techniques were discussed, which need a large amount of data for training. This section provides an overview of the datasets that can be used to train and evaluate ASR models. Furthermore, we present our own low-resource dataset, iCUBE, which has been used to test state-of-the-art ASR techniques in low-resource task.

5.1. Datasets for HRE ASR task

In this section, we summarize datasets which provide speech and the corresponding transcripts, speaker labels, or a large amount of speech data but with limited or no labels.

Librispeech (Panayotov et al., 2015) is a large-scale corpus (more than 1k hours) of read English speech that has been widely used to train and evaluate ASR tasks. This corpus is created from audio-books that are part of the LibriVox project and contains more than 2000 h of speech sampled at 16 kHz (Panayotov et al., 2015). Speakers in Librispeech are divided based on lower-WER speakers and higher-WER speakers, which are cleaned and pre-processed. The training portion of the corpus is divided into three subsets with approximate size of 100, 360, and 500 h.

The Wall Street Journal (WSJ) corpus (Paul and Baker, 1992) consists of speaker-independent (SI) read material, divided into training, development, and evaluation test sets. The training set includes 90 utterances from each of 92 speakers, intended for training speech recognition models. An additional 48 speakers each read 40 sentence utterances, with half containing only words from a fixed 5000-word vocabulary and the other half from a 64,000-word vocabulary, which are used as testing material. All 140 speakers also recorded a common set of 18 adaptation sentences. The corpus incorporates standard close-talking and multiple secondary microphones, with an equal number of male and female speakers to ensure diversity in voice quality and dialect (Paul and Baker, 1992). All materials were sourced from the WSJ text corpus and recorded in a clean environment using close-talking microphones.

Fisher corpus (Cieri et al., 2004) is based on the Fisher telephone conversation collection protocol, which was proposed by the Linguistic Data Consortium (LDC). Fisher data collection asked participants to speak on an assigned topic that was randomly selected from a list that changed periodically. This strategy allowed them to cover a large vocabulary (Cieri et al., 2004). The main purpose of the data collection protocol in Fisher was to be able to produce over 2k hours of conversational speech data from calls. After 11 months, LDC was able to collect 16,454 calls, with an average of 10 min in duration, totaling 2972 h of audio (Cieri et al., 2004). In Fisher, 53% of calls were made by female, with 38% of subjects are aged between 16 and 29, 45% are aged between 30 and 49, and 17% are aged over 50 (Cieri et al., 2004).

VoxCeleb (Nagrani et al., 2017) is a large-scale speaker identification and audio-visual dataset which contains around 100,000 utterances of 1251 celebrities, short clips of human speech, extracted from interview videos uploaded to YouTube. VoxCeleb has about 2000 h of speech. VoxCeleb is gender balanced in which 55% of speakers are male and selected from a wide range of different ethnicity, accents, professions, and ages.

TED-LIUM is a corpus that contains audio transcriptions of TED talks. TED-LIUM is presented in 3 different versions which are TED-LIUM Release 1 (Rousseau et al., 2012), TED-LIUM Release 2 (Rousseau et al., 2014) and TED-LIUM Release 3 (Hernandez et al., 2018). TED-LIUM Release 3 contains 2351 audio talks in NIST sphere format (SPH) and includes talks from TED-LIUM Release 2 and 452 h of audio.

Common Voice (CV) (Ardila et al., 2019) is an open-source dataset designed to provide a diverse collection of speech recordings from speakers of varying ages, genders, and accents, with the goal of supporting the development of more inclusive and accurate speech recognition systems. As of version 7.0, the Common Voice corpus contains approximately 11,000 h of audio in 76 different languages. The dataset is continuously expanding, as new recordings are contributed by volunteers. In addition to the audio recordings, the Common Voice corpus includes metadata such as the speaker's age, gender, and accent, as well as information about the recording environment and any background noise. This metadata is valuable for training and evaluating speech recognition models, especially those that aim to be robust to variations in speaker characteristics and acoustic conditions.

5.2. Datasets for LRE ASR task

A low-resource speech recognition dataset refers to a collection of speech data that is limited in size, quality, or diversity, making it challenging to train robust speech recognition models. Low-resource speech recognition datasets are particularly challenging for automatic speech recognition (ASR) systems because they may lack sufficient

examples of rare or out-of-vocabulary words or may contain significant amounts of noise or speaker variation, which can lead to poor recognition performance. TORGO (Rudzicz et al., 2012) is a low-resource dataset which contains approximately three hours of speech. TORGO consists of aligned acoustic recordings from 15 speakers, including 7 control speakers without any disorder and 8 speakers with different levels of dysarthria. Speakers were asked to read single words or sentences and describe the content of some photos. A total of, 5980 and 2762 utterances were recorded from healthy and dysarthria speakers, respectively.

Nemours (Menendez-Pidal et al., 1996) database is a low-resource speech collection of 74 short sentences spoken by 11 speakers with varying degrees of dysarthria, resulting in a total number of 814 recordings. Furthermore, Nemours contains two connected speech paragraphs, which are produced by each of the 11 speakers.

UASpeech (Kim et al., 2008) database is the largest corpus of dysarthria speech in American English. It is a collection of 541 read speech recordings from 19 individuals with cerebral palsy. The prompt words include three repetitions of the first ten digits, three repetitions of 26 radio alphabet letters, three repetitions of 19 computer commands, common words from the 'Grandfather Passage', and uncommon words from phonetically balanced sentences one time each.

5.3. iCUBE: a human-robot interaction dataset

The iCUBE[1] dataset was developed during the first experimental phase of the iCUBE project (Industrial Co-Bots Understanding Behavior), aimed at addressing the limitations of current collaborative robots, or co-bots, which often lack the ability to naturally interpret human behavior. Traditional co-bots rely on explicit mechanical and hierarchical instructions, without fully integrating more intuitive human inputs like gestures, facial expressions, or language to predict behavior. This gap hampers true collaboration and leaves humans unable to fully comprehend the robot's decision-making process. By integrating advances in online and reinforcement learning, the iCUBE project seeks to endow co-bots with the ability to sense and interpret human actions, language, and expressions, enabling more seamless human–robot collaboration. In addition, iCUBE is currently integrating the BlueMax human sensing component, which will enable co-bots to sense the user's facial expressions and body gestures, further enhancing interaction.

In this experimental phase, participants interacted with an actor posing as a robot, using natural language, facial expressions, and gestures to teach the robot how to sort laundry. The experiments were designed to simulate real-life human–robot collaboration, where participants could either instruct or demonstrate sorting tasks, or simply express their satisfaction with the robot's actions. Soon, iCUBE will integrate NLP tools, such as ASR, dialogue management, and TTS, allowing users to communicate with co-bots through speech. Throughout the experiments, the robot also responded to participant actions and speech. The dataset comprises 42 video recordings, totaling more than 300 min of footage, capturing the entire interaction process, including both visual and audio data. This dataset provides a rich source for studying human–robot interaction in an industrial context, with potential implications for improving co-bots' abilities to learn tasks implicitly from human behavior.

6. Experiments

In this section, we describe the series of comprehensive experimental evaluations we carried out to sufficiently investigate the performance of state-of-the-art HRE approaches in ASR systems in low-resource environments. Section 6.1 describes the methodology used to evaluate the different models in this paper. Evaluation metrics are presented in Section 6.2.

[1] https://www.horizon.ac.uk/industrial-co-bots-understanding-behaviour-icube/

6.1. Evaluation protocol

Our evaluation protocol was designed to obtain evidence on the two questions we present at the beginning of our study. Namely, given a low-resource environment:

- What is the performance achieved by training models using only high-resource benchmark data and testing on low-resource datasets?
- What is performance benefit achievable by pre-training with high-resource benchmark data and fine-tuning the trained model with low-resource data?

We selected two well-known benchmark datasets for pre-training the different ASR methods in this study: LibriSpeech (Panayotov et al., 2015) and WSJ (Paul and Baker, 1992). These datasets are commonly used for evaluating high-resource ASR systems and share similar characteristics: multiple speakers, clean read speech (sourced from texts), and recordings at a 16 kHz sampling rate (Chorowski et al., 2019). To analyze model performance in low-resource tasks, we selected UASpeech, which was used to test the models after pre-training on LibriSpeech and WSJ. In terms of models, we chose different network architectures that have achieved state-of-the-art results in recent years. For LSTM-based networks, we trained and tested LSTM, BLSTM, ltLSTM (Li et al., 2018), cltLSTM (Li et al., 2019b), and Residual LSTM (Zhao et al., 2016; Kim et al., 2017a). Each network was trained with 2, 4, 6, and 8 layers. Additionally, we examined different configurations of 2-layer LSTMs concatenated with fully connected FNNs to evaluate their performance. The combination of CNNs with 2-layer LSTM and 2-layer GRU architectures was also explored to assess their potential for low-resource ASR systems. Furthermore, we selected the basic Transformer model with 6 encoders and 6 decoders to evaluate its performance in a low-resource environment. Finally, we included models such as QuartzNet, wav2vec 2.0, HuBERT, and domain-adaptive self-supervised training (Do et al., 2023), which have demonstrated promising results in ASR tasks.

We consider two training scenarios:

- Train all models from scratch using Librispeech and WSJ separately. Test on the relevant dataset, UASpeech and iCUBE.
- Pre-train all models from scratch using Librispeech and WSJ separately. Fine-tune with the UASpeech and iCUBE datasets. Finally, test on UASpeech and iCUBE.

In both scenarios, we applied 10-fold cross-validation during training, and reported average results with standard deviations. For the pre-training scenario, we split the iCUBE and UASpeech datasets into ten folds and, in each iteration, nine folds were used as the fine-tuning data for the trained model and the remaining fold as the test set. To ensure that all folds are tested, ten iterations are performed. The output alphabet of the target text consisted of 31 classes and 26 lowercase letters.

Since we also wanted to focus on the role of the amount of data during training and its effects, we trained all models in both scenarios with 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 100% of training data in both datasets (LibriSpeech and WSJ).

Finally, to focus on verifying the models with a fair comparison, for all methods which are based on stacked LSTM layers, we use 1024 hidden units, and the output of each LSTM layer is reduced to 512 using a linear projection layer. Furthermore, to examine the FNN methods, we use *tanh(.)* as the activation function for the hidden layer and the *softmax* function for the output layer. As a pre-processing step, we compute Mel Spectrograms to convert the input raw audio into the ASR model. We use the *AdamW* optimizer (Loshchilov and Hutter, 2019) as a hyperparameter setting with an initial learning rate of 0.001. Different models will be trained to predict the probability distribution of all characters in the alphabet by using CTC loss function.

6.2. Metrics

We present the results according to two metrics:

Word error rate (WER) : WER is a standard metric for measuring ASR performance. WER is a word-level measure which takes the predicted transcription of the model and the ground truth transcription, and measures the Levenshtein distance. Levenshtein distance is a minimal number of insertions, deletions, and substitutions of words for the conversion of a hypothesis to a Ref. Chuangsuwanich (2016). This metric is calculated as follows:

$$\begin{aligned} WER &= SubstitutionError + InsertionError + DeletionError \\ SubstitutionError &= \frac{Numberofsubstitutionerrors}{Numberofgroundtruthwords} \\ InsertionError &= \frac{Numberofinsertionerrors}{Numberofgroundtruthwords} \\ DeletionError &= \frac{Numberofdeletionerrors}{Numberofgroundtruthwords} \end{aligned} \tag{9}$$

WER is normally reported as a percentage. Chuangsuwanich (2016).

Character error rate (CER) : CER is an important metric in ASR system evaluation. CER measures the error of the characters between the predicted transcription of the model and the ground-truth transcription. The calculation of the CER is similar to the WER, but it is a character-level measure.

As we carried out cross-validation during our evaluation, we report on average WER and CER across all folds, along with their associated standard deviation.

7 Results

In this section, we present the results obtained by examining different ASR methods on Librispeech and WSJ. We pre-train different models with different percentage of data and then test with low-resource iCUBE and UASpeech data. The best WERs obtained from each model trained on Librispeech and WSJ, and tested on iCUBE are summarized in Table 5. Complete results based on different numbers of layers and percentage of data are explicitly listed in Appendix.

Among LSTM with different numbers of layers (Refer to Appendix for complete comparison), the 6-layer LSTM model performs the best when trained with 100% of the data, achieving WERs of 24.28% and 24.17% in WSJ and Librispeech, respectively. The WER of the model trained with 10% of the WSJ dataset is, 45.79% while it is 45.23 for the Librispeech. When the amount of data increases from 10% to 20%, WERs decrease 5.61% for the WSJ and 4.06% for Librispeech. A significant improvement in WER of 10.15% occur when the amount of the data increase from 30% to 40% in WSJ while this improvement is 8.64% for Librispeech.

The 6-layer ResLSTM outperforms the other number of layers (see Appendix for complete comparison) and gets 25.13%, 24.14% WERs when trained with 100% of WSJ and Librispeech, respectively. By increasing the amount of pre-trained WSJ data from 40% to 50%, WER is improved by 10.07%, which is a significant improvement, whereas WER is improved by 9.84% for Librispeech.

The 6-layer ltLSTM model performed better than the other number of layers in ltLSTM as well as the previous two models (see Appendix for the complete comparison). This model achieved 24.11% and 24.08% WERs when trained with 100% of the data in the two datasets. The WER is improved 9.98% when the data increased from 40% to 50% on WSJ. Although a slight improvement of 2.43% occurred in Librispeech when the model received 10% more data from 90% to 100%. The same number of layers in cltLSTM outperforms all other stack LSTM structures. This model obtained a WER of 23.91% and 26.18% based on the 6 number of layers in its structure on WSJ and Librispeech datasets, respectively. Interestingly, this improved WER in cltLSTM is related to the use of future context frames.

In our results, we evaluate three different configurations of the 2-layer LSTM and the fully connected FNN. The first structure, called FNN-LSTM, is created by cascading 2-layer LSTM after FNN. This model achieved WERs of 33.97% and 29.34% when trained on WSJ and Librispeech. This model achieved a WER of 51.38% when it received 10%5 of the WSJ data and improved by 19.07% when increasing the data amount to 50%. Meanwhile, increasing the amount of data from 60% to 70%, the WER got 8.71%better.

In the second configuration, we insert a FNN layer between two 2-layers of LSTMs, called LSTM-FNN-LSTM, to present another architecture for the combination of the LSTM and FNN. This structure obtained 34.01% and 30.39% WERs on WSJ and Librispeech, respectively. Finally, we create a different model by cascading two FNN layers after one 2-layer LSTM, called LSTM-FNN-FNN, which achieved WERs of 33.92% and 29.18% on WSJ and Librispeech, respectively. The LSTM-FNN-FNN structure achieved 7.63% WER improvement when its data increased from 70% to 80%. In combination of the LSTM and FNN layers, the LSTM-FNN-FNN structure obtained a better WER than the other models.

We evaluated the performance of a 2-layer BLSTM on LibriSpeech and WSJ, achieving WERs of 27.18% and 26.07%, respectively. Initially, the BLSTM had WERs of 44.74% on WSJ and 44.65% on LibriSpeech. The 2-layer BLSTM performed similarly to the 2-layer LSTM but demonstrated better results. However, when a 1-D CNN layer was added before the 2-layer BLSTM, the WER increased slightly, reaching 26.19% and 27.63% on WSJ and LibriSpeech, respectively. The WERs for the 1-D CNN and 2-layer BLSTM individually were 44.92% and 45.12%. When using 10% of the data, adding a 1-D CNN layer before the 2-layer BLSTM decreased the WER by 0.4%. However, when a 1-D CNN was cascaded before the 2-layer LSTM, the WER increased, resulting in WERs of 27.22% and 28.78% on WSJ and LibriSpeech. We found that the hybrid model combining the 1-D CNN with the 2-layer LSTM exhibited similar performance to other models. Additionally, when a 2-layer GRU was added after the 1-D CNN, the WERs were 27.13% and 28.63% on WSJ and LibriSpeech, respectively. Notably, the combination of the CNN with BLSTM outperformed the other CNN-based combinations.

In addition, to investigate the effects of utilizing a large amount of pre-training data on the Transformer model in a low-resource environment, we examined the base model of the Transformer with 6 Encoders and 6 Decoders. This model achieved WERs of 22.32% and 21.27% when trained on WSJ and Librispeech, respectively. In this model, by increasing the Librispeech data amount from 40% to 50%, WER improved by 8.89%, while this improvement on WSJ was 7.75%. Compared with other previous models, Transformer achieves a large margin improvement.

Finally, we examine the same strategy for pre-training for QuartzNet, wav2vec 2.0, HuBERT and Domain Adaptive SSL models. Domain Adaptive SSL, wav2vec 2.0 and HuBERT have similar WERs and outperform QuartzNet. HuBERT obtained 21.52% and 20.15% WERs on WSJ and Librispeech, respectively. While wav2vec 2.0 achieved 21.65% WER on WSJ and 20.14% WER on Librispeech.

Similar results are presented in Table 6 when the same pre-trained models are tested on the UASpeech dataset.

Figs. 1, 2 shows the obtained CER on after training different models on WSJ and Librispeech datasets and testing with iCUBE and UASpeech, respectively. On both datasets, Transformer obtained the best results and at each stage the CER is improved by increasing the amount of training data.

These experiments showed that increasing the amount of data can improve the performance of the ASR system in different architectures, however the WERs tend to be high when models are tested on LRE datasets. Our second series of experiments aims demonstrate the performance of the models when the training and testing data are from the same domain. Therefore, we use LibriSpeech and WSJ datasets to train the test the models. The best results based on the number of layers

Table 5
Best results for each model when pre-training on WSJ and Librispeech and testing on iCUBE. The columns of the table denote the percentage of pre-training data used.

		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
6-layer LSTM	WSJ	45.79	43.22	40.38	36.28	32.61	31.48	29.12	27.86	25.53	24.28
	LibriSpeech	45.23	43.39	41.18	37.62	34.21	32.78	30.67	27.49	25.31	24.17
6-layer ResLSTM	WSJ	45.91	43.87	40.92	37.42	33.65	32.71	29.83	28.29	26.94	25.13
	LibriSpeech	45.68	43.62	41.34	37.57	33.87	32.15	30.18	27.13	24.91	24.14
6-layer ltLSTM	WSJ	45.83	43.51	39.41	35.94	32.35	30.52	28.97	27.18	25.09	24.11
	LibriSpeech	45.39	43.53	40.91	37.29	33.62	31.87	28.73	26.42	24.68	24.08
6-layer cltLSTM	WSJ	45.69	43.19	39.27	35.67	32.14	30.27	28.59	26.76	24.71	23.91
	LibriSpeech	45.28	43.27	39.17	36.73	33.41	31.69	30.42	28.03	26.13	26.18
FNN+2-layer LSTM	WSJ	51.38	49.71	46.13	43.19	41.58	39.75	37.13	35.81	34.69	33.97
	LibriSpeech	50.39	49.13	45.48	42.61	40.13	38.79	35.41	33.29	31.63	29.34
2-layer LSTM+FNN+ 2-layer LSTM	WSJ	51.69	50.37	47.62	44.51	41.85	40.19	38.61	37.61	35.82	34.88
	LibriSpeech	50.73	50.21	46.73	43.67	41.76	40.08	38.19	35.83	33.12	30.39
2-layer LSTM+FNN+ FNN	WSJ	51.53	50.18	47.27	44.23	41.31	39.87	37.63	35.47	34.81	33.92
	LibriSpeech	50.47	49.89	46.91	43.39	41.23	39.72	37.21	34.37	32.59	29.18
2-layer BLSTM	WSJ	44.74	42.21	40.28	36.93	33.81	31.12	29.89	28.31	26.83	26.07
	LibriSpeech	44.58	41.98	40.19	37.28	34.93	33.96	30.38	29.13	28.07	27.18
1-D CNN+2-layer BLSTM	WSJ	44.92	42.53	40.63	37.61	34.28	31.57	30.31	28.91	27.12	26.19
	LibriSpeech	45.12	42.39	40.51	37.59	35.42	34.21	30.82	29.89	28.69	27.63
1-D CNN+2-layer LSTM	WSJ	47.38	45.61	43.12	41.17	38.62	35.58	33.62	31.92	29.71	27.22
	LibriSpeech	48.17	46.21	43.69	41.58	39.27	36.21	34.87	33.47	30.19	28.78
1-D CNN+2-layer GRU	WSJ	47.69	46.13	44.33	41.89	39.58	36.71	33.89	31.29	29.48	27.13
	LibriSpeech	48.81	47.21	45.31	42.43	39.81	36.32	33.51	31.87	29.64	28.63
QuartzNet	WSJ	44.27	42.18	38.85	35.39	31.98	30.11	28.11	25.98	23.89	22.85
	LibriSpeech	44.11	41.53	38.32	35.79	31.72	29.93	27.98	25.65	23.51	22.13
Transformer	WSJ	43.78	41.28	38.31	34.97	31.52	29.92	27.71	25.61	23.46	22.32
	LibriSpeech	42.19	40.87	37.92	35.32	31.49	29.67	27.43	25.29	23.19	21.27
wav2vec 2.0	WSJ	38.15	36.92	34.15	31.18	29.15	27.34	25.83	24.91	22.74	21.65
	LibriSpeech	36.83	35.13	33.98	30.29	28.51	26.12	24.13	23.28	21.29	20.41
HuBERT	WSJ	38.49	36.87	34.11	30.92	29.05	27.10	25.51	24.70	22.62	21.52
	LibriSpeech	36.95	35.05	33.58	30.13	28.17	25.93	24.03	23.12	21.30	20.15
Domain Adaptive SSL	WSJ	38.18	37.12	33.81	30.68	27.55	26.33	25.21	24.23	22.33	20.28
	LibriSpeech	36.65	34.93	33.18	30.23	28.11	25.77	23.93	23.02	20.91	**20.05**

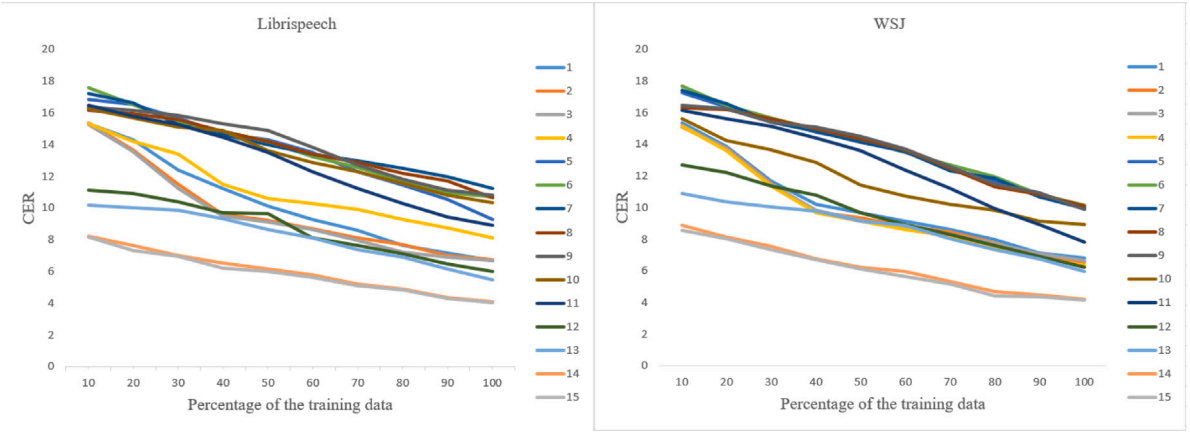

Fig. 1. CER in percentage for models trained with WSJ and Librispeech and tested with iCUBE. 1: 6-layer LSTM, 2: 6-layer ResLSTM, 3: 6-layer ltLSTM, 4: 6-layer cltLSTM, 5: FNN+2-layer LSTM, 6: 2-layer LSTM+FNN+2-layer LSTM, 7: 2-layer LSTM+FNN+FNN, 8: BLSTM, 9: CNN+2-layer BLSTM, 10: CNN+2-layer LSTM, 11:CNN+2-layer GRU, 12: Transformer, 13:QuartzNet, 14:wav2vec 2.0, 15:HuBERT.

and the percentage of data in each model are presented in Table 7. The complete results are listed in Appendix.

The performance of the 6-layer LSTM improved in terms of WERs by 13.77% on WSJ and 13.41% on LibriSpeech when the amount of data increased from 10% to 20%. The improvements are 17.65% and 12.37% respectively in WSJ and Librispeech when the training data amount is increased by 10 percent from 30% to 40%. Almost the same amount of improvement is seen in other models. The most interesting result obtained is when the percentage of the data is increased from 90% to 100%, where the rate of improvement is 3.88%5 and 6.26% on WSJ and Librispeech, respectively. The obtained results by Transformer in terms of WER is 14.21% in WSJ and 13.73% in Librispeech, while QuartzNet, wav2vec 2.0 and HuBERT achieved 14.38%, 6.21, 6.13% WERs on Librispeech, respectively. These results emphasize the importance of the volume of the training data and the relevance of training and test data.

In the third series of our experiments, we aimed to tackle the issue of domain difference in training and testing data for LREs by introducing a fine-tuning step after pre-training the model with HRE datasets. Here, the trained models are fine-tuned by in-domain LRE data

Table 6
Best results for each model when pre-training on WSJ and Librispeech and testing on UASpeech.

		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
6-layer LSTM	WSJ	65.71	59.83	54.48	50.11	47.45	44.51	41.19	39.32	37.12	36.27
	LibriSpeech	63.39	58.72	52.21	49.39	47.35	43.52	39.74	37.89	36.28	35.94
6-layer ResLSTM	WSJ	64.15	59.98	65.51	52.39	48.18	43.37	42.71	40.28	38.51	38.19
	LibriSpeech	63.57	59.38	55.32	51.49	48.61	42.58	41.39	39.15	38.93	37.17
6-layer ltLSTM	WSJ	64.89	59.13	54.23	50.17	47.13	43.78	40.85	38.87	36.57	36.15
	LibriSpeech	63.29	59.10	52.87	49.31	46.93	43.12	38.87	37.12	36.08	35.29
6-layer cltLSTM	WSJ	64.21	59.43	54.21	50.08	47.21	43.27	40.92	38.68	36.19	35.83
	LibriSpeech	63.82	58.39	52.47	48.89	46.39	42.27	38.23	36.89	35.91	35.07
FNN+2-layer LSTM	WSJ	69.77	65.18	62.21	58.63	54.21	50.18	49.51	47.21	45.38	42.39
	LibriSpeech	68.21	64.33	61.79	58.27	53.97	49.87	48.93	47.15	45.31	41.75
2-layer LSTM+FNN+ 2-layer LSTM	WSJ	69.31	65.39	62.89	58.71	54.39	50.48	49.75	47.53	45.42	42.61
	LibriSpeech	68.57	64.69	61.72	58.33	54.12	50.27	49.11	47.28	45.52	42.25
2-layer LSTM+FNN+ FNN	WSJ	69.15	64.98	62.21	58.17	53.97	49.37	48.83	46.93	45.21	41.87
	LibriSpeech	68.78	63.51	60.15	57.34	53.28	48.78	47.51	46.22	44.89	41.53
2-layer BLSTM	WSJ	65.93	59.71	54.68	50.39	47.58	44.12	41.28	39.21	36.83	36.39
	LibriSpeech	64.28	59.41	53.17	49.53	47.12	43.67	39.17	37.65	36.47	35.57
1-D CNN+2-layer BLSTM	WSJ	65.83	59.69	54.83	50.89	48.13	44.28	41.51	39.35	37.05	36.57
	LibriSpeech	63.91	58.98	53.39	49.88	47.78	44.29	39.28	38.12	36.58	36.02
1-D CNN+2-layer LSTM	WSJ	66.78	61.93	55.18	51.12	48.83	44.87	42.18	41.53	37.71	36.98
	LibriSpeech	64.89	59.65	54.12	50.87	47.79	45.17	40.54	38.93	37.62	36.58
1-D CNN+2-layer GRU	WSJ	66.65	61.83	55.28	52.87	48.93	45.91	42.31	41.87	38.12	37.85
	LibriSpeech	64.51	59.95	54.39	51.09	48.08	45.83	40.97	39.51	36.98	36.83
QuartzNet	WSJ	60.12	55.58	50.49	47.21	44.89	40.28	37.39	35.27	33.95	32.83
	LibriSpeech	58.83	54.39	49.87	46.95	43.18	39.28	36.33	34.87	32.28	31.98
Transformer	WSJ	58.39	54.37	49.31	46.98	42.57	39.83	36.41	34.12	32.74	31.29
	LibriSpeech	57.64	52.97	48.51	45.21	41.28	36.95	34.19	33.75	30.83	30.48
wav2vec 2.0	WSJ	51.75	47.28	45.35	43.65	40.81	37.63	35.49	33.71	30.28	28.23
	LibriSpeech	50.29	48.78	44.83	42.92	39.61	36.53	34.74	32.89	29.71	27.65
HuBERT	WSJ	51.78	47.20	45.29	43.60	40.72	37.60	33.45	31.65	30.21	28.18
	LibriSpeech	50.13	48.75	44.70	42.91	39.55	35.15	33.17	32.15	29.58	27.51
Domain Adaptive SSL	WSJ	51.43	47.13	44.89	43.23	40.52	37.39	33.27	31.21	29.81	27.48
	LibriSpeech	50.07	48.41	44.28	42.58	39.23	34.91	32.11	31.85	29.31	**27.11**

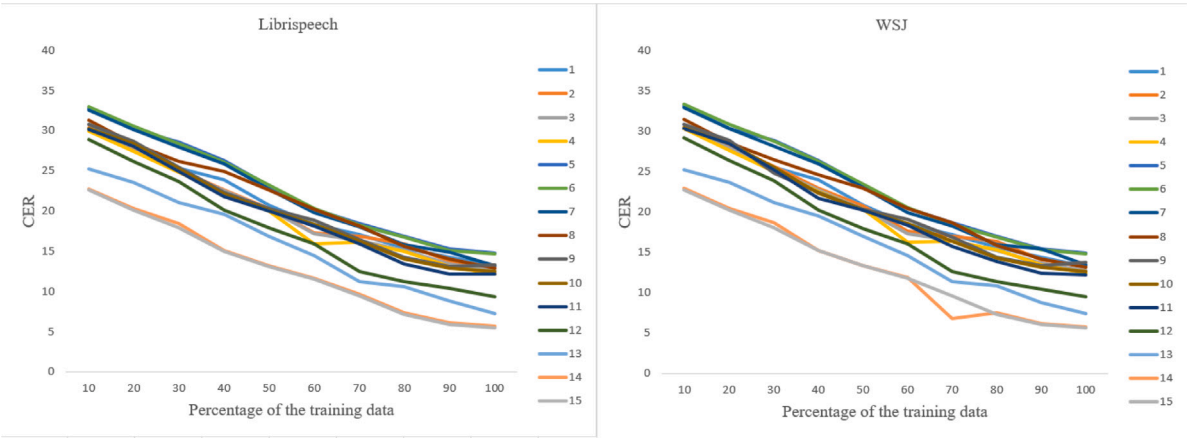

Fig. 2. CER in percentage for models trained with WSJ and Librispeech and tested with UASpeech. 1: 6-layer LSTM, 2: 6-layer ResLSTM, 3: 6-layer ltLSTM, 4: 6-layer cltLSTM, 5: FNN+2-layer LSTM, 6: 2-layer LSTM+FNN+2-layer LSTM, 7: 2-layer LSTM+FNN+FNN, 8: BLSTM, 9: CNN+2-layer BLSTM, 10: CNN+2-layer LSTM, 11:CNN+2-layer GRU, 12: Transformer, 13:QuartzNet, 14:wav2vec 2.0, 15:HuBERT.

to improve the performance of the ASR task. Therefore, we pre-trained the different models on Librispeech and WSJ datasets and then fine-tuned the models using LRE data (iCUBE or UASpeech) to explore these effects. Table 8 presents the results obtained from the pre-training of the different models on WSJ and Librispeech and fine-tuning on iCUBE. The 6-layer model outperforms all other numbers of layers for LSTM, ResLSTM, ltLSTM and cltLSTM. By applying the fine-tuning over LSTM, the WER achieved by the model improved by 0.13% on WSJ, using 10% of data. In 6-layer LSTM, increasing the pre-training data from 40% to 50% enhanced the WER to 9.06# but after fine-tuning WER improved to 9.11% on WSJ data. The 6-layer ResLSTM on Librispeech improved WER by 0.26% while enhancing the WER by 0.19% on Librispeech. The 8-layer ResLSTM achieved 3.91% WER improvement on Librispeech after fine-tuning with iCUBE data. The 2-layer ltLSTM got 1.29% improvement after receiving 10% of Librispeech dataset, while 6-layer ltLSTM model achieved near one percent improvement on WSJ. Furthermore, 8-layer cltLSTM can improve WER by 3.81% on Librispeech dataset.

All 2-layer LSTM and FNN configurations after fine-tuning got better WERs in different percentages of the train data on both datasets.

	10%				20%				30%				40%				50%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	44.68	44.89	45.23	45.58	42.57	42.69	43.39	43.72	40.18	40.32	41.18	42.38	37.61	37.45	37.62	38.69	35.18	34.89	34.21	36.11
ResLSTM	44.84	44.97	45.68	45.73	42.93	43.18	43.62	43.92	40.78	40.91	41.34	42.57	38.12	38.19	37.57	39.12	35.63	35.21	33.87	37.91
ltLSTM	44.73	44.93	45.39	45.69	42.68	42.51	43.53	43.83	39.91	39.87	40.91	42.21	37.22	37.15	37.29	38.52	34.81	34.51	33.62	35.87
cltLSTM	44.65	44.76	45.28	45.48	41.93	41.86	43.27	43.68	39.68	39.37	39.17	41.88	37.06	36.92	36.73	37.86	34.27	34.19	33.41	34.92
FNN+LSTM	50.39	-	-	-	49.13	-	-	-	45.48	-	-	-	42.61	-	-	-	40.13	-	-	-
LSTM+FNN+LSTM	50.73	-	-	-	50.21	-	-	-	46.73	-	-	-	43.67	-	-	-	41.76	-	-	-
LSTM+FNN+FNN	50.47	-	-	-	49.89	-	-	-	46.91	-	-	-	43.39	-	-	-	41.23	-	-	-
BLSTM	44.58	-	-	-	41.98	-	-	-	40.19	-	-	-	37.28	-	-	-	34.93	-	-	-
1-D CNN+BLSTM	45.12	-	-	-	42.39	-	-	-	40.51	-	-	-	37.59	-	-	-	35.42	-	-	-
1-D CNN+LSTM	48.17	-	-	-	46.21	-	-	-	43.69	-	-	-	41.58	-	-	-	39.27	-	-	-
1-D CNN+GRU	48.81	-	-	-	47.21	-	-	-	45.31	-	-	-	42.43	-	-	-	39.81	-	-	-
QuartzNet		44.11			41.53				38.32				35.79				31.72			
Transformer		42.19			40.87				37.92				35.32				31.49			

	60%				70%				80%				90%				100%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	34.12	33.21	32.78	34.28	32.48	32.12	30.67	33.65	29.75	28.73	27.49	31.19	28.43	27.13	25.31	30.27	27.29	26.72	24.17	29.23
ResLSTM	34.68	33.89	32.15	33.92	32.78	32.29	30.18	31.71	29.97	29.27	27.13	30.39	28.81	27.43	24.91	29.52	27.58	26.98	24.14	28.86
ltLSTM	33.47	32.95	31.87	33.17	31.62	31.47	28.73	31.22	29.12	28.47	26.42	29.98	28.07	26.81	24.68	28.75	27.18	25.13	24.08	28.14
cltLSTM	33.12	32.78	31.69	32.98	31.46	31.22	30.42	30.98	28.69	28.19	28.03	29.74	27.83	26.53	26.13	29.13	26.93	26.23	26.18	28.86
FNN+LSTM	38.79	-	-	-	35.41	-	-	-	33.29	-	-	-	31.63	-	-	-	29.34	-	-	-
LSTM+FNN+LSTM	40.08	-	-	-	38.19	-	-	-	35.83	-	-	-	33.12	-	-	-	30.39	-	-	-
LSTM+FNN+FNN	39.72	-	-	-	37.21	-	-	-	34.37	-	-	-	32.59	-	-	-	29.18	-	-	-
BLSTM	33.96	-	-	-	30.38	-	-	-	29.13	-	-	-	28.07	-	-	-	27.18	-	-	-
1-D CNN+BLSTM	34.21	-	-	-	30.82	-	-	-	29.89	-	-	-	28.69	-	-	-	27.63	-	-	-
1-D CNN+LSTM	36.21	-	-	-	34.87	-	-	-	33.47	-	-	-	30.19	-	-	-	28.78	-	-	-
1-D CNN+GRU	36.32	-	-	-	33.51	-	-	-	31.87	-	-	-	29.64	-	-	-	28.63	-	-	-
QuartzNet	29.93				27.98				25.65				23.51				22.13			
Transformer	29.67				27.43				25.29				23.19				21.27			

Fig. A.3. Pre-train different models on WSJ and test on iCUBE.

	10%				20%				30%				40%				50%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	44.81	45.31	45.79	45.93	42.37	42.83	43.22	44.87	40.59	40.19	40.38	44.21	37.17	36.91	36.28	38.16	34.12	33.86	32.61	36.43
ResLSTM	44.93	45.59	45.91	46.11	42.61	43.48	43.87	44.93	41.18	41.29	40.92	41.59	37.89	37.51	37.42	37.75	34.58	34.12	33.65	36.18
ltLSTM	44.87	45.43	45.83	45.97	42.21	42.62	43.51	44.93	40.13	39.87	39.41	41.89	36.82	36.47	35.94	37.79	33.67	33.27	32.35	36.55
cltLSTM	44.79	44.83	45.69	45.91	42.18	42.38	43.19	44.71	39.95	39.71	39.27	41.28	36.29	36.28	35.67	37.22	33.48	32.98	32.14	36.19
FNN+LSTM	51.38	-	-	-	49.71	-	-	-	46.13	-	-	-	43.19	-	-	-	41.58	-	-	-
LSTM+FNN+LSTM	51.69	-	-	-	50.37	-	-	-	47.62	-	-	-	44.51	-	-	-	41.85	-	-	-
LSTM+FNN+FNN	51.53	-	-	-	50.18	-	-	-	47.27	-	-	-	44.23	-	-	-	41.31	-	-	-
BLSTM	44.74	-	-	-	42.21	-	-	-	40.28	-	-	-	36.93	-	-	-	33.81	-	-	-
1-D CNN+BLSTM	44.92	-	-	-	42.53	-	-	-	40.63	-	-	-	37.61	-	-	-	34.28	-	-	-
1-D CNN+LSTM	47.38	-	-	-	45.61	-	-	-	43.12	-	-	-	41.17	-	-	-	38.62	-	-	-
1-D CNN+GRU	47.69	-	-	-	46.13	-	-	-	44.33	-	-	-	41.89	-	-	-	39.58	-	-	-
QuartzNet		44.27			42.18				38.85				35.39				31.98			
Transformer		43.78			41.28				38.31				34.97				31.52			

	60%				70%				80%				90%				100%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	31.63	31.29	31.48	34.62	30.22	29.81	29.12	32.51	28.67	27.69	27.86	30.71	27.43	26.11	25.53	28.17	26.13	25.81	24.28	27.92
ResLSTM	32.41	32.28	32.71	33.61	30.45	30.12	29.83	31.48	29.15	28.61	28.29	29.31	27.68	27.31	26.94	28.42	26.28	26.12	25.13	27.88
ltLSTM	31.28	30.92	30.52	34.61	29.72	29.67	28.97	33.83	28.43	27.41	27.18	31.22	27.19	25.29	25.09	29.48	26.11	24.91	24.11	27.63
cltLSTM	30.97	30.51	30.27	33.86	29.48	29.18	28.59	32.46	28.21	27.29	26.76	30.92	26.73	25.07	24.71	28.75	25.98	24.73	23.91	26.33
FNN+LSTM	39.75	-	-	-	37.13	-	-	-	35.81	-	-	-	34.69	-	-	-	33.97	-	-	-
LSTM+FNN+LSTM	40.19	-	-	-	38.61	-	-	-	37.61	-	-	-	35.82	-	-	-	34.88	-	-	-
LSTM+FNN+FNN	39.87	-	-	-	37.63	-	-	-	35.47	-	-	-	34.81	-	-	-	33.92	-	-	-
BLSTM	31.12	-	-	-	29.89	-	-	-	28.31	-	-	-	26.83	-	-	-	26.07	-	-	-
1-D CNN+BLSTM	31.57	-	-	-	30.31	-	-	-	28.91	-	-	-	27.12	-	-	-	26.19	-	-	-
1-D CNN+LSTM	35.58	-	-	-	33.62	-	-	-	31.92	-	-	-	29.71	-	-	-	27.22	-	-	-
1-D CNN+GRU	36.71	-	-	-	33.89	-	-	-	31.29	-	-	-	29.48	-	-	-	27.13	-	-	-
QuartzNet	30.11				28.11				25.98				23.89				22.85			
Transformer	29.92				27.71				25.61				23.46				22.32			

Fig. A.4. pre-train different models on Librispeech and test on iCUBE.

	10%				20%				30%				40%				50%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	44.77	45.27	45.73	45.88	42.31	42.78	43.17	44.81	40.53	40.13	40.31	44.17	37.13	36.86	36.21	38.12	34.06	33.79	32.55	36.35
ResLSTM	44.89	45.51	45.83	45.94	42.57	43.42	43.75	44.81	41.12	41.19	40.81	41.47	37.81	37.42	37.21	37.63	34.49	34.04	33.42	36.07
ltLSTM	44.81	45.37	45.76	45.89	42.18	42.56	43.46	44.83	40.07	39.76	39.34	41.78	36.73	36.38	35.83	37.71	33.59	33.18	32.27	36.47
cltLSTM	44.71	44.76	45.62	45.83	42.11	42.32	43.11	44.62	39.87	39.64	39.15	41.21	36.18	36.21	35.58	37.19	33.39	32.87	31.97	36.11
FNN+LSTM	51.29	-	-	-	49.62	-	-	-	45.97	-	-	-	43.08	-	-	-	41.47	-	-	-
LSTM+FNN+LSTM	51.61	-	-	-	50.28	-	-	-	47.53	-	-	-	44.42	-	-	-	41.73	-	-	-
LSTM+FNN+FNN	51.46	-	-	-	50.11	-	-	-	47.18	-	-	-	44.11	-	-	-	41.17	-	-	-
BLSTM	44.71	-	-	-	42.17	-	-	-	40.21	-	-	-	36.82	-	-	-	33.69	-	-	-
1-D CNN+BLSTM	44.87	-	-	-	42.48	-	-	-	40.53	-	-	-	37.49	-	-	-	34.17	-	-	-
1-D CNN+LSTM	47.31	-	-	-	45.55	-	-	-	43.05	-	-	-	41.08	-	-	-	38.51	-	-	-
1-D CNN+GRU	47.62	-	-	-	46.06	-	-	-	44.24	-	-	-	41.78	-	-	-	39.48	-	-	-
QuartzNet		44.06			42.01				38.79				35.3				31.91			
Transformer		43.69			41.15				38.17				34.77				31.44			

	60%				70%				80%				90%				100%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	31.56	31.21	31.41	34.51	30.17	29.75	28.95	32.42	28.61	27.62	27.74	30.63	27.38	26.03	25.44	28.04	26.08	25.73	24.17	27.81
ResLSTM	32.33	32.18	32.49	33.47	34.36	29.98	29.66	31.24	29.06	28.49	27.81	28.79	27.59	27.19	26.18	27.91	26.19	25.98	24.11	27.29
ltLSTM	31.19	30.87	30.38	34.48	29.61	29.59	28.83	33.71	28.31	27.32	27.03	31.12	27.11	25.17	24.93	29.36	26.03	24.71	23.87	27.32
cltLSTM	30.86	30.43	30.08	33.54	29.37	29.08	28.44	32.29	28.19	27.16	26.59	30.74	26.64	24.91	24.58	28.62	25.91	24.61	23.67	26.24
FNN+LSTM	39.58	-	-	-	36.88	-	-	-	35.69	-	-	-	34.57	-	-	-	33.84	-	-	-
LSTM+FNN+LSTM	40.08	-	-	-	38.49	-	-	-	37.48	-	-	-	35.71	-	-	-	33.91	-	-	-
LSTM+FNN+FNN	39.73	-	-	-	37.51	-	-	-	35.34	-	-	-	34.42	-	-	-	33.74	-	-	-
BLSTM	31.01	-	-	-	29.78	-	-	-	28.15	-	-	-	26.65	-	-	-	25.87	-	-	-
1-D CNN+BLSTM	31.46	-	-	-	30.23	-	-	-	28.83	-	-	-	26.98	-	-	-	26.08	-	-	-
1-D CNN+LSTM	35.47	-	-	-	33.52	-	-	-	31.81	-	-	-	29.57	-	-	-	27.13	-	-	-
1-D CNN+GRU	36.62	-	-	-	33.78	-	-	-	31.19	-	-	-	29.12	-	-	-	26.98	-	-	-
QuartzNet	30.04				28.03				25.91				23.81				22.78			
Transformer	29.81				27.57				25.38				23.21				22.01			

Fig. A.5. Pre-train different models on WSJ, Testing and Fine-tuning on iCUBE.

After fine-tuning with iCUBE data, LSTM after 2 layers of the FNN outperform the other similar structures. The 2-layer BLSTM model improved WER by 1.72% with Librispeech, which outperforms all the combination of the LSTMs and FNNs. By fine-tuning the models, which are a combination of the 1-D CNN with 2-layer BLSTM, LSTM, and GRU, all them got better WER compared with just pre-training. Transformer achieved a WER of 22.01% and 20.87% on WSJ and Librispeech datasets, respectively, which got 1.25% and 1.88% improvement after fine-tuning. In this scenario, Quartzet obtained 22.78% and 22.08% WERs on WSJ and Librispeech while wav2vec 2.0 achieved 20.21% on Librispeech. HuBERT model obtained 20.03% WER on Librispeech dataset which outperforms all models.

Similar trends can be seen when the models are fine-tuned with UASpeech data, as shown in Table 9.

	10%				20%				30%				40%				50%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	44.62	44.81	45.16	45.48	42.51	42.61	43.31	43.64	40.11	40.24	41.12	42.29	37.54	37.36	37.54	38.59	35.08	34.81	34.12	36.02
ResLSTM	44.78	44.89	45.59	45.65	42.86	43.11	43.54	43.82	40.71	40.82	41.26	42.48	37.98	38.11	37.48	38.99	35.52	35.09	33.75	37.79
ltLSTM	44.68	44.84	45.31	45.61	42.58	42.43	43.46	43.75	39.78	39.81	40.82	42.13	37.16	37.07	37.18	38.43	34.69	34.39	33.51	35.72
cltLSTM	44.57	44.68	45.18	45.41	41.84	41.74	43.21	43.62	39.57	39.26	39.11	41.79	36.96	36.81	36.62	37.76	34.17	34.09	32.33	34.79
FNN+LSTM	50.31	-	-	-	49.03	-	-	-	45.38	-	-	-	42.53	-	-	-	39.94	-	-	-
LSTM+FNN+LSTM	50.67	-	-	-	50.14	-	-	-	46.64	-	-	-	43.53	-	-	-	41.64	-	-	-
LSTM+FNN+FNN	50.39	-	-	-	49.78	-	-	-	46.79	-	-	-	43.27	-	-	-	41.12	-	-	-
BLSTM	44.51	-	-	-	41.87	-	-	-	40.04	-	-	-	37.07	-	-	-	34.75	-	-	-
1-D CNN+BLSTM	45.06	-	-	-	42.32	-	-	-	40.41	-	-	-	37.47	-	-	-	35.29	-	-	-
1-D CNN+LSTM	48.11	-	-	-	46.15	-	-	-	43.59	-	-	-	41.46	-	-	-	39.19	-	-	-
1-D CNN+GRU	48.75	-	-	-	47.17	-	-	-	45.22	-	-	-	42.31	-	-	-	39.73	-	-	-
QuartzNet		43.93			41.39				38.25				35.69				31.64			
Transformer		42.11			40.63				37.78				35.15				31.38			

	60%				70%				80%				90%				100%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	34.03	33.12	32.68	34.21	32.31	32.02	30.55	33.54	29.65	28.61	27.37	31.11	28.33	26.98	25.19	30.18	27.18	26.58	23.94	29.14
ResLSTM	34.58	33.78	31.98	33.81	32.69	32.17	30.05	31.59	29.86	29.08	26.97	30.22	28.68	27.31	24.79	29.41	27.42	26.75	23.98	27.73
ltLSTM	33.36	32.82	31.73	33.02	31.51	31.34	28.62	31.11	29.03	28.38	26.29	29.72	27.98	26.69	24.51	28.62	27.12	24.89	23.63	27.51
cltLSTM	32.98	32.66	31.54	32.83	31.32	31.14	30.29	30.81	28.54	28.06	27.89	29.65	27.71	26.39	25.49	28.88	26.83	26.15	25.83	27.76
FNN+LSTM	38.65	-	-	-	35.28	-	-	-	33.18	-	-	-	31.51	-	-	-	29.25	-	-	-
LSTM+FNN+LSTM	39.91	-	-	-	38.04	-	-	-	35.71	-	-	-	32.97	-	-	-	30.27	-	-	-
LSTM+FNN+FNN	39.61	-	-	-	37.12	-	-	-	34.26	-	-	-	32.48	-	-	-	28.89	-	-	-
BLSTM	33.67	-	-	-	30.23	-	-	-	28.91	-	-	-	27.87	-	-	-	26.71	-	-	-
1-D CNN+BLSTM	34.14	-	-	-	30.73	-	-	-	29.74	-	-	-	28.57	-	-	-	27.51	-	-	-
1-D CNN+LSTM	36.14	-	-	-	34.73	-	-	-	33.32	-	-	-	30.07	-	-	-	28.61	-	-	-
1-D CNN+GRU	36.19	-	-	-	33.38	-	-	-	31.68	-	-	-	29.49	-	-	-	28.52	-	-	-
QuartzNet	29.86				27.91				25.57				23.47				22.08			
Transformer	29.53				27.21				25.03				22.91				20.87			

Fig. A.6. Pre-train different models on Librispeech, Testing and Fine-tuning on iCUBE.

	10%				20%				30%				40%				50%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	45.83	45.62	45.31	45.49	39.73	39.42	39.07	39.28	35.49	35.21	34.72	34.98	30.21	30.08	29.48	29.73	26.72	26.38	25.97	26.19
ResLSTM	45.91	45.83	45.43	45.52	39.89	39.68	39.19	39.21	35.61	35.43	35.19	34.48	30.62	30.41	29.91	29.58	26.93	26.68	26.31	26.09
ltLSTM	45.71	45.32	45.12	45.2	39.43	39.28	38.89	39.17	35.21	34.83	34.51	34.72	30.07	29.91	29.27	29.48	26.42	26.01	25.62	25.89
cltLSTM	45.62	45.19	44.96	45.27	39.28	39.15	38.71	39.29	35.12	34.65	34.39	34.43	29.91	29.68	29.16	29.57	26.28	25.87	25.48	25.71
FNN+LSTM	47.21	-	-	-	41.83	-	-	-	34.28	-	-	-	30.42	-	-	-	27.53	-	-	-
LSTM+FNN+LSTM	47.93	-	-	-	42.87	-	-	-	36.21	-	-	-	32.49	-	-	-	29.78	-	-	-
LSTM+FNN+FNN	47.39	-	-	-	41.58	-	-	-	34.32	-	-	-	30.27	-	-	-	27.21	-	-	-
BLSTM	46.51	-	-	-	40.32	-	-	-	33.79	-	-	-	29.51	-	-	-	26.89	-	-	-
1-D CNN+BLSTM	46.39	-	-	-	40.12	-	-	-	32.87	-	-	-	28.32	-	-	-	25.69	-	-	-
1-D CNN+LSTM	45.83	-	-	-	39.98	-	-	-	32.48	-	-	-	27.89	-	-	-	25.21	-	-	-
1-D CNN+GRU	45.23	-	-	-	40.18	-	-	-	32.69	-	-	-	28.13	-	-	-	25.62	-	-	-
QuartzNet		43.79			38.51				32.13				28.04				24.93			
Transformer		42.15			37.62				31.35				27.17				23.69			

	60%				70%				80%				90%				100%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	23.47	23.26	22.89	23.08	22.68	22.41	21.93	22.13	20.93	20.69	20.28	20.51	19.61	19.28	18.79	18.93	18.53	18.27	18.06	18.13
ResLSTM	23.78	23.52	23.3	22.98	22.89	22.68	22.39	22.31	21.18	20.93	20.53	20.18	19.93	19.57	19.21	18.75	18.92	18.65	18.27	18.39
ltLSTM	23.26	22.78	22.43	22.61	22.51	21.62	21.32	21.49	20.62	20.13	20.02	20.12	19.47	18.89	18.53	18.72	18.17	18.21	17.91	18.13
cltLSTM	23.07	22.45	22.29	22.53	22.18	21.42	21.19	21.34	20.41	20.24	19.94	20.15	19.32	18.69	18.28	18.52	18.08	18.03	17.41	17.79
FNN+LSTM	25.19	-	-	-	23.61	-	-	-	21.49	-	-	-	20.32	-	-	-	19.18	-	-	-
LSTM+FNN+LSTM	27.52	-	-	-	25.18	-	-	-	22.83	-	-	-	21.59	-	-	-	19.97	-	-	-
LSTM+FNN+FNN	24.97	-	-	-	23.17	-	-	-	21.28	-	-	-	20.12	-	-	-	18.93	-	-	-
BLSTM	24.62	-	-	-	23.08	-	-	-	20.95	-	-	-	19.89	-	-	-	18.57	-	-	-
1-D CNN+BLSTM	24.17	-	-	-	22.77	-	-	-	20.51	-	-	-	19.32	-	-	-	18.05	-	-	-
1-D CNN+LSTM	23.72	-	-	-	22.13	-	-	-	20.18	-	-	-	18.91	-	-	-	17.65	-	-	-
1-D CNN+GRU	23.92	-	-	-	22.56	-	-	-	20.39	-	-	-	19.11	-	-	-	17.92	-	-	-
QuartzNet	21.53				18.39				17.71				16.39				15.19			
Transformer	21.18				18.72				16.53				15.28				14.21			

Fig. A.7. Train and Test different models on WSJ.

	10%				20%				30%				40%				50%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	41.81	41.07	40.67	41.28	38.73	37.87	37.08	37.61	31.58	30.62	30.18	30.48	28.41	27.85	27.32	27.72	25.79	25.03	24.89	25.38
ResLSTM	42.19	41.39	40.97	41.21	38.08	37.21	36.41	36.92	31.92	30.97	30.28	30.35	28.19	27.83	26.61	26.78	25.09	24.38	24.13	24.51
ltLSTM	41.97	40.53	40.17	40.37	36.65	35.71	34.89	35.32	31.47	30.53	30.03	30.28	27.27	26.62	25.97	26.53	24.61	23.91	23.51	23.47
cltLSTM	41.75	40.21	39.96	40.11	36.58	35.64	34.32	34.71	31.18	30.41	29.91	30.35	26.19	26.59	25.83	26.29	23.47	22.73	22.42	22.83
FNN+LSTM	45.63	-	-	-	40.68	-	-	-	33.78	-	-	-	29.81	-	-	-	26.41	-	-	-
LSTM+FNN+LSTM	46.27	-	-	-	41.29	-	-	-	34.79	-	-	-	31.19	-	-	-	26.83	-	-	-
LSTM+FNN+FNN	45.59	-	-	-	40.37	-	-	-	33.45	-	-	-	29.37	-	-	-	26.15	-	-	-
BLSTM	44.83	-	-	-	39.65	-	-	-	33.12	-	-	-	28.36	-	-	-	25.62	-	-	-
1-D CNN+BLSTM	44.65	-	-	-	39.58	-	-	-	32.92	-	-	-	28.17	-	-	-	25.43	-	-	-
1-D CNN+LSTM	44.53	-	-	-	39.47	-	-	-	32.17	-	-	-	28.07	-	-	-	25.28	-	-	-
1-D CNN+GRU	44.68	-	-	-	39.55	-	-	-	32.78	-	-	-	28.18	-	-	-	25.48	-	-	-
QuartzNet		41.38			35.21				30.82				26.62				23.95			
Transformer		39.17			34.72				29.48				25.75				22.49			

	60%				70%				80%				90%				100%			
	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8
LSTM	23.68	22.91	22.53	22.79	21.75	21.49	21.23	21.38	20.61	20.43	20.28	20.35	19.23	19.05	18.53	19.08	18.17	18.83	17.61	18.04
ResLSTM	22.89	22.17	21.98	22.28	21.97	21.21	20.78	21.93	20.89	20.45	20.23	20.37	19.48	19.32	18.73	19.98	18.49	18.18	17.51	18.01
ltLSTM	22.43	21.72	21.58	21.97	21.28	20.98	20.33	20.87	20.53	20.36	19.61	20.03	18.13	17.73	17.28	17.83	17.03	16.61	16.47	16.52
cltLSTM	21.28	20.68	20.52	20.61	20.51	20.18	19.71	20.02	19.18	19.03	18.63	18.95	17.93	17.39	16.97	17.21	16.39	15.97	15.61	15.78
FNN+LSTM	24.79	-	-	-	22.58	-	-	-	20.49	-	-	-	19.27	-	-	-	18.63	-	-	-
LSTM+FNN+LSTM	25.27	-	-	-	22.91	-	-	-	20.83	-	-	-	19.57	-	-	-	18.87	-	-	-
LSTM+FNN+FNN	24.45	-	-	-	22.31	-	-	-	20.29	-	-	-	19.13	-	-	-	18.42	-	-	-
BLSTM	23.17	-	-	-	21.78	-	-	-	19.83	-	-	-	18.86	-	-	-	18.21	-	-	-
1-D CNN+BLSTM	23.58	-	-	-	21.48	-	-	-	19.49	-	-	-	18.73	-	-	-	17.98	-	-	-
1-D CNN+LSTM	23.36	-	-	-	21.31	-	-	-	19.27	-	-	-	18.61	-	-	-	17.83	-	-	-
1-D CNN+GRU	23.68	-	-	-	21.53	-	-	-	19.39	-	-	-	18.87	-	-	-	18.08	-	-	-
QuartzNet	21.83				19.61				17.38				15.51				14.38			
Transformer	20.23				17.83				16.19				14.69				13.73			

Fig. A.8. Train and Test different models on LibriSpeech.

8 Discussion

This paper aims to assess the impact of data size on the pre-training and fine-tuning of ASR models in low-resource environments. The amount of data available during the pre-training and fine-tuning phases is a critical factor for improving ASR performance in such settings. By analyzing the results presented in Table 7, we can conclude that increasing the amount of related training data has a significant positive effect on the performance of the ASR system. Since the training and test data are from the same domain, each incremental increase in the percentage of training data led to a corresponding improvement in WER. These substantial improvements highlight a strong correlation between the WER and the availability of domain-related data for training the ASR system.

Table 7
Best results for each model when trained and tested on WSJ and Librispeech datasets.

		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
6-layer LSTM	WSJ	45.31	39.07	34.72	29.48	25.97	22.98	21.93	20.28	18.79	18.06
	LibriSpeech	40.67	37.08	30.18	27.32	24.89	22.53	21.23	20.28	18.53	17.61
6-layer ResLSTM	WSJ	45.43	39.19	35.19	29.91	26.31	23.30	22.39	20.53	19.21	18.27
	LibriSpeech	40.97	36.41	30.28	26.61	24.13	21.98	20.78	20.23	18.73	17.51
6-layer ltLSTM	WSJ	45.12	38.89	34.51	29.27	25.62	22.43	21.32	20.02	18.53	17.91
	LibriSpeech	40.17	34.89	30.03	35.97	23.51	21.58	20.33	19.61	17.28	16.47
6-layer cltLSTM	WSJ	44.96	38.71	34.39	29.16	25.48	22.29	21.19	19.94	18.28	17.41
	LibriSpeech	39.96	34.32	29.91	25.83	22.42	20.52	19.71	18.63	16.97	15.61
FNN+2-layer LSTM	WSJ	47.21	41.83	34.28	30.42	27.53	25.19	23.61	21.49	20.32	19.18
	LibriSpeech	45.63	40.68	33.78	29.81	26.41	24.79	22.58	20.49	19.27	18.63
2-layer LSTM+FNN+ 2-layer LSTM	WSJ	47.93	42.87	36.21	32.49	29.78	27.52	25.18	22.83	21.59	19.97
	LibriSpeech	46.27	41.29	34.79	31.19	26.83	25.27	22.91	20.81	19.57	18.87
2-layer LSTM+FNN+ FNN	WSJ	47.39	41.58	34.32	30.27	27.21	24.97	23.17	21.28	20.12	18.93
	LibriSpeech	45.59	40.37	33.45	29.37	26.15	24.45	22.31	20.29	19.13	18.42
2-layer BLSTM	WSJ	46.51	40.32	33.79	29.51	26.89	24.62	23.08	20.95	19.89	18.57
	LibriSpeech	44.83	39.65	33.12	28.36	25.62	23.71	21.78	19.83	18.86	18.21
1-D CNN+2-layer BLSTM	WSJ	46.39	40.12	32.87	28.32	25.69	24.17	22.77	20.51	19.32	18.05
	LibriSpeech	44.65	39.58	32.92	28.17	25.43	23.58	21.48	19.49	18.73	17.98
1-D CNN+2-layer LSTM	WSJ	45.83	39.98	32.48	27.89	25.21	23.72	22.13	20.18	18.91	17.65
	LibriSpeech	44.53	39.47	32.71	28.07	25.28	23.36	21.31	19.27	18.61	17.83
1-D CNN+2-layer GRU	WSJ	46.23	40.18	32.69	28.13	25.62	23.92	22.56	20.39	19.11	17.92
	LibriSpeech	44.68	39.55	32.78	28.18	25.48	23.68	21.53	19.39	18.87	18.08
QuartzNet	WSJ	43.79	38.51	32.13	28.04	24.93	21.53	18.39	17.71	16.39	15.19
	LibriSpeech	41.38	35.21	30.82	26.62	23.95	21.83	19.61	17.38	15.51	14.38
Transformer	WSJ	42.15	37.62	31.35	27.17	23.69	21.18	18.72	16.53	15.28	14.21
	LibriSpeech	39.17	34.72	29.48	25.75	22.49	20.23	17.83	16.19	14.69	13.73
wav2vec 2.0	WSJ	29.81	26.65	22.93	19.23	17.65	14.29	12.55	10.48	8.93	7.78
	LibriSpeech	27.39	23.38	21.58	18.78	14.28	13.92	11.39	9.31	8.75	6.21
HuBERT	WSJ	29.75	25.31	22.73	19.10	17.21	13.91	12.15	10.13	8.70	7.53
	LibriSpeech	27.19	23.18	21.27	18.35	16.13	13.51	11.12	9.08	8.49	**6.13**

However, by analyzing the results in Tables 5 and 6, obtained by testing different models on the iCUBE and UASpeech datasets, only a slight improvement was observed with each increase in the percentage of training data across all models. These findings further support the assertion that data from an unrelated domain is insufficient to substantially enhance the performance of the ASR system. Consequently, even a significant increase in the amount of training data from a different domain does not lead to a notable improvement in ASR system performance.

Increasing the amount of pre-training data significantly improves the performance of stacked LSTM models across different layer configurations. When the training data on WSJ increased from 10% to 100%, the performance of the 2-layer LSTM model improved by 41.68%, while the improvement for the LibriSpeech dataset was 38.92%. The 4-, 6-, and 8-layer LSTM models showed WER improvements of 43.03%, 46.97%, and 39.21% on the WSJ dataset, respectively. Similarly, for the same models on the LibriSpeech dataset, the performance increased by 40.47%, 46.56% and 35.87%, respectively. Although increasing the amount of data from 10% to 100% resulted in significant accuracy improvements, adding more layers to the models led to a degradation in WER. The ResLSTM, ltLSTM, and cltLSTM models demonstrated consistent improvements as the amount of training data increased to 100% for both the WSJ and LibriSpeech datasets. Notably, the cltLSTM achieved the best WER results on both datasets, likely due to the use of future context frames, which provide more valuable information for the model. Overall, stacked LSTM models perform better with increasing amounts of training data at each percentage level, highlighting their effectiveness when sufficient relevant data is available for training ASR systems.

All stacked LSTM models achieved better WERs after fine-tuning on low-resource data. The results clearly indicate that domain-specific data plays a crucial role in training models for particular ASR tasks. The 6-layer ResLSTM model showed the greatest improvement, with a 4.05% reduction in WER when fine-tuned on iCUBE for the WSJ dataset, while the ltLSTM and cltLSTM models achieved close to a 1% improvement. The 6-layer cltLSTM model demonstrated a greater WER improvement when fine-tuned on iCUBE for LibriSpeech, with a 1.33% reduction. These findings underscore the strong positive correlation between the amount of domain-specific data and the performance of ASR systems in low-resource environments. Another important conclusion is that pre-training on high-resource data followed by fine-tuning on relevant domain data is the most effective approach for handling low-resource settings.

The results of this study indicate a positive correlation between dataset size and model structure. FNN-LSTM is a model where the FNN component aids in detecting factors of variation in the inputs, allowing the LSTM to effectively learn temporal correlations (Chien and Misbullah, 2016). The most notable finding is that such functionality is only effective in high-resource data environments, and this topology does not improve the WER of ASR systems in low-resource settings. Similar architectures, which combine LSTMs and FNNs, exhibited comparable performance in such environments. As the amount of pre-training data increased, the WER steadily improved across these architectures. Furthermore, the inclusion of domain-related data led to a significant reduction in WER.

The Transformer model achieved WER improvements of 40.01% on WSJ and 49.58% on LibriSpeech. At each step, increasing the amount of data led to further WER improvements, highlighting that neural networks are highly data-dependent. After fine-tuning the Transformer, WER increased slightly by 1.25% on WSJ and 1.88% on LibriSpeech. The most notable finding from the fine-tuning results is that pre-training followed by fine-tuning on domain-specific data can significantly enhance ASR performance within that specific domain.

Table 8
Best results for each model when pre-training on WSJ and Librispeech, and Fine-Tuning and test on iCUBE.

		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
6-layer LSTM	WSJ	45.73	43.17	40.31	36.21	32.55	31.41	28.95	27.74	25.44	24.17
	LibriSpeech	45.16	43.31	41.12	37.54	34.12	32.68	30.55	27.37	25.19	23.94
6-layer ResLSTM	WSJ	45.83	43.75	40.81	37.21	33.42	32.49	29.66	27.81	26.18	24.11
	LibriSpeech	45.59	43.54	41.26	37.48	33.75	31.98	30.05	26.97	24.79	23.98
6-layer ltLSTM	WSJ	45.76	43.46	39.34	35.83	32.27	30.38	28.83	27.03	24.93	23.87
	LibriSpeech	45.31	43.46	40.82	37.18	33.51	31.73	28.62	26.29	24.51	23.63
6-layer cltLSTM	WSJ	45.62	43.11	39.15	35.58	31.97	30.08	28.44	26.59	24.58	23.67
	LibriSpeech	45.18	43.21	39.11	36.62	33.23	31.54	30.29	27.89	25.94	25.83
FNN+2-layer LSTM	WSJ	51.29	49.62	45.97	43.08	41.47	39.58	36.88	35.69	34.57	33.84
	LibriSpeech	50.31	49.03	45.38	42.53	39.94	38.65	35.28	33.18	31.51	29.25
2-layer LSTM+FNN+ 2-layer LSTM	WSJ	51.61	50.28	47.53	44.42	41.73	40.08	38.49	37.48	35.71	33.91
	LibriSpeech	50.67	50.14	46.64	43.53	41.64	39.91	38.04	35.71	32.97	30.27
2-layer LSTM+FNN+ FNN	WSJ	51.46	50.11	47.18	44.11	41.17	39.73	37.51	35.34	34.42	33.74
	LibriSpeech	50.39	49.78	46.79	43.27	41.12	39.61	37.12	34.26	32.48	28.89
2-layer BLSTM	WSJ	44.71	42.17	40.21	36.82	33.69	31.01	29.78	28.15	26.65	25.87
	LibriSpeech	44.51	41.87	40.04	37.07	34.75	33.67	30.23	28.91	27.87	26.71
1-D CNN+2-layer BLSTM	WSJ	44.87	42.48	40.53	37.49	34.17	31.46	30.23	28.83	26.98	26.08
	LibriSpeech	45.06	42.32	40.41	37.47	35.29	34.14	30.73	29.74	28.57	27.51
1-D CNN+2-layer LSTM	WSJ	47.31	45.55	43.05	41.08	38.51	35.47	33.52	31.81	29.57	27.13
	LibriSpeech	48.11	46.15	43.59	41.46	39.19	36.14	34.73	33.32	30.07	28.61
1-D CNN+2-layer GRU	WSJ	47.62	46.06	44.24	41.78	39.48	36.62	33.78	31.19	29.12	26.98
	LibriSpeech	48.75	47.17	45.22	42.31	39.73	36.19	33.38	31.68	29.49	28.52
QuartzNet	WSJ	44.06	42.01	38.79	35.30	31.91	30.04	28.03	25.91	23.81	22.78
	LibriSpeech	43.93	41.39	38.25	35.69	31.64	29.86	27.91	25.57	23.47	22.08
Transformer	WSJ	43.69	41.15	38.17	34.77	31.44	29.81	27.57	25.38	23.21	22.01
	LibriSpeech	42.11	40.63	37.78	35.15	31.38	29.53	27.21	25.03	22.91	20.87
wav2vec 2.0	WSJ	38.03	36.71	33.98	31.03	29.02	27.21	25.68	24.71	22.50	21.48
	LibriSpeech	36.71	34.99	33.71	30.19	28.42	26.03	23.97	23.12	21.07	20.21
HuBERT	WSJ	38.30	36.75	33.92	30.89	28.89	26.91	25.30	24.49	22.42	21.32
	LibriSpeech	36.81	34.91	33.39	30.01	28.05	25.78	23.88	22.91	21.17	20.03
Domain Adaptive SSL	WSJ	38.16	36.53	33.77	30.48	28.71	26.78	25.11	24.23	22.11	21.14
	LibriSpeech	36.62	34.77	33.37	29.91	28.01	25.71	23.78	22.86	21.07	**20.01**

9 Conclusion

In this paper, we addressed the challenge of training and evaluating state-of-the-art ASR methods in high-resource environments and applying them to low-resource settings. Low-resource environments refer to scenarios where insufficient training data is available, such as speech recognition for children, speakers with speech disorders, or those with accents. We highlighted the issue of limited training data as a major challenge for state-of-the-art ASR models in low-resource environments. To investigate this, we compared various ASR system configurations to understand the effect of training data size on model performance in such settings. Our results emphasize that these models require a substantial amount of relevant data to achieve superior performance in specific tasks. For example, the WER for 10% of the WSJ dataset is 45.79%, while for LibriSpeech, it is 45.23. When the data volume increased from 10% to 20%, the WERs dropped to 5.61% for WSJ and 4.06% for LibriSpeech. A significant WER improvement of 10.15% occurred when the data size increased from 30% to 40% for WSJ, with an 8.64% improvement for LibriSpeech. Moreover, we demonstrated that training and testing on data from the same domain yielded better results than testing with data from a different domain. This highlights that the relevance of training data to the ASR task's domain significantly impacts WER performance. One of the key findings of this study is that pre-training on high-resource data followed by fine-tuning on domain-relevant data in low-resource ASR tasks produces the best results.

CRediT authorship contribution statement

Kavan Fatehi: Writing – review & editing, Writing – original draft, Validation, Formal analysis, Data curation, Conceptualization. **Mercedes Torres Torres:** Writing – review & editing, Supervision. **Ayse Kucukyilmaz:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Kavan Fatehi reports was provided by University of Nottingham.

Acknowledgments

This work is supported by UKRI Trustworthy Autonomous Systems Hub (EP/V00784X/1).

Appendix. Complete results

Figs. A.3 and A.4 show the results obtained from the pre-training different models based on the different number of layers and percentage of the data on WSJ and Librispeech datasets, respectively. Furthermore, the results of the pre-training and fine-tuning on iCUBE data are shown in Fig. A.5 for WSJ and Fig. A.6 for Librispeech. In addition, the full results after training and testing on WSJ and Librispeech are presented in Figs. A.7 and A.8, respectively.

Data availability

The authors do not have permission to share data.

Table 9
Best results for each model when pre-training on WSJ and Librispeech, and Fine-Tuning and Testing on UASpeech.

		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
6-layer LSTM	WSJ	65.39	59.70	54.21	49.83	47.21	44.30	40.93	39.11	37.01	36.15
	LibriSpeech	63.21	58.50	51.97	49.21	47.15	43.35	39.61	37.62	36.11	35.73
6-layer ResLSTM	WSJ	63.97	59.73	56.21	52.13	47.93	43.17	42.61	40.07	38.23	37.97
	LibriSpeech	63.38	59.17	55.08	51.13	48.33	42.27	41.65	38.92	38.65	37.01
6-layer ltLSTM	WSJ	64.60	58.87	53.98	50.01	46.87	43.61	40.68	38.57	36.29	35.91
	LibriSpeech	63.06	58.81	52.57	49.10	46.61	42.91	38.58	36.98	35.88	35.01
6-layer cltLSTM	WSJ	63.98	59.17	53.93	49.81	46.93	43.01	40.63	38.39	35.97	35.61
	LibriSpeech	63.51	58.11	52.21	48.59	46.11	42.03	37.93	36.57	35.69	34.83
FNN+2-layer LSTM	WSJ	69.28	64.87	62.17	58.41	53.91	49.89	49.23	46.98	45.09	42.04
	LibriSpeech	68.02	64.07	61.48	58.03	53.62	49.68	48.69	46.83	45.17	41.31
2-layer LSTM+FNN+ 2-layer LSTM	WSJ	69.05	65.07	62.51	58.32	54.12	50.28	49.33	47.21	45.19	42.33
	LibriSpeech	68.28	64.31	61.48	58.08	53.83	50.03	48.33	47.05	45.31	41.92
2-layer LSTM+FNN+ FNN	WSJ	68.87	64.52	61.97	57.83	53.62	49.08	48.61	46.62	45.02	41.55
	LibriSpeech	68.33	63.28	59.93	56.91	53.01	48.52	47.21	65.93	44.62	41.18
2-layer BLSTM	WSJ	65.71	59.38	54.37	50.11	47.21	43.79	40.98	38.85	36.57	36.09
	LibriSpeech	64.03	59.20	52.88	49.27	46.82	43.42	38.33	37.49	36.15	35.17
1-D CNN+2-layer BLSTM	WSJ	65.53	59.31	54.57	50.51	47.83	43.98	41.17	39.04	36.83	36.21
	LibriSpeech	63.75	58.62	53.11	49.47	47.45	44.08	39.05	37.79	36.17	35.78
1-D CNN+2-layer LSTM	WSJ	66.43	61.65	54.93	50.83	48.52	44.51	41.89	41.21	37.31	36.67
	LibriSpeech	64.57	59.36	53.71	50.51	47.31	46.83	40.17	38.65	37.28	36.23
1-D CNN+2-layer GRU	WSJ	66.28	61.57	55.03	52.46	48.69	46.62	42.05	41.51	37.73	37.51
	LibriSpeech	64.31	59.68	54.11	50.83	47.79	45.49	40.63	39.13	36.61	36.41
QuartzNet	WSJ	59.81	55.29	50.17	46.93	44.53	39.97	37.02	34.92	33.61	32.51
	LibriSpeech	58.62	54.07	49.51	46.63	42.78	38.88	36.02	34.57	31.89	31.63
Transformer	WSJ	58.07	54.08	48.97	46.61	42.31	39.51	36.11	33.78	32.51	30.97
	LibriSpeech	57.31	52.67	48.28	44.83	40.93	36.64	33.83	33.41	30.51	30.13
wav2vec 2.0	WSJ	51.63	47.13	45.17	43.48	40.65	37.49	35.20	33.58	30.15	28.03
	LibriSpeech	50.11	48.51	44.67	42.71	39.45	36.21	34.45	32.60	29.51	27.39
HuBERT	WSJ	51.61	47.03	45.09	43.39	40.51	36.39	33.28	31.39	30.12	28.05
	LibriSpeech	49.98	48.61	44.58	42.69	39.28	35.01	32.98	32.51	29.31	27.32
Domain Adaptive SSL	WSJ	51.59	47.01	44.99	43.34	40.39	36.21	33.19	31.27	30.02	28.01
	LibriSpeech	49.91	48.59	44.49	42.58	39.23	34.91	32.91	32.47	29.25	**27.21**

References

Al-Rfou, R., Choe, D., Constant, N., Guo, M., Jones, L., 2019. Character-level language modeling with deeper self-attention. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3159–3166.

Alam, M., Samad, M.D., Vidyaratne, L., Glandon, A., Iftekharuddin, K.M., 2020. Survey on deep neural networks in speech and vision systems. Neurocomputing 417, 302–321.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., Zhu, Z., 2016a. Deep speech 2 : End-to-end speech recognition in english and mandarin. In: Balcan, M.F., Weinberger, K.Q. (Eds.), Proceedings of the 33rd International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 48, PMLR, New York, New York, USA, pp. 173–182, URL https://proceedings.mlr.press/v48/amodei16.html.

Amodei, D., Anubhai, R., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Chen, J., Chrzanowski, M., Coates, A., Diamos, G., et al., 2016b. End to end speech recognition in english and mandarin.

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F.M., Weber, G., 2019. Common voice: A massively-multilingual speech corpus. In: International Conference on Language Resources and Evaluation.

Atal, B.S., Hanauer, S.L., 1971. Speech analysis and synthesis by linear prediction of the speech wave. J. Acoust. Soc. Am. 50 (2B), 637–655.

Audhkhasi, K., Ramabhadran, B., Saon, G., Picheny, M., Nahamoo, D., 2017. Direct acoustics-to-word models for english conversational speech recognition. In: Proc. Interspeech 2017. pp. 959–963. http://dx.doi.org/10.21437/Interspeech.2017-546.

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: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (Eds.), Proceedings of the 39th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 162, PMLR, pp. 1298–1312, URL https://proceedings.mlr.press/v162/baevski22a.html.

Baevski, A., Mohamed, A., 2020. Effectiveness of self-supervised pre-training for asr. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7694–7698.

Baevski, A., Zhou, Y., Mohamed, A., Auli, M., 2020. Wav2vec 2.0: A framework for self-supervised learning of speech representations. In: Advances in Neural Information Processing Systems, vol. 33, pp. 12449–12460.

Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y., 2016. End-to-end attention-based large vocabulary speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4945–4949.

Bahl, L.R., Jelinek, F., Mercer, R.L., 1983. A maximum likelihood approach to continuous speech recognition. IEEE Trans. Pattern Anal. Mach. Intell. (2), 179–190.

Baker, J.K., 1975a. Stochastic Modeling as a Means of Automatic Speech Recognition. Carnegie Mellon University.

Baker, J., 1975b. The DRAGON system–An overview. IEEE Trans. Acoust. Speech Signal Process. 23 (1), 24–29.

Battenberg, E., Chen, J., Child, R., Coates, A., Li, Y.G.Y., Liu, H., Satheesh, S., Sriram, A., Zhu, Z., 2017. Exploring neural transducers for end-to-end speech recognition. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop. ASRU, IEEE, pp. 206–213.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1798–1828.

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

Cai, M., Liu, J., 2016. Maxout neurons for deep convolutional and LSTM neural networks in speech recognition. Speech Commun. 77, 53–64.

Celikyilmaz, A., Deng, L., Hakkani-Tür, D., 2018. Deep learning in spoken and text-based dialog systems. In: Deep Learning in Natural Language Processing. Springer, pp. 49–78.

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. ICASSP, IEEE, pp. 4960–4964.

Chan, W., Lane, I., 2015. Deep convolutional neural networks for acoustic modeling in low resource languages. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 2056–2060.

Chen, K., Huo, Q., 2016. Training deep bidirectional LSTM acoustic model for LVCSR by a context-sensitive-chunk BPTT approach. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (7), 1185–1193.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al., 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE J. Sel. Top. Sign. Proces. 16 (6), 1505–1518.

Chen, X., Zhang, S., Song, D., Ouyang, P., Yin, S., 2020. Transformer with bidirectional decoder for speech recognition. In: Proc. Interspeech 2020. pp. 1773–1777. http://dx.doi.org/10.21437/Interspeech.2020-2677.

Chen, Y., Zhang, H., Yang, X., Zhang, W., Qu, D., 2024. Meta-adaptable-adapter: Efficient adaptation of self-supervised models for low-resource speech recognition. Neurocomputing 609, 128493. http://dx.doi.org/10.1016/j.neucom.2024.128493, URL https://www.sciencedirect.com/science/article/pii/S0925231224012645.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al., 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. 2, (3), p. 6, See https://vicuna.lmsys.org. (Accessed 14 April 2023).

Chien, J.-T., Misbullah, A., 2016. Deep long short-term memory networks for speech recognition. In: 2016 10th International Symposium on Chinese Spoken Language Processing. ISCSLP, IEEE, pp. 1–5.

Chiu, C.-C., Qin, J., Zhang, Y., Yu, J., Wu, Y., 2022. Self-supervised learning with random-projection quantizer for speech recognition. In: International Conference on Machine Learning. PMLR, pp. 3915–3924.

Chiu*, C.-C., Raffel*, C., 2018. Monotonic chunkwise attention. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=Hko85plCW.

Cho, J., Baskar, M.K., Li, R., Wiesner, M., Mallidi, S.H., Yalta, N., Karafiat, M., Watanabe, S., Hori, T., 2018. Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling. In: 2018 IEEE Spoken Language Technology Workshop. SLT, IEEE, pp. 521–527.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1251–1258.

Chorowski, J., Bahdanau, D., Cho, K., Bengio, Y., 2014a. End-to-end continuous speech recognition using attention-based recurrent nn: First results. In: NIPS 2014 Workshop on Deep Learning, December 2014.

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

Chorowski, J., Weiss, R.J., Bengio, S., van den Oord, A., 2019. Unsupervised speech representation learning using wavenet autoencoders. IEEE/ACM Trans. Audio Speech Lang. Process. 27 (12), 2041–2053.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al., 2023. Palm: Scaling language modeling with pathways. J. Mach. Learn. Res. 24 (240), 1–113.

Chuangsuwanich, E., 2016. Multilingual techniques for low resource automatic speech recognition. Technical Report, Massachusetts Institute of Technology Cambridge United States.

Chung, Y.-A., Glass, J., 2020. Improved speech representations with multi-target autoregressive predictive coding. arXiv preprint arXiv:2004.05274.

Chung, J.S., Nagrani, A., Zisserman, A., 2018. VoxCeleb2: Deep speaker recognition. In: Proc. Interspeech 2018. pp. 1086–1090. http://dx.doi.org/10.21437/Interspeech.2018-1929.

Chung, Y.-A., Tang, H., Glass, J., 2020. Vector-quantized autoregressive predictive coding. In: Proc. Interspeech 2020. pp. 3760–3764. http://dx.doi.org/10.21437/Interspeech.2020-1228.

Chung, Y.-A., Zhang, Y., Han, W., Chiu, C.-C., Qin, J., Pang, R., Wu, Y., 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop. ASRU, IEEE, pp. 244–250.

Cieri, C., Miller, D., Walker, K., 2004. The Fisher corpus: a resource for the next generations of speech-to-text. In: LREC, vol. 4, pp. 69–71.

Collobert, R., Puhrsch, C., Synnaeve, G., 2016. Wav2letter: An end-to-end convnet-based speech recognition system. arXiv preprint arXiv:1609.03193.

Dahl, G.E., Yu, D., Deng, L., Acero, A., 2011. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. 20 (1), 30–42.

Davis, K.H., Biddulph, R., Balashek, S., 1952. Automatic recognition of spoken digits. J. Acoust. Soc. Am. 24 (6), 637–642.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A.P., Caron, M., Geirhos, R., Alabdulmohsin, I., et al., 2023. Scaling vision transformers to 22 billion parameters. In: International Conference on Machine Learning. PMLR, pp. 7480–7512.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

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.

Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics, pp. 4171–4186. http://dx.doi.org/10.18653/v1/n19-1423.

Do, C.-T., Doddipatla, R., Hain, T., 2021. Multiple-hypothesis CTC-based semi-supervised adaptation of end-to-end speech recognition. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6978–6982.

Do, C.-T., Doddipatla, R., Li, M., Hain, T., 2023. Domain adaptive self-supervised training of automatic speech recognition. In: Proc. INTERSPEECH 2023. pp. 4389–4393. http://dx.doi.org/10.21437/Interspeech.2023-1091.

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. ICASSP, IEEE, pp. 5884–5888.

Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al., 2023. Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378.

Dudley, H., Balashek, S., 1958. Automatic recognition of phonetic patterns in speech. J. Acoust. Soc. Am. 30 (8), 721–732.

Durrett, G., Pauls, A., Klein, D., 2012. Syntactic transfer using a bilingual lexicon. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1–11.

Eyben, F., Wöllmer, M., Schuller, B., Graves, A., 2009. From speech to letters-using a novel neural network architecture for grapheme based asr. In: 2009 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, pp. 376–380.

Fantaye, T.G., Yu, J., Hailu, T.T., 2020. Advanced convolutional neural network-based hybrid acoustic models for low-resource speech recognition. Computers 9 (2), 36.

Fatehi, K., Kucukyilmaz, A., 2023. LABERT: A combination of local aggregation and self-supervised speech representation learning for detecting informative hidden units in low-resource ASR systems. In: INTERSPEECH 2023. pp. 211–215. http://dx.doi.org/10.21437/Interspeech.2023-2001.

Fatehi, K., Torres Torres, M., Kucukyilmaz, A., 2022. ScoutWav: Two-step fine-tuning on self-supervised automatic speech recognition for low-resource environments. In: Proc. Interspeech 2022. pp. 3523–3527. http://dx.doi.org/10.21437/Interspeech.2022-10270.

Feng, S., Lee, T., 2018. Improving cross-lingual knowledge transferability using multilingual TDNN-BLSTM with language-dependent pre-final layer. In: INTERSPEECH. pp. 2439–2443.

Forgie, J.W., Forgie, C.D., 1959. Results obtained from a vowel recognition computer program. J. Acoust. Soc. Am. 31 (11), 1480–1489.

Frinken, V., Zamora-Martinez, F., Espana-Boquera, S., Castro-Bleda, M.J., Fischer, A., Bunke, H., 2012. Long-short term memory neural networks language modeling for handwriting recognition. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). IEEE, pp. 701–704.

Gales, M.J., 1998. Maximum likelihood linear transformations for HMM-based speech recognition. Comput. Speech Lang. 12 (2), 75–98.

Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., 1993. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon technical report n 93, p. 27403.

Ghahremani, P., Manohar, V., Hadian, H., Povey, D., Khudanpur, S., 2017. Investigation of transfer learning for ASR using LF-MMI trained neural networks. In: 2017 IEEE Automatic Speech Recognition and Understanding Workshop. ASRU, IEEE, pp. 279–286.

Godfrey, J.J., Holliman, E.C., McDaniel, J., 1992. Switchboard: Telephone speech corpus for research and development. In: Acoustics, Speech, and Signal Processing, IEEE International Conference on, vol. 1, IEEE Computer Society, pp. 517–520.

Gong, Y., Luo, H., Liu, A.H., Karlinsky, L., Glass, J., 2023. Listen, think, and understand. arXiv preprint arXiv:2305.10790.

Goodfelow, I., Bengio, Y., Courville, A., 2016. Deep Learning (Adaptive Computation and Machine Learning Series). e MIT Press, Cambridge, England.

Goodman, J.T., 2001. A bit of progress in language modeling. Comput. Speech Lang. 15 (4), 403–434.

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

Graves, A., Fernández, S., Gomez, F., Schmidhuber, J., 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376.

Graves, A., Fernández, S., Schmidhuber, J., 2005. Bidirectional LSTM networks for improved phoneme classification and recognition. In: International Conference on Artificial Neural Networks. Springer, pp. 799–804.

Graves, A., Jaitly, N., 2014. Towards end-to-end speech recognition with recurrent neural networks. In: International Conference on Machine Learning. pp. 1764–1772.

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. IEEE, pp. 6645–6649.

Graves, A., Schmidhuber, J., 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. 18 (5–6), 602–610.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., Pang, R., 2020. Conformer: Convolution-augmented transformer for speech recognition. In: Proc. Interspeech 2020. pp. 5036–5040. http://dx.doi.org/10.21437/Interspeech.2020-3015.

Guo, J., Tiwari, G., Droppo, J., Segbroeck, M.V., Huang, C.-W., Stolcke, A., Maas, R., 2020. Efficient minimum word error rate training of RNN-transducer for end-to-end speech recognition. In: Proc. Interspeech 2020. pp. 2807–2811. http://dx.doi.org/10.21437/Interspeech.2020-1557.

Gutmann, M., Hyvärinen, A., 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, pp. 297–304.

Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R., Wu, Y., 2020. ContextNet: Improving convolutional neural networks for automatic speech recognition with global context. In: Proc. Interspeech 2020. pp. 3610–3614. http://dx.doi.org/10.21437/Interspeech.2020-2059.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al., 2014a. Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.

Hannun, A., Lee, A., Xu, Q., Collobert, R., 2019. Sequence-to-sequence speech recognition with time-depth separable convolutions. In: Proc. Interspeech 2019. pp. 3785–3789. http://dx.doi.org/10.21437/Interspeech.2019-2460.

Hannun, A.Y., Maas, A.L., Jurafsky, D., Ng, A.Y., 2014b. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. abs/1408.2873.

Hau, D., Chen, K., 2011. Exploring hierarchical speech representations with a deep convolutional neural network. In: Proceedings of UKCI'11. United Kingdom Annual Workshop on Computational Intelligence (UKCI'11) ; Conference date: 07-09-2011 Through 09-09-2011.

He, Y., Sainath, T.N., Prabhavalkar, R., McGraw, I., Alvarez, R., Zhao, D., Rybach, D., Kannan, A., Wu, Y., Pang, R., et al., 2019. Streaming end-to-end speech recognition for mobile devices. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6381–6385.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., Estève, Y., 2018. TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In: International Conference on Speech and Computer. Springer, pp. 198–208.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag. 29 (6), 82–97.

Hinton, G.E., Zemel, R., 1993. Autoencoders, minimum description length and Helmholtz free energy. In: Advances in Neural Information Processing Systems, vol. 6.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Hori, T., Watanabe, S., Zhang, Y., Chan, W., 2017. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. In: Proc. Interspeech 2017. pp. 949–953. http://dx.doi.org/10.21437/Interspeech.2017-1296.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S., 2019. Parameter-efficient transfer learning for NLP. In: International Conference on Machine Learning. PMLR, pp. 2790–2799.

Hrinchuk, O., Popova, M., Ginsburg, B., 2020. Correction of automatic speech recognition with transformer sequence-to-sequence model. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7074–7078.

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 Trans. Audio Speech Lang. Process. 29, 3451–3460.

Hsu, J.-Y., Chen, Y.-J., Lee, H.-y., 2020. Meta learning for end-to-end low-resource speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7844–7848.

Hsu, W.-N., Zhang, Y., Glass, J., 2016. A prioritized grid long short-term memory RNN for speech recognition. In: 2016 IEEE Spoken Language Technology Workshop. SLT, IEEE, pp. 467–473.

Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141.

Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., 2022. LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Huang, W., Hu, W., Yeung, Y.T., Chen, X., 2020. Conv-transformer transducer: Low latency, low frame rate, streamable end-to-end speech recognition. In: Proc. Interspeech 2020. pp. 5001–5005. http://dx.doi.org/10.21437/Interspeech.2020-2361.

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. IEEE, pp. 7304–7308.

Inaguma, H., Cho, J., Baskar, M.K., Kawahara, T., Watanabe, S., 2019. Transfer learning of language-independent end-to-end ASR with language model fusion. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6096–6100.

Irie, K., 2020. Advancing Neural Language Modeling in Automatic Speech Recognition (Ph.D. thesis). RWTH Aachen University.

Ivanko, D., Karpov, A., Fedotov, D., Kipyatkova, I., Ryumin, D., Ivanko, D., Minker, W., Zelezny, M., 2018. Multimodal speech recognition: increasing accuracy using high speed video data. J. Multimodal User Interfaces 12, 319–328.

Jaitly, N., Le, Q.V., Vinyals, O., Sutskever, I., Sussillo, D., Bengio, S., 2016. An online sequence-to-sequence model using partial conditioning. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (Eds.), In: Advances in Neural Information Processing Systems, vol. 29, Curran Associates, Inc., URL https://proceedings.neurips.cc/paper_files/paper/2016/file/312351bff07989769097660a56395065-Paper.pdf.

Jaitly, N., Nguyen, P., Senior, A.W., Vanhoucke, V., 2012. Application of pretrained deep neural networks to large vocabulary speech recognition. In: Interspeech.

Jeon, J., Kim, E., 2020. Multitask learning and joint optimization for transformer-RNN-transducer speech recognition. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 6793–6797, URL https://api.semanticscholar.org/CorpusID:226227027.

Jiang, D., Li, W., Cao, M., Zou, W., Li, X., 2021. Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning. In: Proc. Interspeech 2021. pp. 1544–1548. http://dx.doi.org/10.21437/Interspeech.2021-391.

Jordan, M.I., Mitchell, T.M., 2015. Machine learning: Trends, perspectives, and prospects. Science 349 (6245), 255–260.

Jozefowicz, R., Zaremba, W., Sutskever, I., 2015. An empirical exploration of recurrent network architectures. In: International Conference on Machine Learning. pp. 2342–2350.

Juan, S., Flora, S., 2015. Exploiting Resources from Closely-Related Languages for Automatic Speech Recognition in Low-Resource Languages from Malaysia (Ph.D. thesis). Université Grenoble Alpes (ComUE).

Juang, B.-H., 1985. Maximum-likelihood estimation for mixture multivariate stochastic observations of Markov chains. AT&T Tech. J. 64 (6), 1235–1249.

Kahn, J., Lee, A., Hannun, A., 2020. Self-training for end-to-end speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7084–7088.

Kalchbrenner, N., Danihelka, I., Graves, A., 2015. Grid long short-term memory. arXiv preprint arXiv:1507.01526.

Kannan, A., Wu, Y., Nguyen, P., Sainath, T.N., Chen, Z., Prabhavalkar, R., 2018. An analysis of incorporating an external language model into a sequence-to-sequence model. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 1–5828.

Katz, S., 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans. Acoust. Speech Signal Process. 35 (3), 400–401.

Kim, J., El-Khamy, M., Lee, J., 2017a. Residual LSTM: Design of a deep recurrent architecture for distant speech recognition. In: Proc. Interspeech 2017. pp. 1591–1595. http://dx.doi.org/10.21437/Interspeech.2017-477.

Kim, H., Hasegawa-Johnson, M., Perlman, A., Gunderson, J., Huang, T.S., Watkin, K., Frame, S., 2008. Dysarthric speech database for universal access research. In: Ninth Annual Conference of the International Speech Communication Association.

Kim, S., Hori, T., Watanabe, S., 2017b. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4835–4839.

Kim, K., Lee, K., Gowda, D., Park, J., Kim, S., Jin, S., Lee, Y.-Y., Yeo, J., Kim, D., Jung, S., et al., 2019. Attention based on-device streaming speech recognition with large speech corpus. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop. ASRU, IEEE, pp. 956–963.

Klatt, D.H., 1977. Review of the ARPA speech understanding project. J. Acoust. Soc. Am. 62 (6), 1345–1366.

Ko, T., Peddinti, V., Povey, D., Khudanpur, S., 2015. Audio augmentation for speech recognition. In: Interspeech, vol. 2015, p. 3586.

Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M., Mengesha, Z., Toups, C., Rickford, J.R., Jurafsky, D., Goel, S., 2020. Racial disparities in automated speech recognition. Proc. Natl. Acad. Sci. 117 (14), 7684–7689.

Kombrink, S., Mikolov, T., Karafiát, M., Burget, L., 2011. Recurrent neural network based language modeling in meeting recognition. In: Interspeech, vol. 11, pp. 2877–2880.

Kriman, S., Beliaev, S., Ginsburg, B., Huang, J., Kuchaiev, O., Lavrukhin, V., Leary, R., Li, J., Zhang, Y., 2020. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6124–6128.

Kunze, J., Kirsch, L., Kurenkov, I., Krug, A., Johannsmeier, J., Stober, S., 2017. Transfer learning for speech recognition on a budget. In: Blunsom, P., Bordes, A., Cho, K., Cohen, S., Dyer, C., Grefenstette, E., Hermann, K.M., Rimell, L., Weston, J., Yih, S. (Eds.), Proceedings of the 2nd Workshop on Representation Learning for NLP. Association for Computational Linguistics, Vancouver, Canada, pp. 168–177. http://dx.doi.org/10.18653/v1/W17-2620, URL https://aclanthology.org/W17-2620.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1 (4), 541–551.

Lee, K.-F., Hon, H.-W., Reddy, R., 1990. An overview of the SPHINX speech recognition system. IEEE Trans. Acoust. Speech Signal Process. 38 (1), 35–45.

Lee, H., Pham, P., Largman, Y., Ng, A.Y., 2009. Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Advances in Neural Information Processing Systems. pp. 1096–1104.

Lee, J., Watanabe, S., 2021. Intermediate loss regularization for ctc-based speech recognition. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6224–6228.

Levinson, S.E., Rabiner, L.R., Sondhi, M.M., 1983. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. Bell Syst. Tech. J. 62 (4), 1035–1074.

Li, J., Deng, L., Haeb-Umbach, R., Gong, Y., 2015. Robust Automatic Speech Recognition: A Bridge to Practical Applications. Academic Press.

Li, J., Lavrukhin, V., Ginsburg, B., Leary, R., Kuchaiev, O., Cohen, J.M., Nguyen, H., Gadde, R.T., 2019a. Jasper: An end-to-end convolutional neural acoustic model. In: Proc. Interspeech 2019. pp. 71–75. http://dx.doi.org/10.21437/Interspeech.2019-1819.

Li, X.L., Liang, P., 2021. Prefix-tuning: Optimizing continuous prompts for generation. In: Zong, C., Xia, F., Li, W., Navigli, R. (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Online, pp. 4582–4597. http://dx.doi.org/10.18653/v1/2021.acl-long.353, URL https://aclanthology.org/2021.acl-long.353.

Li, J., Liu, C., Gong, Y., 2018. Layer trajectory LSTM. In: Proc. Interspeech 2018. pp. 1768–1772. http://dx.doi.org/10.21437/Interspeech.2018-1485.

Li, J., Lu, L., Liu, C., Gong, Y., 2019b. Improving layer trajectory LSTM with future context frames. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6550–6554.

Li, X., Wu, X., 2015. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4520–4524.

Li, J., Zhao, R., Sun, E., Wong, J.H., Das, A., Meng, Z., Gong, Y., 2020. High-accuracy and low-latency speech recognition with two-head contextual layer trajectory LSTM model. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7699–7703.

Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y., 2017. A structured self-attentive sentence embedding. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=BJC_jUqxe.

Ling, S., Liu, Y., Salazar, J., Kirchhoff, K., 2020. Deep contextualized acoustic representations for semi-supervised speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6429–6433.

Liptchinsky, V., Synnaeve, G., Collobert, R., 2017. Letter-based speech recognition with gated convnets. 1, CoRR, abs/1712.09444.

Liu, A.H., Chung, Y.-A., Glass, J., 2021a. Non-autoregressive predictive coding for learning speech representations from local dependencies. In: Proc. Interspeech 2021. pp. 3730–3734. http://dx.doi.org/10.21437/Interspeech.2021-349.

Liu, A.T., Li, S.-W., Lee, H.-y., 2021b. Tera: Self-supervised learning of transformer encoder representation for speech. IEEE/ACM Trans. Audio Speech Lang. Process. 29, 2351–2366.

Liu*, P.J., Saleh*, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., Shazeer, N., 2018. Generating wikipedia by summarizing long sequences. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=Hyg0vbWC-.

López-Cózar, R., Callejas, Z., Griol, D., Quesada, J.F., 2014. Review of spoken dialogue systems. Loquens 1 (2), 012.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lowerre, B.T., 1976. The HARPY speech recognition system. URL https://api.semanticscholar.org/CorpusID:61409851.

Lu, L., Zhang, X., Renais, S., 2016. On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5060–5064.

Ma, J., Matsoukas, S., Kimball, O., Schwartz, R., 2006. Unsupervised training on large amounts of broadcast news data. In: 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 3, IEEE, III–III.

Maas, A.L., Hannun, A.Y., Jurafsky, D., Ng, A.Y., 2014. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. CoRR abs/1408.2873.

Menendez-Pidal, X., Polikoff, J.B., Peters, S.M., Leonzio, J.E., Bunnell, H.T., 1996. The nemours database of dysarthric speech. In: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96, vol. 3, IEEE, pp. 1962–1965.

Meng, L., Xu, J., Tan, X., Wang, J., Qin, T., Xu, B., 2021. Mixspeech: Data augmentation for low-resource automatic speech recognition. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7008–7012.

Meyer, J., 2019. Multi-Task and Transfer Learning in Low-Resource Speech Recognition (Ph.D. thesis). The University of Arizona.

Miao, H., Cheng, G., Gao, C., Zhang, P., Yan, Y., 2020a. Transformer-based online CTC/attention end-to-end speech recognition architecture. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6084–6088.

Miao, H., Cheng, G., Zhang, P., Yan, Y., 2020b. Online hybrid ctc/attention end-to-end automatic speech recognition architecture. IEEE/ACM Trans. Audio Speech Lang. Process. 28, 1452–1465.

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. ASRU, IEEE, pp. 167–174.

Miao, Y., Li, J., Wang, Y., Zhang, S.-X., Gong, Y., 2016. Simplifying long short-term memory acoustic models for fast training and decoding. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 2284–2288.

Miao, Y., Metze, F., 2015. On speaker adaptation of long short-term memory recurrent neural networks. In: Sixteenth Annual Conference of the International Speech Communication Association.

Milde, B., Biemann, C., 2018. Unspeech: Unsupervised speech context embeddings. In: Proc. Interspeech 2018. pp. 2693–2697. http://dx.doi.org/10.21437/Interspeech.2018-2194.

Mohamed, A., Okhonko, D., Zettlemoyer, L., 2019. Transformers with convolutional context for ASR. CoRR abs/1904.11660.

Mohri, M., 1997. Finite-state transducers in language and speech processing. Comput. Linguist. 23 (2), 269–311.

Moore, M., Venkateswara, H., Panchanathan, S., 2018. Whistle-blowing asrs: evaluating the need for more inclusive automatic speech recognition systems. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 2018, pp. 466–470.

Moritz, N., Hori, T., Le Roux, J., 2019a. Streaming end-to-end speech recognition with joint CTC-attention based models. In: 2019 IEEE Automatic Speech Recognition and Understanding Workshop. ASRU, IEEE, pp. 936–943.

Moritz, N., Hori, T., Le Roux, J., 2019b. Unidirectional neural network architectures for end-to-end automatic speech recognition. In: INTERSPEECH. pp. 76–80.

Nagrani, A., Chung, J.S., Zisserman, A., 2017. VoxCeleb: A large-scale speaker identification dataset. In: Proc. Interspeech 2017. pp. 2616–2620. http://dx.doi.org/10.21437/Interspeech.2017-950.

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.

Nozaki, J., Komatsu, T., 2021. Relaxing the conditional independence assumption of CTC-based ASR by conditioning on intermediate predictions. In: Interspeech 2021. pp. 3735–3739. http://dx.doi.org/10.21437/Interspeech.2021-911.

Ochiai, T., Watanabe, S., Katagiri, S., Hori, T., Hershey, J., 2018. Speaker adaptation for multichannel end-to-end speech recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6707–6711.

Olson, H.F., Belar, H., 1956. Phonetic typewriter. J. Acoust. Soc. Am. 28 (6), 1072–1081.

Oord, A.v.d., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

O'Shaughnessy, D., 1988. Linear predictive coding. IEEE Potentials 7 (1), 29–32.

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. ICASSP, IEEE, pp. 5206–5210.

Pandey, A., Wang, D., 2018. On adversarial training and loss functions for speech enhancement. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5414–5418.

Park, D.S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E.D., Le, Q.V., 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In: Proc. Interspeech 2019. pp. 2613–2617. http://dx.doi.org/10.21437/Interspeech.2019-2680.

Pascual, S., Ravanelli, M., Serrà, J., Bonafonte, A., Bengio, Y., 2019. Learning problem-agnostic speech representations from multiple self-supervised tasks. In: Proc. Interspeech 2019. pp. 161–165. http://dx.doi.org/10.21437/Interspeech.2019-2605.

Paul, D.B., Baker, J., 1992. The design for the wall street journal-based CSR corpus. In: Speech and Natural Language: Proceedings of a Workshop Held At Harriman, New York, February 23-26, 1992.

Peddinti, V., Povey, D., Khudanpur, S., 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In: Sixteenth Annual Conference of the International Speech Communication Association.

Pitt, M.A., Johnson, K., Hume, E., Kiesling, S., Raymond, W., 2005. The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability. Speech Commun. 45 (1), 89–95.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al., 2011. The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. (CONF), IEEE Signal Processing Society.

Pundak, G., Sainath, T., 2016. Lower frame rate neural network acoustic models.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77 (2), 257–286.

Raffel, C., Luong, M.-T., Liu, P.J., Weiss, R.J., Eck, D., 2017. Online and linear-time attention by enforcing monotonic alignments. In: International Conference on Machine Learning. PMLR, pp. 2837–2846.

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. ASRU, IEEE, pp. 193–199.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., Bengio, Y., 2020. Multi-task self-supervised learning for robust speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6989–6993.

Rezende, D.J., Mohamed, S., Wierstra, D., 2014. Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning. PMLR, pp. 1278–1286.

Roger, V., Farinas, J., Pinquier, J., 2022. Deep neural networks for automatic speech processing: a survey from large corpora to limited data. EURASIP J. Audio Speech Music Process. 2022 (1), 19.

Rousseau, A., Deléglise, P., Esteve, Y., 2012. TED-LIUM: an automatic speech recognition dedicated corpus.. In: LREC. pp. 125–129.

Rousseau, A., Deléglise, P., Esteve, Y., et al., 2014. Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks. In: LREC. pp. 3935–3939.

Rudzicz, F., Namasivayam, A.K., Wolff, T., 2012. The TORGO database of acoustic and articulatory speech from speakers with dysarthria. Lang. Resour. Eval. 46 (4), 523–541.

Sadhu, S., He, D., Huang, C.-W., Mallidi, S.H., Wu, M., Rastrow, A., Stolcke, A., Droppo, J., Maas, R., 2021. Wav2vec-c: A self-supervised model for speech representation learning. arXiv preprint arXiv:2103.08393.

Sainath, T.N., Kingsbury, B., Ramabhadran, B., Fousek, P., Novak, P., Mohamed, A.-r., 2011. Making deep belief networks effective for large vocabulary continuous speech recognition. In: 2011 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, pp. 30–35.

Sainath, T.N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A.-r., Dahl, G., Ramabhadran, B., 2015a. Deep convolutional neural networks for large-scale speech tasks. Neural Netw. 64, 39–48.

Sainath, T.N., Li, B., 2016. Modeling time-frequency patterns with LSTM vs. Convolutional architectures for LVCSR tasks. In: Interspeech.

Sainath, T.N., Mohamed, A.-r., Kingsbury, B., Ramabhadran, B., 2013. Deep convolutional neural networks for LVCSR. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, pp. 8614–8618.

Sainath, T.N., Vinyals, O., Senior, A., Sak, H., 2015b. Convolutional, long short-term memory, fully connected deep neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4580–4584.

Sak, H., Senior, A.W., Beaufays, F., 2014a. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: Interspeech.

Sak, H., Vinyals, O., Heigold, G., Senior, A., McDermott, E., Monga, R., Mao, M., 2014. Sequence discriminative distributed training of long short-term memory recurrent neural networks. In: Fifteenth Annual Conference of the International Speech Communication Association.

Scholz, K.W., Irwin, J.S., Tamri, S., 2006. Dialogue flow interpreter development tool. US Patent 7, 024, 348.

Shahnawazuddin, S., Dey, A., Sinha, R., 2016. Pitch-adaptive front-end features for robust children's ASR. In: Interspeech. pp. 3459–3463.

Soltau, H., Liao, H., Sak, H., 2017. Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition. In: Proc. Interspeech 2017. pp. 3707–3711. http://dx.doi.org/10.21437/Interspeech.2017-1566.

Song, W., Cai, J., 2015. End-to-End Deep Neural Network for Automatic Speech Recognition. Standford CS224D Reports.

Song, X., Wang, G., Huang, Y., Wu, Z., Su, D., Meng, H., 2020. Speech-xlnet: Unsupervised acoustic model pretraining for self-attention networks. In: Proc. Interspeech 2020. pp. 3765–3769. http://dx.doi.org/10.21437/Interspeech.2020-1511.

Sperber, M., Niehues, J., Neubig, G., Stüker, S., Waibel, A., 2018. Self-attentional acoustic models. In: Proc. Interspeech 2018. pp. 3723–3727. http://dx.doi.org/10.21437/Interspeech.2018-1910.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Stuttle, M.N., 2003. A Gaussian Mixture Model Spectral Representation for Speech Recognition (Ph.D. thesis). University of Cambridge.

Sun, S., Yeh, C.-F., Hwang, M.-Y., Ostendorf, M., Xie, L., 2018. Domain adversarial training for accented speech recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4854–4858.

Sunder, V., Thomas, S., Kuo, H.-K.J., Kingsbury, B., Fosler-Lussier, E., 2023. Fine-grained textual knowledge transfer to improve RNN transducers for speech recognition and understanding. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 1–5.

Sundermeyer, M., Schlüter, R., Ney, H., 2012. LSTM neural networks for language modeling. In: Thirteenth Annual Conference of the International Speech Communication Association.

Szymański, P., Żelasko, P., Morzy, M., Szymczak, A., Żyła-Hoppe, M., Banaszczak, J., Augustyniak, L., Mizgajski, J., Carmiel, Y., 2020. WER we are and WER we think we are. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, Online, pp. 3290–3295. http://dx.doi.org/10.18653/v1/2020.findings-emnlp.295, URL https://aclanthology.org/2020.findings-emnlp.295.

Tachbelie, M.Y., Abate, S.T., Besacier, L., 2014. Using different acoustic, lexical and language modeling units for ASR of an under-resourced language–Amharic. Speech Commun. 56, 181–194.

Tóth, L., Kovács, G., Van Compernolle, D., 2018. A perceptually inspired data augmentation method for noise robust cnn acoustic models. In: Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20. Springer, pp. 697–706.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al., 2023. Llama: Open and efficient foundation language models (2023). arXiv preprint arXiv:2302.13971.

Tripathi, A., Sak, H., Lu, H., Zhang, Q., Kim, J., 2022. Transformer transducer: One model unifying streaming and non-streaming speech recognition. US Patent App. 17/210, 465.

Van Den Oord, A., Vinyals, O., et al., 2017. Neural discrete representation learning. In: Advances in Neural Information Processing Systems, vol. 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008.

Vintsyuk, T.K., 1968. Speech discrimination by dynamic programming. Cybernetics 4 (1), 52–57.

Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., Huang, H., Tjandra, A., Zhang, X., Zhang, F., et al., 2020. Transformer-based acoustic modeling for hybrid speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6874–6878.

Wang, D., Wang, X., Lv, S., 2019a. An overview of end-to-end automatic speech recognition. Symmetry 11 (8), 1018.

Wang, Y., Wang, H., et al., 2017. Multilingual convolutional, long short-term memory, deep neural networks for low resource speech recognition. Procedia Comput. Sci. 107, 842–847.

Wang, C., Wu, Y., Du, Y., Li, J., Liu, S., Lu, L., Ren, S., Ye, G., Zhao, S., Zhou, M., 2019b. Semantic mask for transformer based end-to-end speech recognition. arXiv preprint arXiv:1912.03010.

Wang, D., Zheng, T.F., 2015. Transfer learning for speech and language processing. In: 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. APSIPA, IEEE, pp. 1225–1237.

Watanabe, S., Hori, T., Kim, S., Hershey, J.R., Hayashi, T., 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE J. Sel. Top. Sign. Proces. 11 (8), 1240–1253.

Woodland, P.C., Young, S.J., 1993. The HTK tied-state continuous speech recogniser. In: Eurospeech. Citeseer.

Wu, Z., Li, B., Zhang, Y., Aleksic, P.S., Sainath, T.N., 2020. Multistate encoding with end-to-end speech RNN transducer network. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7819–7823.

Wu*, Z., Liu*, Z., Lin, J., Lin, Y., Han, S., 2020. Lite transformer with long-short range attention. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=ByeMPlHKPH.

Wu, F., et al., 2020. Child Speech Recognition as Low Resource Automatic Speech Recognition (Ph.D. thesis). Johns Hopkins University.

Xie, J., Hansen, J.H.L., 2023. Mixrep: Hidden representation mixup for low-resource speech recognition. In: INTERSPEECH 2023. URL https://api.semanticscholar.org/CorpusID:260913408.

Yan, R., 2018. " chitty-chitty-chat bot": Deep learning for conversational AI. In: IJCAI. 18, pp. 5520–5526.

Yan, J., Yu, H., Li, G., 2018. Tibetan acoustic model research based on TDNN. In: 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. APSIPA ASC, IEEE, pp. 601–604.

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

Yi, J., Tao, J., Wen, Z., Bai, Y., 2018. Language-adversarial transfer learning for low-resource speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 27 (3), 621–630.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al., 2002. The HTK book. Camb. Univ. Eng. Dep. 3 (175), 12.

Yu, D., Deng, L., Dahl, G., 2010. Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition. In: Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning.

Yu, C., Kang, M., Chen, Y., Li, M., Dai, T., 2019. Endangered tujia language speech enhancement research based on improved DCGAN. In: Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18. Springer, pp. 394–404.

Yu, D., Li, J., 2017. Recent progresses in deep learning based acoustic models. IEEE/CAA J. Autom. Sin. 4 (3), 396–409.

Yu, C., Yu, J., Qian, Z., Tan, Y., 2023. Improvement of acoustic models fused with lip visual information for low-resource speech. Sensors 23 (4), 2071.

Zeyer, A., Schlüter, R., Ney, H., 2016. Towards online-recognition with deep bidirectional LSTM acoustic models. In: Interspeech. pp. 3424–3428.

Zhang, Y., Chen, G., Yu, D., Yaco, K., Khudanpur, S., Glass, J., 2016a. Highway long short-term memory rnns for distant speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5755–5759.

Zhang, S., Liu, C., Jiang, H., Wei, S., Dai, L., Hu, Y., 2017. Nonrecurrent neural structure for long-term dependence. IEEE/ACM Trans. Audio Speech Lang. Process. 25 (4), 871–884.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Laurent, C., Bengio, Y., Courville, A., 2016b. Towards end-to-end speech recognition with deep convolutional neural networks. In: Proc. Interspeech 2016. pp. 410–414. http://dx.doi.org/10.21437/Interspeech.2016-1446.

Zhao, Y., Xu, S., Xu, B., 2016. Multidimensional residual learning based on recurrent neural networks for acoustic modeling. In: Interspeech. pp. 3419–3423.

Zhou, W., Schlüter, R., Ney, H., 2020. Robust beam search for encoder-decoder attention based speech recognition without length bias. In: Proc. Interspeech 2020. pp. 1768–1772. http://dx.doi.org/10.21437/Interspeech.2020-1958.

Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M., 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592.